The Usefulness of Multi-Sensor Affect Detection on User Experience

An Application of Biometric Measurement Systems on Online Purchasing

by

Irfan Kula

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2018 by the
Graduate Supervisory Committee:

Robert K. Atkinson, Co-Chair
Rod D. Roscoe, Co-Chair
Russell J. Branaghan

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

Traditional usability methods in Human-Computer Interaction (HCI) have been extensively used to understand the usability of products. Measurements of user experience (UX) in traditional HCI studies mostly rely on task performance and observable user interactions with the product or services, such as usability tests, contextual inquiry, and subjective self-report data, including questionnaires, interviews, and usability tests. However, these studies fail to directly reflect a user's psychological involvement and further fail to explain the cognitive processing and the related emotional arousal. Thus, capturing how users think and feel when they are using a product remains a vital challenge of user experience evaluation studies. Conversely, recent research has revealed that sensor-based affect detection technologies, such as eye tracking, electroencephalography (EEG), galvanic skin response (GSR), and facial expression analysis, effectively capture affective states and physiological responses. These methods are efficient indicators of cognitive involvement and emotional arousal and constitute effective strategies for a comprehensive measurement of UX. The literature review shows that the impacts of sensor-based affect detection systems to the UX can be categorized in two groups: (1) *confirmatory* to validate the results obtained from the traditional usability methods in UX evaluations; and (2) *complementary* to enhance the findings or provide more precise and valid evidence. Both provided comprehensive findings to uncover the issues related to mental and physiological pathways to enhance the design of product and services. Therefore, this dissertation claims that it can be efficient to integrate sensor-based affect detection technologies to solve the current gaps or weaknesses of traditional usability methods. The dissertation revealed that the multi-sensor-based UX evaluation approach through biometrics tools and software corroborated user experience identified by traditional UX methods during an online purchasing task. The use these systems enhanced the findings and provided more precise and valid evidence to predict the consumer purchasing preferences. Thus, their impact was “*complementary*” on overall UX evaluation. The dissertation also provided information of the unique contributions of each tool and recommended some ways user experience researchers can combine both sensor-based and traditional UX approaches to explain consumer purchasing preferences.

DEDICATION

To my wife, Filiz'im

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my co-advisors Dr. Robert K. Atkinson and Dr. Rod D. Roscoe for the continuous support of my Ph.D. study and related research, for their patience, motivation, and immense knowledge. Particularly, Dr. Atkinson's great vision on biometrics, endless support and trust during all my academic adventures since my master program on Educational Technologies were milestones on my academic achievement. Providing the internship opportunity was fabulous as well. Likewise, Dr. Roscoe's great academic writing feedback and excellent teaching skills were priceless. His precious cordiality and patience during several brainstorming sessions, and excellent guidance on the rhetorical structure of my thesis was great marks on my academic acknowledgment. Therefore, I consider myself as a very lucky student as I had a chance to work with them.

I would like to thank my committee member, Prof. Russell J. Branaghan, for their insightful comments and encouragement, excellent user experience practices, which motivated me to widen my research from various perspectives. Also, I thank all professors that I had a chance to take their classes or get their recommendations in ASU, particularly Dr. George Runger, Dr. Nancy Cooke and Dr. Javier Gonzalez Sanchez.

Likewise, I express my deepest thanks to Aurel Coza, the director of the consumer behavior lab at Adidas, for his excellent coaching and I am grateful to meet so many wonderful professionals who led me through the internship period at Adidas.

I thank my fellow lab mate, Helen Chavez, in for her excellent lab management, friendship and for all the fun we have had in the last 6 years. Besides, endless thanks to Robert Christopherson and all support staff in iMotions for their excellent services. In addition, I am deeply indebted to my friends Nedim Yel, Burhan Senturk, Aysegul Demirtas, Kerem Demirtas, Mustafa Demir, Ghazal Shams for their support on the analysis section of this thesis.

Going back to academia as a graduate student after 35 years old was the greatest challenge for a family with two kids. I express my deep sense of gratitude and appreciation to my friends Edvin Cetegen for sharing his academic experience and fabulous coaching, and Erkan Kurnaz for his limitless financial consultancy to Caroline Savio-Ramos to let me know this Ph.D. opportunity and encourage me to apply for it. Also, I would like to thank Mike Brown, Emily Dimson, Sarah Snyder, Laurie Johnsonn and Sara Mayo-Knab for their continued support and help for editing and academic writing recommendations. Likewise, special thanks to Dobson, Cetegen and Demirtas families for their great supports and motivations, particularly to my family in Arizona while I was in Portland for the 6-month internship. Besides, I thank all my friends who help me to find subjects for my study and who support and motivate me to perform my best in this academic journey.

I am so grateful to my mom-in-law, Servet Kilavuz, for her great support to my family. Also, I would like to thank my mom, dad and sister in Turkey for supporting me spiritually throughout my academic adventure and all my life in general. Besides, special thanks to my daughters Ezgi and Deniz. I overcome my fatigue depressed mood and quickly refreshed, inspired and motivated and had a lot of fun with you.

Last but not the least, Filiz Kula, I am grateful that you were the part of my life. I cannot thank you enough for your endless love, trust, understanding, support, encouragement, and sunlight. I am looking forward to our new adventures ahead!

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

AC	: Affective Computing
BMS	: Biometric Measurement Systems
EEG	: Electroencephalography
ET	: Eye Tracking
FEA	: Facial Expression Analysis
GSR	: Galvanic Skin Response
HCI	: Human Computer Interaction
UX	: User Experience

CHAPTER 1

INTRODUCTION

The concept of the interactive user experience (UX) refers to a user's momentary interactions, stances, acts, cognitions, and emotions while experiencing a product or service (Kuniavsky, 2010; Albert & Tullis, 2013). Likewise, UX is defined as "a person's perceptions and responses that result from the use and/or anticipated use of a product, system or service" in ISO 9241-110:2010. These definitions incorporate all essential features of UX. They additionally focus on users' affective and cognitive states to explore what they feel and think while using a product. Thus, UX functions as an umbrella discipline for design (visual, interaction, instructional, and architectural) and usability, using user-centered principles and fundamental methods for measuring usability (Patterson & Erturk, 2015).

UX researchers who hope to provide useful user experience reports have sought ways to effectively measure the experience of users or customers in relation to specific products, systems or services. However, improving product usability/cost ratios and increasing user satisfaction is not a straightforward cause and effect issue. Researchers should develop expertise in understanding how users think and feel when they are using products, systems or services. This study aimed to help understand how to determine optimum tool selection to maximize the efficiency of UX evaluation reports. Particularly, the study investigated the usefulness of a biometric multisensor-based UX evaluation approach by focusing an application of biometric measurement systems on online purchasing.

This chapter is intended to provide brief information on the current problem and its recommended solutions. To that end, it is organized in the following subsections: (1) overview of the issues, (2) statement of the problem, (3) purpose of the study, (4) significance of the study, (5) limitations of the study, (6) brief comments and discussions of the study, and (7) organization of the dissertation document.

Overview of the Issues

Traditional UX approaches in Human-Computer Interaction (HCI) have been extensively used to understand the usability of products (Mirza-Babaei, Long, Foley, & McAllister, 2014). However, despite the advance of traditional UX methods, their measures of effectiveness, efficiency, and satisfaction may not good enough to provide evidence about how a user feels using a product (Zaharias & Poylymenakou, 2009). Such insights would benefit numerous fields, including home technology (Monk, 2002), global computing (Mankoff, Dey, Hsieh, Kientz, Ames, & Ledered, 2003), e-learning (Soloway, Guzdial, & Hay, 1994; Zaharias, 2004), video games (Mirza-babaei at al., 2014), as well as neuromarketing (Ramsey, 2014). Also, traditional methods tend to focus on objective usability measures (Hassenzahl, Burmester, & Koller, 2003) for pragmatic UX evaluations such as task completion time, error rate, and success rate. However, this measurement approach limits UX evaluations to a relatively narrow scope and may only partially capture users' actual psychological demands, processing, or affect (Laugwitz, Held, & Schrepp, 2008). In fact, the highly subjective nature of user experience (e.g., Norman, 2004, 2013) introduces substantial measurement challenges. Therefore, UX evaluation requires not only the collection of existing approaches for usability evaluation

(Vermeeren et al., 2010) but also requires human behavior data collection and analysis tools and expertise. Understanding how users think and feel when they are using the product is still a challenge for UX evaluation studies, and understanding human behavior is a baseline to overcome this challenge.

Statement of the Problem

Before discussing the statement of the problem, a brief mention of the interactions between various elements of human behavior might be helpful in understanding the problem, so the following paragraphs first provide some brief information about actions, cognitions, and emotions, and then discuss the essential traditional UX approaches to understanding how users think and feel when they are using the product, systems or services

Behavior can be simply called any observable action, such as driving, cleaning, talking, and texting. Scientific definitions include deeper understandings of behavior. Human behaviors represent combinations of actions, cognitions, and emotions. These elements are essential, since each takes part in forming the observed user response. Thus, better understanding user behavior requires addressing these key elements. In turn, addressing these elements may help to simplify the understanding of the complexity of human behavior.

An action represents every observable occurrence happening along a timeline. Actions are usually associated with perceptual processes. We may walk faster in a restaurant when we are hungry and slower after eating. Cognition describes thoughts and image processing in our mind. Adding a note to “buy a glass water bottle” to your to-do list involves verbal cognition. Meanwhile, imagining drinking water from that glass water bottle at the top of a hill after a trekking session is an example of nonverbal (imaginary) cognition. Unlike actions, cognitions cannot be observed or measured easily as summarized in Table 1. They can only be inferred from behavior.

Emotion is a neurological reaction to an emotional stimulus (Gonzalez-Sanchez, Baydogan, Chavez-Echeagaray, Atkinson, & Burleson, 2017). It occurs in the brain, particularly in a part of the limbic system, and causes reactions in the body that change its physical state. Like cognitions, emotions cannot be observed directly but can be inferred indirectly. Finally, emotions and feelings are used interchangeably in daily life conversations, though they are not the same. A feeling is generated by emotions and associated with previous knowledge, practices, and thoughts related to that specific emotion. An emotion is the affective state leading to that feeling. In other words, the feeling can be regarded as meaning assigned emotions based on what brain perceived.

Table 1: The Elements of the Human Behavior

Human Behavior	Detection Type	Conscious vs subconscious
Action	Observable	Mostly conducted by conscious processes.
Cognition	Inferred indirectly from observed behavior or psychophysiological data	Mostly conducted by conscious processes.
Emotion	Inferred indirectly from observed behavior or psychophysiological data	Mostly conducted by subconscious processes.

The interaction between actions, cognitions, and emotions enables a user to respond to his or her environment, (Frijda, Kuiper, & Ter Schure, 1989). However, these interactions are rarely one-way, sequential, cause-effect relationships. In other words, there is not a one-way sequence from action through emotion to cognition. In fact, interactions can involve several combinations, including all three core elements. Therefore, researchers who want to understand behavior must investigate action, cognition, and emotion. Similarly, a researcher who wants to understand emotion should research cognition, behaviors, and action.

Although human behavior is observable and at times seen to be rational and intentional, much human behavior is conducted by subconscious, often automated, processes (Nijboer, van de Laar, Gerritsen, Nijholt, & Poel, 2015). While some human behaviors are visible and easy to measure with the naked eye or simple tools and techniques, the vast majority of human cognitive processes is hidden and requires analysis and inference from visible behaviors. The challenges of measuring human behavior and cognitive processes are not easy to understand. To analyze and infer human behaviors, UX researchers have created several approaches and techniques that allow for accurate and precise data collection as well as possible behavioral, cognitive, and affective strategies.

Heuristic evaluations, usability tests, focus groups, contextual inquiry, verbal reports, and human performance tests are just a few examples of well-known UX measurement methods commonly used by researchers (Hanington & Martin, 2012). These methods allow researchers to collect data that may help to make the product more user-centered. However, the challenge is understanding how people think, feel and act while using or checking a product. Most of the time, researchers try to overcome this challenge by using verbal reports (surveys, questionnaires and interviews), focus groups, contextual inquiries, and other observation-based methods. For example, surveys require users to identify and articulate their insights and psychological responses to product features. However, users can be unreliable in performing these reflective and expressive tasks. Nevertheless, most interview methods allow analysts to clarify questions and tasks and to ask probing follow-up questions, which can improve the accuracy or thoroughness of responses (Hanington & Martin, 2012). These responses are still subjective, however, and users may not accurately recall their internal, psychological responses.

Focus groups are used to learn opinions, feelings, and attitudes about a product from a group of purposefully selected subjects and may provide some deep insights; however, these groups capture only conscious thoughts and feelings but are not effective in collecting real-time user insights and behaviors. Also, a focus group may not reflect the larger population and may be misleading if is used as the sole source of data.

Contextual inquiry, a UX method composed of observations and interviews in a relevant environment—usually where the real experience is taking place—is useful to understand communication flows and task sequence, as well as the artifacts and influence of the cultural and physical features that are present where the experience occurs (Beyer, Hugh, & Holtzblatt, 1997). However, researchers should observe all user attitudes and behaviors accurately to fully understand what users do and what they communicate, all of which requires expertise in research skills.

Likewise, while observing users' behaviors or performance reveals patterns of errors, skills, and interactions, users' psychological states nevertheless remain hidden and must be imperfectly inferred from external data. Thus, traditional measures, particularly surveys, questionnaires, interviews and observation-based methods, might have data gaps that could undermine the efficacy of user experience assessments. Neither contextual inquiry nor traditional methods can reveal real-time insights into users' unconscious processes and emotional dynamics. Unfortunately, measurements of UX in traditional HCI may fail to reflect a user's psychological involvement directly and further fail to explain cognitive processing and related emotional arousal (Yao et al., 2006). It is essential to capture users' thoughts and feelings when they are actually using the product, but this is still a significant challenge for UX evaluation studies. For this reason, researchers must develop new approaches and complimentary technologies, enabling researchers to gather data through comprehensive UX evaluation systems.

Researchers may be able to overcome this challenge with the use of a biometric sensor-based measurement system (BMS). These systems make use of technologies, such as eye tracking, electroencephalography (EEG), galvanic skin response (GSR), and facial expression, to automatically detect affective states (Calvo & D'Mello, 2012; Healey, 2014). Recent research has revealed that biometric sensor-based affect detection technologies are an effective indicator of cognitive involvement and emotional arousal and are therefore recommended as a complementary measure of UX (Yao et al., 2014). These high-tech tools let researchers infer emotional and cognitive states by capturing physiological and behavioral signals, such as electro-dermal activity, heart rate, eye tracking, and facial expressions (Calvo & D'Mello, 2010). Capturing these user dynamics provides researchers with uninterrupted and nonintrusive data about behavior and reactions during the real-time experience (Picard, 2010). Unlike the other tools used in traditional approaches, one of the most important benefits of biometric sensor-based measurement systems is the ability to capture physiological, emotional, and cognitive dynamics while users are actually interacting with the product. Traditional UX approaches cannot accomplish this because users may not explain clearly or accurately recall details when asked after the product experience (Ward & Marsden, 2003).

Furthermore, UX evaluations become more efficient when complimentary measures are combined and synchronized. By combining sensors in an integrated system, a BMS allows researchers to collect and analyze physiological and behavioral data. This information helps in developing a more detailed and holistic understanding of factors impacting human performance. For instance, combining EEG and GSR measurements connects the valence picked up by EEG with arousal derived from GSR. Similarly, combining eye tracking and EEG allows researchers to identify user interest in specific stimuli through variations in workload. Eye tracking delivers information about the exact orientation of the eyeball, enabling researchers to identify artifacts, such as blinks and saccades, to help decontaminate the EEG data. Thus, biometric multi-sensor-based affect detection technologies measure less conscious and less voluntary psychophysiological signals of affective states and may provide a more direct and objective inference into user insights (Nijboer, van de Laar, Gerritsen, Nijholt, & Poel, 2015). In the following sections, particularly in chapters 2 and 3, biometric affect detection technologies, the types of data they provide, and the research applying these technologies to user experience was analyzed and discussed in more detail.

Purpose and Research Questions of the Study

The study has four essential aims. (1) First one is to investigate the multi-sensor affect detection in UX evaluation systems by comparing and combining them with traditional UX methods, particularly self-reports. (2) Second is to make UX researchers' life easier by providing evidence if they need sensors for their experimental set-up, and if yes, which one or which combination. (3) Third, one is to decrease the research cost of UX studies by optimizing the tool selection for further study designs. The final one (4) was to test and reveal the challenges of biometric human behavior studies and provide feasible solutions. Therefore, the study intended to investigate the following research questions:

1. How closely can a sensor-based evaluation approach corroborate user experience identified by traditional UX methods during an online purchasing task?
2. Which model (sensor-based, traditional, or combined) can best explain customer preferences for purchasing?
3. Which sensors (separately or integrated) most fully explain customer preferences for purchasing?

To test these research questions, an experiment was therefore designed with a focus on BMS and consumer product ratings and ranking. The study leveraged eye tracking, GSR, EEG, facial expression analysis, and survey data to reveal cognitive and emotional responses in user experience evaluation systems.

Significance of the Study

Little has been published regarding the feasibility and efficacy of BMS or investigating the multi-sensor based UX evaluations systems. A study including all four sensors would fill current gaps in our understanding of user behaviors and how users think and feel when they are using products, systems or services. Biometric affect detection technologies are suggested as either a *confirmatory* way to confirm or validate traditional measures or as a *complementary* way to complement traditional UX approaches by assessing users' underlying psycho-psychological states (Kula, Atkinson, Branaghan, & Roscoe, 2018). The study would reveal if the multi-sensor based UX evaluations systems serve as a confirmatory or a complimentary function for comprehensive UX evaluation systems. Thus, the findings of the study would be highly important to create an optimized experimental set-up for UX researchers who are considering research with these devices. The study would be a key source in determining if sensor based UX measurements are necessary for more reliable and accurate UX evaluations. UX researchers may benefit from the study and plan future research accordingly. With limited budgets and time, determining which tools (either traditional or sensor-based) are more effective is imperative. The study may help UX researchers decide on which sensor combination is best for their study. Beside tool selection and design options, UX researchers may also learn about required skills, potential challenges, and their solutions. Therefore, this study would be a practical, evidence-based study to help researchers effectively design, conduct, and analyze data using optimized hybrid approaches using traditional and sensor-based measures.

The Organization of the Dissertation

This dissertation focuses on the overlapping fields of UX and affective computing (AC). Next, the background literature provided theoretical background and field practices related to these two domains. The feasibility of using multisensor-based affect detection technology was discussed. The following chapter, chapter 3, detailed the methodology and design of the experiment by discussing each research question. Chapter 4 stated the analyses and results. Finally, chapter 5 discussed the findings of both traditional and multisensor-based approaches, as well as discussed challenges, limitations and opportunities for further research.

CHAPTER 2

BACKGROUND LITERATURE

This background literature review aims to discuss if sensor-based affect detection systems can rectify current gaps in knowledge or weaknesses of traditional user experience evaluations. In fact, it inquires if the application of biometric multisensor affect detection systems, recently called biometric measurement systems (BMS) or biometrics, can provide the best solution for comprehensive UX evaluations. The chapter intends to identify the contributions of sensor-based affect detection technology on UX evaluation systems. In addition, the present research aims to determine appropriate BMS tools and variables for a variety of UX evaluation needs, as well as their benefits and limitations.

The chapter includes three main sections. The first section focuses on conceptualizations of UX, principles and key concepts that fall under the heading of UX, and the wide variety of usability methods and variables. The second section addresses the significance of affect detection on affective computing and provides evidence about the use of common sensor-based affect detection technologies, namely eye tracking, electroencephalography (EEG), galvanic skin response (GSR), and facial expression analysis (FEA). The final section of the chapter focuses on the usefulness of multi-sensor affect detection on UX evaluation systems.

The Domain of User Experience

Interest in UX practices and research has increased over the past 15 years. A search for the term "user experience" on the Google Scholar search engine reported 1.18M results from 2000 to 2005, 1.33M results from 2006 to 2010, 1.43M results from 2011 to 2015, and 262,000 results since 2016, even when excluding citations and patents. This chapter of the dissertation defines the fundamentals of the UX domain, concentrating on user-centered design, usability, and current methods in usability studies.

User Experience, Usability & User-Centered Design

The concept of UX refers to a user's momentary interactions, stances, cognition, and emotions while experiencing a particular product or service (Albert & Tullis, 2013; Kuniavsky, 2010). This definition incorporates all essential features of UX, such as timing, interaction, and affective or cognitive states. UX functions as an umbrella discipline combining user-centered principles and methods in design (visual, interactive, instructional, and architectural) and usability (Farrell & Nielsen, 2014; Patterson & Erturk, 2015).

The User Experience Professionals Association (UXPA) states that user experience design (UXD) is a discipline that encompasses all the components of design, such as navigational elements, interactive features, visuals and aesthetics, audio, and multimedia features (Baxter et al., 2015). Particularly, UX design is a multidisciplinary arena based on HCI, computer science, ergonomics, and psychology (Yayici, 2014, p. 14). Therefore, UX professionals and design teams may come from highly varied disciplines. Relevant fields include visual design, architecture, human factors, ergonomics, instructional design, education technology, computer science, psychology, marketing, business, anthropology, and industrial engineering (Farrell & Nielsen, 2014). This wide multidisciplinary range in UX design brings diverse skills, techniques and approaches to UX evaluation.

The International Organization for Standardization (ISO) defined usability in ISO 9241-11 as the "*extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*." For most corporate usability labs, usability simply means "*methods for improving ease-of-use during the design process"* (Nielsen, 2012). After several long-term studies on usability, Nielsen claimed that learnability, efficiency, memorability, errors, and satisfaction are the five key components of usability that make the concept of usability clear to researchers in HCI (Nielsen, 2012). While discussing the usability of a product, we must identify how easy it is for the users to learn to use it (learnability), how quickly users can use it (efficiency), how easy to remember to use it after a period of time (memorability), how many and what type of errors users make while using it (errors), and how pleasant the product is (satisfaction).

User-centered design (UCD), also called human centered design in the ISO 9241-210:2010 standards, is defined as "*an approach to systems design and development that purposes to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques.*" The emergence of UCD goes back to the 1980s; Norman and Draper created the concept of UCD in 1986 (Norman & Draper, 1986). Norman claimed that, rather than using an exclusive technology or serving as a well-designed piece of programming, the ultimate purpose of the UCD system is to serve the user directly. Therefore, the users' requirements must lead the interface design (Norman, 1986). In other words, the philosophy of UCD is that the product, service, or the system must adapt to the user requirements, instead of forcing the user to adapt to the system (Baxter et al., 2015).

The principle approach of UCD has evolved through the years, and so has the concept of usability. In 1985, Gould & Lewis described 3 essential principles: *(1) an early focus on users and tasks, (2) empirical measurement of product usage, and (3) iterative design as UCD principles.* An early focus on users means the identification of the user's needs, missions, visions, and environment where the proposed tasks are performed (Gould & Lewis, 1985). It requires detailed, user-centered, need analysis for effective production, implementation and revision processes. In practical cases, the first principle also implies that the earlier the user is involved, the less revision is required at later stages of the production cycle (Baxter et al., 2015). The second primary empirical measure of product usage focuses on classical usability, which encompasses the ease of learning and effectiveness, or error-free use. In the application of this principle, users run simulations and use samples to perform work. Meanwhile, their interactions and responses are observed, recorded, and analyzed. Once users report issues, they must be resolved by revising the product or service. The revision stage is the starting point of the third principle. The system iteratively checks to ensure the issue no longer exists (Gould & Lewis, 1985).

Gould and Lewis (1985) conducted a survey to reveal how much their proposed principles benefited UX researchers. Surprisingly, their study showed that these principles were not sufficiently understood and were not used or recommended enough.

Thus, Gould and Lewis (1985) commented that the principles were undervalued. They claimed that once these three principles are well understood and applied to the systems, systems would be more effective, efficient, and satisfactory.

Along with the evolution of UX and user needs through the last decades, ISO 9241-210 standards state six user-centered key principles:

1. *The design is based upon an explicit understanding of users, tasks, and environments.*
2. *Users are involved throughout design and development.*
3. *The design is driven and refined by user-centered evaluation.*
4. *The process is iterative.*
5. *The design addresses the whole user experience.*
6. *The design team includes multidisciplinary skills and perspectives.*

User Experience Evaluation Approaches

Although human behavior is observable and may appear to be rational and intentional, much human behavior is driven by subconscious processes, which may be automated and out of consciousness (Kahneman, 1973). This means that, while some human behaviors are easily measured with the naked eye or simple tools, much human motivation is hidden and requires inference from visible behaviors. The challenges of measuring human behavior make determining adequate algorithms complex. To analyze and infer human behavior, user experience researchers have developed several approaches and techniques that allow for accurate and precise data collection. Heuristic evaluations, usability tests, focus groups, contextual inquiry, verbal reports, and human performance tests are just a few examples of well-known UX methods. These methods allow researchers to attain evidence for making the product user-centered. However, the challenge is to understand how people think, feel and act while actually using or viewing a product. Often, researchers try to overcome this challenge by conducting verbal reports, focus groups, customer satisfaction measures and other observation-based methods.

Verbal reports, including surveys, interviews, and questionnaires, are tools to capture self-reported behaviors, skills, profiles, and demographics of users. Collecting data from a large sample in a limited time with little cost may be an additional reason why researchers often prefer surveys. However, verbal reports are quick snapshots and gather limited or filtered inferences of behavior, thoughts, and emotions related to the performed experimental activities in the studies. For example, Net Promoter Score (NPS), which is a well-known but also well-criticized user satisfaction measurement through one question developed by Reichheld in 2003. Although NPS was highly innovative, particularly for corporations looking for agile consumer satisfaction measurements in the first decade of the millennium age (Reichheld & Markey, 2011), it seems that NPS is severely limited, since it is mostly focused on rational aspects of an experience and ignores emotional ones (Shaw, 2007, p120-135).

Surveys including questionnaires and scales are among the most flexible and useful methods, but that ease may also lead to misuse (Green, Dunn, & Hoonhout, 2008). This concern arises from uncertainty regarding the extent to which these tools should be used and under what conditions in order to maintain validity and reliability (Vermeeren, Law, Roto, Obrist, Hoonhout, & Väänänen-Vainio-Mattila, 2010). Deviation of users' answers from their true values on the measure may cause measurement error, which may stem from users' lack of motivation, concentration, or comprehension. It could also arise from weak survey design, perhaps using inappropriate wording or instructional and technical flaws. Thus, verbal reports may fail to reflect and record a user's real-time psychological involvement directly and further fail to explain cognitive processing and related emotional arousal (Yao et al., 2014; Mandryk, Atkins, & Inkpen, 2006).

Besides surveys, another approach to gathering verbal data is focus groups, a small number of carefully recruited participants—usually less than 10. Researchers using focus groups aim to focus on opinions, experiences, feelings, and attitudes towards a product. The role of the moderator is crucial. When guided by a capable moderator, a focus group can provide deep insights. Unlike individual evaluations, the strength of focus groups is dynamism, which provides collaborative discourse for generalizations and consensus about the discussion topics and the product. Focus groups may also be time-efficient if participants can quickly accept one another as peers, thus reducing fear of being judged and making it more likely they will openly share opinions, experiences, and feelings. However, focus groups capture only conscious thoughts and feelings, so they are not able to collect actual, real-time user insights and behaviors. Further, a focus group may not reflect a larger population and may be misleading if used alone.

Researchers should keep in mind that, though surveys and focus groups can be effective UX tools to capture conscious thoughts and emotions, much of human behavior is driven subconsciously. These subconscious processes regulate how user behavior is eventually manifested. Traditional methods, such as surveys and focus groups, collect only a minor segment of this, so researchers should supplement these methods with appropriate quantitative and qualitative methods.

To minimize the potential weakness of surveys and focus groups, researchers may add additional UX approaches, such as contextual inquiry, think-aloud sessions and retrospective analysis to make their design multimodal. Contextual inquiry consists of observations and interviews in the in-context environment, usually where the intended real experience would take place. This helps to understand communication flows, task sequence, artifacts, and the role of cultural and physical features as they would typically be experienced (Beyer & Holtzblatt, 1997). Though contextual inquiry usually takes place in a workplace to reveal the insights and hidden pieces of evidence that cause user experience problems, it can be used in many environments, such as shopping establishments, sports arenas, and any online interactive environment. In this approach, customers or users show how and why problems occur. The researchers' role is to ask clear direct questions related to the tasks or navigational features of the product and keep the user on topic. While performing a contextual inquiry approach, researchers should carefully observe all user attitudes and behaviors. This is not only while users are interacting with the product but also while communicating with other team members or their environment (Holtzblatt, Wendell, & Wood, 2004).

Besides verbal reports, focus groups, observations and interviews, more novel methods exist. Diary studies, directed storytelling, user journey maps, and think-alouds are also useful UX methods to analyze human behavior. However, none of these methods can reveal real-time user insights into unconscious processes and emotions. Although data about the users' cognitive processing and affective states provide crucial information for identifying usability issues and increasing user satisfaction (Gaver & Martin, 2000; Hassenzahl & Tractinsky, 2006), UX measurements in traditional HCI studies rely primarily on task performance and subjective self-report data, such as surveys, questionnaires, and interviews. Unfortunately, these methods may fail to fully reflect a user's psychological involvement, cognitive processes, and related emotional arousal (Yao et al., 2014; Mandryk, Atkins, & Inkpen, 2006). Therefore, the crucial problem is capturing how users think and feel while they are actually using the product. This is still a challenge for user experience evaluation studies. For this reason, researchers need to develop new approaches and technologies that enable them to gather data into comprehensive UX evaluation systems.

Researchers may be able to overcome this challenge with the use of a biometric sensor-based measurement system (BMS). These systems make use of technologies, such as eye tracking, electroencephalography (EEG), galvanic skin response (GSR), and facial expression, to automatically detect affective states (Calvo & D'Mello, 2012; Healey, 2014). Recent research has revealed that biometric sensor-based affect detection technologies are an effective indicator of cognitive involvement and emotional arousal and are therefore recommended as a complementary measure of UX (Yao et al., 2014). These high-tech tools let researchers infer emotional and cognitive states by capturing physiological and behavioral signals, such as electro-dermal activity, heart rate, eye tracking, and facial expressions (Calvo & D'Mello, 2010). Capturing these user dynamics provides researchers with uninterrupted and nonintrusive data about users' behavior and reactions during a real-time experience (Picard, 2010). Unlike the other tools used in traditional approaches, one of the most important benefits of biometric sensor-based measurement systems is the ability to capture physiological, emotional, and cognitive dynamics while users interact with the product. Traditional UX approaches cannot accomplish this goal because users may not explain accurately or recall details when asked after the product experience (Ward & Marsden, 2003).

Furthermore, UX evaluations become more efficient when complimentary measures are combined and synchronized. By combining sensors in an integrated system, a BMS allows researchers to collect and analyze physiological and behavioral data. This information helps in developing a more detailed and holistic understanding of factors impacting human performance. Thus, biometric multi-sensor-based affect detection technologies measure psycho-physiological, less conscious, and less voluntary signals of affective states and may provide a more direct and objective inference into those user insights (Nijboer, van de Laar, Gerritsen, Nijholt, & Poel, 2015). The next section of this chapter will discuss biometric affect detection technologies, the types of data they provide, and research applying these technologies to user experience in more detail.

Affect detection on the Domain of Affective Computing

This section focuses on fundamentals, technologies, and applications of affect detection. After emphasizing the significance of affect detection on the domain of affective computing (AC), this section provides detailed information about the most common sensor-based affect detection technologies, namely eye tracking, electroencephalography (EEG), galvanic skin response (GSR), and facial expression analysis. Finally, the chapter concludes with evidence of sensor-based affect detection models on UX evaluations.

Significance of Affect Detection

In psychology, affect is highly dynamic in response a stimulus as a result of neural activity. Additionally, affect is intimately linked to human cognition and human needs (Picard, 1997). It impacts rational decision making and action selection (Picard, 2010). Although individual affect fluctuates based on several stimuli (just as emotions can be expressed variably from person to person), analyzing electro-physiological neural changes in human body and identifying patterns related to a specific emotion can detect affective states. This process requires a particular computing branch, AC, in which researchers can recognize, interpret, process, and simulate human affects. AC is a burgeoning interdisciplinary field incorporating psychology, neuroscience, education, computer science, engineering, and related disciplines. The ultimate goal of AC systems is to enhance the quality of interaction by making a computer interface more usable, pleasant, and operational. This can be accomplished by automatically detecting and reacting to a users' affective states during the HCI processes (Calvo & D'Mello, 2010). For example, an affect-sensitive learning system that detects and reacts to users' frustration is anticipated to increase motivation and advance learning gains when compared to a system disregarding user affect (D'Mello et al., 2008). The focus of research in the field of AC ranges from theories on how affective dynamics influence interactions between a user and a product to how affect detection can be improved to increase efficiency, performance, and analysis of systems that invariably possess affect at their essence (Calvo, D'Mello, Gratch & Kappas, 2014).

Note that affect detection should focus on sensing users' affective states, so that an affect-sensitive system can act precisely. However, this is a very challenging process, since emotions are abstract concepts that cannot be measured in a straightforward manner. In fact, affect detection is not a simple process that can be run quickly and efficiently all the time, but it is rather a potentially effective option to establish a balance between emotion and cognition in technologies which address human factors and user experiences in human systems engineering. Additionally, huge variations in expression and experience exist, and these vary from person to person or mood to mood (Calvo & D'Mello, 2010).

Although affect detection is fundamental to a user-centered design experience (inducing cognition, perception, learning, communication, and decision making), UX researchers have mostly discounted emotion due to the challenges in detection, comprehensive analysis, and inferrence of the affective measurement. Fortunately, the significance of acknowledging the emotional segments of HCI in constructing the field of AC has been on the rise for more than twenty years (Baker & Ocumpaugh, 2014). Developments in affect-sensitive systems have been progressing to provide benefits for several domains, varying from educational technologies to mental health (Calvo & D'Mello, 2010). Although most AC researchers have traditionally remained skeptical about the diverse theories of emotion, the affective technologies being developed are based on with theoretical assumptions that impact their effectiveness. Therefore, a well-versed and cohesive inspection of emotion theories from multiple fields will be apparent if effective real-world models are to be accomplished. Merging theories of emotion with real world engineering objectives to develop affect-sensitive systems entails a review of the literature that examines engineering objectives within psychological attitudes (Calvo & D'Mello, 2010). To this end, this section aims to provide brief information about the most common affect detection technologies and applications. This is followed by a discussion of potential contributions and impacts on research where UX, cognition, and affective domains overlap.

Sensor Based Affect Detection Technologies

A literature review shows that biometric affect detection technologies are described as either *confirmatory* to confirm or validate the traditional measures or as *complementary* to complement the traditional UX approaches that could not detect users' underlying psycho-physiological states (Kula, Atkinson, Branaghan, & Roscoe, 2018). The following paragraphs first describe the essential sensor-based affect detection technologies and then provide evidence of their contributions in terms of confirmatory and complementary UX evaluations.

Eye-Tracking. Usability and design have strong impacts on user satisfaction. Detecting design and usability problems by integrating eye tracking into the product development cycle and solving problems based on the data gathered from users' affective states could improve user experience. Therefore, the UX researchers in HCI utilize eye-tracking technologies to provide persuasive and objective data related to visual attention that help them analyze the usability problems and also enhance usability and comprehension. Primarily, these studies use eye-tracking technology to evaluate interfaces and advance UX design (Nielsen & Pernice, 2010).

In eye-tracking technologies, corneal reflection and pupil dilation are captured using infrared cameras directed at a participant's eyes (Goldberg & Wichansky, 2002; Duchowski, 2007). Gaze movements (fixations and saccades), blinks and pupil dilation variables are collected to interpret visual attention, engagement, emotional arousal, drowsiness, and fatigue (Albert & Tullis, 2013). The eye-mind hypothesis is the basis of eye-tracking research and holds true in experimental scenarios when participants are looking at visual stimuli (Just & Carpenter, 1980). The hypothesis states that what people are gazing at is what they are currently thinking about (Just & Carpenter, 1980; Goldberg & Wichansky, 2002; Poole & Ball, 2006). Although the hypothesis does not always work in real life, people typically pay attention to and think about what they are looking at (Nielsen & Pernice, 2010). The eye-mind hypothesis holds true in experimental scenarios when participants are looking at visual stimuli.

The development of eye-tracking technologies can be grouped into four developmental eras (Rayner, 1998; Duchowski, 2007). The first era was the discovery of essential eye movement variables, such as saccadic suppression, saccade latency, and the size of the perceptual span (ca. 1879–1920). Then eye-tracking technologies were explored with a more applied research focus (ca. 1930–1958). In the third era, advances in eye movement recording systems provided accurate measurements as well as ease of recording (ca. 1970–1998) (Rayner, 1998). Finally, in the fourth era, digital video-based combined pupil/corneal reflection techniques, supplemented by computer vision techniques and Digital Signal Processors (DSPs) were widely used (Duchowski, 2007). While the first two eras were highly intrusive, the recent advances provide the assessment of the proposed Point of Regard (POR) without intrusion (Duchowski, 2007).

Confirmatory Eye-Tracking Research in the UX Domain. Researchers integrated eye-tracking technology into traditional website usability tasks to provide additional evidence on overall usability. For example, Wolpin et al. (2014) conducted a study to develop and optimize some user interfaces specialized for both patients' and caregivers' needs. In this study, the researchers used a traditional usability method, specifically the think-aloud method, and gathered eye-tracking data to identify usability problems. It was found that naming of the navigation buttons was the most critical problems for subjects with a lack of Internet experience. The eye-tracking data confirmed this finding by providing the gaze plot pathways (Wolpin et al., 2014). The study also showed how sensor-based affect detection systems could be incorporated beneficially for usability studies.

Another study integrating eye-tracking methodology to enhance and validate traditional UX measures concerned end user development (EUD) research (Tzafilkou & Protogeros, 2017). Researchers aimed to explore the correlation between end-users' perception and acceptance attributes and reflected eye behavior during the interactions on web-based EUD systems. The results revealed significant correlations between eye behavior and acceptance and perception, so the study showed that both traditional EUD measures and eye-tracking variables are confirmatory for the reliability and accuracy of user perception and acceptance on UX researches.

Complementary Eye Tracking Research in the UX Domain. The evidence revealed by using eye tracking went beyond the validation of traditional UX measures and provided some more empirical data as well. For example, Caroux, Le Bigot, and Vibert (2013) proposed a study investigating how motion and background visual complexity effects players' performance in game interfaces. Specifically, the researchers aimed to assess how radial and/or lateral motion of patterned backgrounds effects users' response times and gaze switches during visual searching. Participants played the same video game several times, trying to get a high score by playing as fast as possible. Their game performance was measured using the average time needed to hit the target as the dependent measurement. Researchers also analyzed fixation durations and gaze movements. The data obtained from fixation durations were more precise in understanding the background complexity than task performance itself. This finding suggests that empirically measured eye-tracking data, such as gaze analysis, may provide more reliable evidence about the mechanisms involved in human performance than measurements from the traditional usability methods in this field.

Likewise, a recent usability study investigated usability testing on websites performed on a target audience that are multilingual by using think-aloud (TA) techniques and eye-tracking technology (Sivaji & Ahmad, 2014). It was shown that using TA technique alone may not be good enough to provide precise and valid data in usability studies in some multilingual environments; however, with the significant correlation and consistency between the TA feedback and the eye-tracking analysis, researchers showed that one website had more navigational efficiency than the other.

Similarly, in another usability study, Loyola et al. (2014) proposed a model to identify key objects on a website. The researchers examined visual gaze activity, such as fixation time and changes in pupil dilation, to investigate the relationship between pupil dynamics and user preferences on a web page. Their findings supported the hypothesis that the integration of pupillary activity analysis allows researchers to get better object classification, specifically yielding a 14% increase in overall accuracy (Loyola et al., 2014). Eye-tracking technology has been used to discover users' interest areas in interfaces. Knowing the users' needs and interests is highly beneficial for the customization of current web systems as well as keeping up users' engagement and attention.

As a result of these studies, it is revealed that the application of eye-tracking technology is highly effective for the evaluation of UX systems, since it provides objective, precise and non-intrusive data to researchers intending to validate a user's visual attention. However, note that eye tracking alone doesn't provide specific information about all of the cognitive processes and emotional states that initiate eye movements. Besides, qualification and experience of researchers should also be considered for an error-free experiments and analysis. Therefore, harmonized eye tracking with other biometric sensors technologies or other approaches for comprehensive UX evaluations are strongly suggested.

Electroencephalography (EEG). EEG is a portable non-invasive neuroimaging technique having high temporal resolution that records the brain's electrical activity over a period using multiple electrodes positioned on the scalp (Teplan, 2002; Niedermeyer & da Silva, 2005). Synchronized electrical impulses from several neurons, communicating with each other according to what the subject is doing and feeling, produces brain signals. These synchronized activities of large groups of neurons create an electrical field that is quite enough to be detected from outside the skull (Lee & Tan, 2006). The signal gathered from electrodes is delivered to a differential amplifier, and the frequency of consequential EEG signals shows voltage changes over time.

By using EEG signals, researchers can observe and evaluate five main types of human brain waves: beta, alpha, theta, delta and the lesser-known gamma. Although EEG technology allows researchers to gather data on five different kinds of brain waves at the same time, only one particular brain wave is dominant at a time depending on the user's state of consciousness. Additionally, brain waves can change rapidly based on personal activity and feeling. Taken together, specific patterns of electrical activity have been associated with emotional states, for example, frustration, cognitive load, and engagement (Albert & Tullis, 2013). The different frequency of EEG signals have been linked to changes in affective states, such as engagement, excitement, and frustration in addition to cognitive functions, such as cognitive load changes due to the different tasks including creativity, problem solving and hands-on activities (Mühl, Heylen, & Nijholt, 2014).

Likewise the contributions of the eye-tracking section above, the following EEG subsections purposefully focused on recent examples of confirmatory or complementary EEG studies for the traditional UX evaluations.

Confirmatory EEG Research in the UX Domain. The studies showed that EEG sensor-based technology was an effective method to gather quantitative affect data on UX evaluations. For instance, the study about the validation of EEG for UX evaluations done by Berka et al. (2007) had a remarkable impact. The researches investigated the utilization of two cognitive states—task engagement and mental workload—for an effective and productive work environment design. In their study, the researchers conducted several cognitive tests consisting of "grid location," "forward digit span," "mental arithmetic," "backward digit span," "trail making," "vigilance," and "image-learning and memory" tests. Each test targeted a different ability. For example, while "trail making" focused on users' spatial memory, the "mental arithmetic" test measured working memory and executive function. Although each test had its own mission, they also shared common features. Each test had subtasks that varied from three to six and had levels of increasing difficulty from easy to difficult. While the users performed the given tasks, the researchers recorded engagement and cognitive load data using EEG sensor-based technology. This provided nondestructive, quantitative data. The researchers also validated the EEG data by correlating the objective task performance data and then by computing the correlations of EEG-based engagement and workload data with the results of subjective reports obtained from each user.

Besides revealing the significantly correlated EEG sensor-based quantitative engagement and workload data with both subjective and objective performance variables, this study also showed three more significant pieces of evidence to support the correlation of users' engagement and mental workload in vigilance, learning, and memory tasks. First, it was observed that users' engagement level declined over the vigilance test. Second, users' workload levels increased when difficulty level increased in grid span, mental addition and both of the digit-span tests. Third, while users were encoding verbal and image stimuli in learning and memory tests, engagement and workload levels of the users were greater than in the recognition process. Also, engagement and workload levels of users increased as a function of task difficulty. According to the results of Berka at al. (2007), users' engagement levels can be captured using EEG sensor-based technology during various UX evaluations. The flow of engagement changes according to the amount of information, visual processing, and attention. Similarly, the level of users' workload is correlated with the performance of working memory load and can be higher in some specific activities, such as decision-making, critical thinking, and problem solving, while performing UX evaluations.

Following the studies focused on the validation of EEG in UX evaluations done by Berka et al. (2007), some studies compared traditional usability methods with EEG sensor-based methods. For example, Lee and Seo (2010) compared traditional usability testing to sensor-based usability testing, particularly electroencephalograph (EEG) and electrocardiogram (ECG). The researchers measured the affective state dynamics of users while examining the design of four web pages. In their study, they found similar results from both methods. This evidence showed that the sensor-based usability testing was as reasonable and valuable as the traditional usability testing.

Similarly, Masaki et al. (2011) compared the traditional usability method, the questionnaire, with the EEG sensor-based usability method. All participants completed different tasks using Excel 2007 software. Researchers gathered the EEG values of the participants, particularly alpha and beta waves, and used the alpha/beta ratio to assess the affective states of participants after each task. Participants also filled in a questionnaire about their experience with the given software at the end of each task. The questionnaire aimed to measure user satisfaction and how often the users used each function in Excel 2007. The analysis of the study revealed significant correlations between alpha/beta ratios and each item on the questionnaire. This analysis also demonstrates that the EEG sensor-based technology was a validated method for UX evaluations, as are traditional methods.

Complementary EEG Research in the UX Domain. Like the eye-tracking technology, application of EEG on UX evaluations provided more confirmation of traditional UX approaches. Ghergulescu and Muntean (2014) critiqued the idea that questionnaires are adequate to gather reliable data while users perform interactive tasks in a game-based learning activity. They claimed that questionnaires disrupted players' motivation and engagement. As a solution, they designed a study using EEG sensor-based affect detection, which provides real-time non-intrusive quantitative data collection for investigating the users' affective states while they are learning by playing a game. The researchers associated the EEG methodology with the traditional questionnaire method to make a comparison. The comparison of the two methodologies was remarkable. The traditional questionnaire had some limitations in analyzing users' motivation while playing the short game, and it was not effective enough to analyze users' total motivation over long game-playing sessions. On the other hand, the EEG sensor-based methodology was able to analyze users' motivation in both cases of the gameplay sessions without disturbing the users' interactions or their engagement and motivation. The evidence from this study shows that EEG sensor-based affect detection methodology is superior to traditional questionnaire-based methodologies.

Similarly, a recent study compared the usability of three different EEG headsets: particularly dry, solution-based, and gel-based (Grummett et al., 2015). Each participant used all three headsets for a task to measure the accuracy of each headset. In addition, by using questionnaires, researchers collected the user responses regarding the perceived efficiency and user satisfaction to measure the subjective usability of the EEGs. Although the gel-based EEG provided the highest accuracy, and the solution-based one provided both the lowest accuracy and the lowest level of comfort, the subjective self-responses indicated that the solutions-based EEG was the most preferred. The reason for this preference was linked to aesthetics evaluation of EEGs, where solution-based EEG was ranked the best. However, when considering all aspects of the evaluations, researchers suggested the use of the gel-based EEG because of its high accuracy and efficiency. The conclusion about the best usable EEG would be totally different if only the traditional self-report approach were considered. Fortunately, the results gathered using EEG provide complementary evidence to guide researchers to the correct conclusion.

Based on evidence from these studies, it is posited that the application of sensor-based affect detection using EEG is preferable for the evaluation of UX systems, since it provides quantitative, non-intrusive data for researchers intending to validate user opinions and identify the level of user satisfaction with a particular system. However, note that some researchers are skeptical about the reliability of EEG technology due to the poor spatial resolution of this technology (Mühl, Heylen, & Nijholt, 2014). For example, during EEG data collection, it is obvious that the user blinks, twists, and glances while doing repetitive tasks during the studies. These blinks, twists, and glances may cause some electrical noise or artifacts that may be difficult to separate from neural activity (Turnip & Kusumandari, 2014). Also, the recorded electrical potentials vary broadly from user to user due to variations in such tissues as brain matter, blood, and bones (Zurawicki, 2010). Therefore, it is crucial to remove all data containing artifacts or noise by using noise cancelation methods on the EEG records. Independent component analysis, neural networks, Kohonen maps, and PCA are some techniques to clean the data for that purpose (Turnip & Kusumandari, 2014). Moreover, combining EEG technologies with the other sensor-based technologies, such as eye tracking, GSR, or facial expressions, helps to detect and remove the artifacts.

Galvanic Skin Response (GSR). GSR measures changes in skin conductance due to sweating. Once sweating levels increase, skin conductance increases. In biosensor technology, increase in skin conductance as a response of involuntary sweating is measured to infer the increase of emotional arousal, engagement, and congruency of self-reports (Boucsein, 1992; Damasio, 1994; Cohn & De la Torre, 2014). Moreover, the stress level in studies on anticipatory anxiety and stress during task performance was measured by using GSR (Boucsein, 1992). The following sub-sections present some samples from literature focused on confirmatory and complementary functions of GSR for the UX evaluations.

Confirmatory GSR Research in the UX Domain. Like the eye-tracking and EEG literatures, comparison and validation research was essential to make sure that GSR provided valid variables related to the user's emotional arousal and stress levels while using and examining any given product or service. Lin, Omata, Hu, and Imayima (2005) examined users' experiences during a video game that required quick reflexes and skills. The researchers tested stress levels during activities such as avoiding an aggressor while simultaneously completing other objectives or navigating difficult terrain without falling. Galvanic skin conductivity through GSR and self-reported perceived stress patterns were measured. The study revealed correlated results—while easy tasks caused low amount of stress, stress increased in harder tasks.

Likewise, Foglia, Prete, and Zanda (2008) conducted a study focused on a comparison of traditional usability methods and GSR sensor-based affect detection on UX evaluations. A government web page providing an animated face to facilitate user interactions and navigational guidance was used to investigate the efficiency of the web page and, in particular, the impact of the animated face. Group 1 used the web page's traditional user interface, while Group 2 used the page enhanced with the animated face. Users in both groups watched a relaxing clip first and then performed two different tasks.

At the end of each task, users rated their mental effort. As a last step in the experiment, users filled out a questionnaire based on a 5-point Likert scale. Researchers gathered the GSR of each user during all experiment sessions. The GSR analysis quantitatively revealed that users had higher arousal and emotional interactions in the page having an animated face, because the animated face decreased the number of navigational interactions and the amount of mental effort. This evidence was also supported by the results of the traditional usability test. Thus, the study provides remarkable evidence to show the consistency across traditional usability approaches and sensor-based affect detection approaches, as well as to highlight the feasibility of sensor-based affect detection technologies in UX evaluations.

Similarly, another study aimed to combine self-reported Valence-Arousal (VA) ratings and GSR to investigate how multiple data sources contribute to the identification of specific stress region(s) in the VA space (Liapis, Katsanos, Sotiropoulos, Karousos, & Xenos, 2016). Participants performed five stressful tasks, and GSR data was collected during this performance. After each task, participants articulated their perceived emotional experience using the VA rating space. The study revealed which regions in the VA rating space may reliably show (from 60%-85%) self-reported stress that is in alignment with one's measured skin conductance; thus GSR confirmed the VA rating space.

These sample studies point out that GSR-based affect detection systems provide reliable quantitative data about human emotions and are consistent with traditional UX methods. GSR can be reliably used as a confirmatory method to validate the results obtained from traditional usability methods in UX evaluations.

Complementary GSR Research in the UX Domain. Mirza-Babaei et al. (2011) attempted to measure the impact of GSR as a confirmatory method to traditional usability methods. This research specifically used observation-based user testing methods for the production processes of the video games. The researchers chose six qualified participants who were familiar with video games but lacked experience with the studied games. Participants played two different games using the same game platform. Researchers gathered real-time data using GSR technology. After conducting two separate usability tests and experience evaluations for the gameplay video recordings, researchers analyzed the data in two different ways.

First, they examined sensor-based analyses and observation-based analyses. In sensor-based analysis, the exact moments of gameplay that caused peaks in a user's GSR graph were replayed after the game playing session using video recordings. At that point researchers asked the players to express their opinions. This study allowed researchers to collect players' inferences related to their specific affective states. Researchers noted players' responses and all notes were evaluated to uncover usability problems with the games.

Second, in the observation-based analysis, two experienced researchers analyzed the videos recorded during sensor-based analysis. Researchers determined the problems based on the players' interactions in video recordings. The analysis of their study revealed that 29 of 89 (32.6%) issues were identified by both GSR and observation-based approaches. Thirty-four issues (38.2%) were reported only during observation-based user testing sessions, while 26 (29.2%) issues were established only from the sensor-based user testing sessions. As a result, the authors concluded that, although observation-based techniques could fail to identify all of the problems regarding usability, it succeeded in detecting most of them. In this case, sensor-based affect detection technology empowered UX studies to uncover hidden problems related to users' affective states. This evidence is remarkable but also requires further research, since the researchers recruited only six subjects to participate in the study and used only one sensor-based technology rather than multimodal sensors.

Likewise, Yao et al. (2014) questioned whether the data gathered by sensor-based affect detection technologies differs from traditional task performance data while using mobile phones, and if so, how that data would correlate with self-report data in UX. To address these questions, the researchers designed an experiment to test the feasibility of affect detection in UX evaluations. Twenty subjects completed five tasks, such as finding a place and booking it by using a searching and booking app on the mobile phone. They also reported the difficulty level of the task using a five-point Likert scale following each task. After completing all five tasks, participants reported their satisfaction using a questionnaire. Researchers gathered subjects' affective responses using GSR technology, task performance, and self-report data. The experiment revealed that GSR values differed with task performance. In fact, changes in GSR when subjects failed at a task were higher than when they succeeded. The study additionally found significant correlations between GSR and subjective self-assessment of user experience when examining attractiveness, efficiency, dependability, and novelty. The results of this study verified the overall impact of data gathered by sensor-based affect detection technologies on UX evaluations.

As in the eye-tracking and EEG section above, these recent studies point out that GSR-based affect detection systems provide reliable quantitative data about emotional arousal, and it is correlated with traditional UX methods, though it lacks emotional arousal data. GSR technologies have been used as a confirmatory for some of the traditional usability methods in several settings to understand users' affective responses, or as a complementary method for users to confirm the results obtained from traditional usability methods in UX evaluations.

Facial Expression Analysis. Facial expression analysis records facial movements on a moment-to-moment basis, detecting muscle groups in action during different emotional responses such as smiling, crying, and expressing disgust (Ekman & Friesen, 1978, Cohn & De la Torre, 2014). For example, furrowing of the eyebrows might indicate anger or concentration (Picard, 2002), which can be disambiguated by other muscle movements (for example, furrowed eyebrows combined with narrowed eyes and a clenched jaw can signal anger). Historically, these analyses were conducted by humans who were extensively trained to detect action units (Ekman and Friesen, 1978), but modern tracking systems and software can perform these tasks in an automated fashion (Cohn & De la Torre, 2014). Facial expressions could be an effective instrument to understand user emotions and responses (Ramsey, 2014). Researchers utilize facial expressions to comprehend emotional reactions, since the face reveals both conscious and non-conscious responses, implying a strong link between facial features and affective states (Bosch, Chen, & D'Mello, 2014).

Confirmatory Facial Expression Analysis Research in the UX Domain. Facial expression analysis has been used as a confirmatory for some of the traditional usability methods in several settings to understand users' affective responses or as a complementary method for users to confirm the results obtained from traditional usability methods in UX evaluations. He, Boesveldt, Graaf, and Wijk (2015) questioned if emotional responses to food stimuli reveal information about consumers' eating behavior. In the study, researchers used two food odors, orange (pleasant) and fish (unpleasant). During the sessions, researchers recorded the emotional responses to these odors in 26 participants. Participants stated their responses using non-verbal subjective reports at the end of their session. The results of the non-verbal reports indicated that the orange odor (pleasant) usually caused positive emotional responses, such as joy, satisfaction, and hope. On the other hand, the overall responses to the fish odor (unpleasant) were mostly negative emotions, such as dissatisfaction, fear, and disgust. The correlation between non-verbal individual reports and facial expressions was greatest after two seconds ($r = 0.97$). In addition, researchers linked the pleasant odors with neutral and surprised expressions and with fewer expressions of disgust. Also, more intense odors were associated with more expressions of disgust and fewer neutral expressions. The study revealed that, while non-verbally self-reported emotions were rather one-dimensional reflecting the odor's valence, facial expressions exposed both the odor's valence and other aspects, such as intensity. Moreover, facial expressions produce data to allow researchers to monitor sequential appraisal of emotions, which deliver more insight about the users' initial and subsequent behaviors.

Complementary Facial Expression Analysis Research in the UX Domain. Whitehill et al. (2011) focused on the relevance of users' learning performance to their facial expressions. Particularly, they questioned if an occurrence of a smile implies mastery. They designed an experiment providing game play to users, and recorded students' facial expressions using Computer Expression Recognition Toolbox (CERT), a software that detects fully unconscious, real-time facial expression recognition. Surprisingly, the study revealed that subjects who learned more also smiled less ($r = -0.34$, $p < 0.05$). Researchers interpreted this data as follows: smiles often happen when someone felt embarrassed rather than when s/he achieved mastery. This conclusion was similar to other evidence showing that smiling often happens during normal periods of frustration (Hoque & Picard, 2011).

Therefore, facial expression analysis technologies have been used as a confirmatory for some of the traditional usability methods in several settings to understand users' affective responses, or as a complementary method for users to confirm the results obtained from traditional usability methods in UX evaluations, since they provide reliable quantitative data set about emotional valence, and it is correlated with traditional UX methods. However, it does not collect any data related with the emotional arousal data, so that it is recommended to both GSR and facial expression analysis to get enriched data for emotional arousal and valance.

The Impacts of Affect Detection Technologies on UX

This section of the background literature discusses the impacts of sensor-based affect detection technologies on UX by using the findings of research discussed in previous sections and the present author's in-house UX lab experiences related to the utilization of the multisensor-based affect detection technologies to enhance UX evaluations. Data related to users' cognitive processing and affective states provide crucial information and evidence for identifying usability issues and increasing user satisfaction (Gaver & Martin, 2000; Hassenzahl & Tractinsky, 2006). Recent research has revealed that sensor-based affect detection systems provide reliable, quantitative data about human emotions, and is correlated with traditional UX methods, such as subjective and objective performance (Berka et al., 2007; Foglia, Prete and Zanda, 2008,;Mirza-babaei et al., 2011; Wolpin et al., 2014, Masaki et al., 2011).

Table 2: The Impacts of Sensor-Based Technology on UX

Impact	**Eye tracking**	**EEG**	**GSR**	**Facial Expression Analysis**
Confirmatory (Verification and validation)	Wolpin et al., (2014) Tzafilkou & Protogeros, (2017)	Berka at al., (2007) Lee and Seo (2010) Masaki et al. (2011)	Foglia, Prete and Zanda (2008) Liapis, Katsanos, Sotiropoulos, Karousos, & Xenos, (2016)	He, Boesveldt, Graaf and Wijk (2015)
Complementary (Enhancement of Findings and/or more precise and valid evidence)	Caroux, Le Bigot and Nicolas Vibert, (2013) Loyola et al., (2014) Sivaji & Ahmad, (2014)	Ghergulescu & Muntean (2014) Grummett et al., (2015)	Mirza-babaei et al. (2011) Yao et al. (2014)	Whitehill et al. (2011)

The contributions of sensor-based affect detection systems to the UX can be categorized in two groups: (1) Confirmatory and (2) Complementary as summarized in Table 2. First, several studies in the literature showed that they were used as a confirmatory method to validate the results obtained from the traditional usability methods in the UX evaluations. Second, sensor-based affect detection systems were used as a complementary method. They reveal remarkable evidence to show where traditional usability approaches fail. They provided comprehensive findings to uncover the issues related to mental and physiological pathways to enhance the design of product and services. Therefore, this proposal claims that it is necessary to integrate sensor-based affect detection technologies, which enable researchers to confirm or validate their traditional UX studies or complement where the traditional UX approach could not detect for the comprehensive UX evaluation systems.

Biometric Measurement Systems

Biometric Measurement Systems (BMS) recognize and visualize the sequential deviation of psychophysiological signals that contain the information about cognitive and affective states related to behaviors, cognitions, and emotional underlying human learning and performance in complex settings.

Equipment for the BMS. BMS are scalable platforms including several types of biosensor sources. Eye tracking, facial expression analysis, EEG and GSR are four common types of sensors. Eye tracking technology captures corneal reflection and pupil dilation using infrared cameras directed at participants' eyes. Gaze movements (fixations and saccades), blinks and pupil dilation can be used to assess participants' visual attention, engagement, emotional arousal, drowsiness, and fatigue.

Next, the activity of facial muscles are used for facial expression analysis. Position and orientation of the head and facial landmarks, activation of action units (AUs) and variables from emotion channels can be used to interpret the emotional valence, engagement and validity of self-reports.

EEG technology measures changes in the brain's electrical activity using gel-based electrodes on the scalp. Event-related potentials, wavelets and frequency band power variables are analyzed to interpret cognitive and emotional constructs such as workload, engagement, distraction and drowsiness.

GSR measures changes in skin conductance due to sweating. In this biosensor technology, skin conductance response is used to infer the intensity of emotional arousal, engagement, and congruency with self-reports.

Table 3: Biometric Tools

Tools	What is measured?	How is measured?	Which variables are derived?	How is the data interpreted?
Eye Tracking	Corneal reflection & pupil dilation	Infrared camera pointed towards eye	Eye movements (Fixations and saccades), blinks, pupil dilation	Visual attention, engagement, drowsiness & fatigue, emotional arousal
EEG	Changes in electrical activity of the brain	Electrodes placed on scalp	Event-related potentials, wavelets, frequency band power	Engagement, distraction, drowsiness, workload
GSR	Changes in skin conductance due to sweating	Electrodes attached to fingers, palms or soles	Skin conductance response	Emotional arousal, engagement, congruency of self reports.
Facial Analysis	Activity of facial muscle & muscle groups	Webcam pointed towards face along with computer algorithms for feature extraction	Position and orientation of head & facial landmarks, activation of action units (AUs) & emotion channels	Emotional valence, engagement, congruency of self-reports

These four different sensors are used simultaneously with the BMS, synchronized and visualized on a graph. Then, they are stored in the database, saved with the study, and exported for further analysis. Table 3 summarizes the biometric tools, their functions and the aim of use in the proposed study.

Benefits of BMS. Each biosensor module can effectively reveal particular aspects of users' behaviors, cognition, and emotional response, though each has limitations as seen in Table 4. Although eye tracking allows researchers to detect where, when, and what a person is looking at, tracking eye movements alone does not provide enough information, particularly about the cognitive processes and emotional states that drive eye movements. EEG activity, which cannot be controlled consciously, helps to infer the global emotional state of a user but does not provide precise data about the intensity of that emotion (i.e., arousal). Whereas facial expression analysis assesses the valence of emotion, it cannot evaluate the power of that emotion (arousal). GSR is a highly reliable and efficient measure to analyze emotional arousal, but it does not reveal the emotional valence or the quality of the emotions. Together, these four tools contribute to a holistic understanding of cognitive and physiological processes in UX.

Table 4: The Essential Benefits and Limitations of the Sensors

Sensors	Benefit	Limitation
Eye Tracking	Visual attention	Cognitive processes and the emotional states that drive eye movements
EEG	Global emotional state of a user	Emotional arousal
Facial Expression	Emotional valence	Emotional arousal
GSR	Emotional arousal and stress	Emotional valence

By combining sensors in an integrated system, BMS allows researchers to collect and analyze cognitive, affective and behavioral data to develop a more detailed and holistic understanding of factors limiting human performance. For instance, combining

EEG and GSR measurements connects the valence picked up by EEG with arousal derived from GSR. Likewise, combining EEG and facial expression analysis captures variations in emotional states across time alongside a real-time sequence of emotional expression. Combining eye tracking and EEG allows researchers to identify interest produced by particular stimuli during variations in workload. Eye-tracking delivers information about the exact orientation of the eyeball, enabling researchers to identify artifacts, such as blinks and saccades, and decontaminate the EEG data.

Table 5: Source Distributions of Inferred Data

Inferred Data	**Eye Tracking**	**EEG**	**GSR**	**Facial Analysis**	**Verbal Report**	**Performance report**
Visual Attention	+				+	
Engagement	+	+		+	+	
Distraction		+			+	
Drowsiness	+	+			+	
Fatigue	+				+	
Emotional Arousal	+		+		+	
Emotional Valence				+	+	
Workload		+			+	
Confusion				+	+	
Frustration				+	+	
Joy				+	+	
Anger				+	+	
Surprise				+	+	
Fear				+	+	
Contempt				+	+	
Sadness				+	+	
Disgust				+	+	
Excitement			+		+	
Stress			+		+	
Completion Time						+
Error Rate						+

The proposed BMS provides non-intrusive real-time data on human cognition, emotion, and behavior. Table 5 shows the distribution of variables from biometric devices and surveys. By combining biosensors, researchers can collect richer and more accurate insights into cognitive and behavioral processing.

Limitations and Challenges of BMS. By combining sensors with an integrated system, BMS allows researchers to collect and analyze both physiological and behavioral data and to develop a more thorough and holistic understanding of factors limiting human performance. However, based on in-house lab experiments, using these systems takes additional time and requires research expertise because multiple channels must be combined, analyzed, and interpreted. A well-trained research team is a keystone to managing data collection as well as to data cleaning, analysis and interpretation of outputs. Moreover, some project requirements include advanced statistics skills, including data mining, and time series analysis. Unpredictable issues during data collection may additionally occur, though the manufacturers may guarantee that systems should work without any issue under normal conditions. These issues vary widely and may be related to the conductivity of the sensors, software crashes, or other technical issues. Potential issues may arise from low battery status, limited data storage capacity, or software bugs. Once a problem occurs, quick resolution may be a challenge. Typically, researchers contact the manufacturer support team to resolve issues as soon as possible, but a resolution may not be readily available. Therefore, being a problem-solver, both in the lab and in manufacture side, is required to overcome potential issues.

Running a pilot test is an important milestone in BMS experiments. Pilots sessions are good indicators of if the proposed design works efficiently in practice.

Further, they give ample opportunity to optimize sensor settings before the actual data collection. It is strongly suggested to implement any changes based on the collection of revisions and suggestions during the pilot phase.

Another challenge is the cost of BMS technologies. When compared with the cost of traditional usability tools, such as some software and recruitment cost, the cost of BMS tools is extremely high. However, once the potential benefits of all-in-one solutions are considered, they may still be a worthwhile investment, particularly for UX evaluations. Though sensor-based affect detection technologies have significant limitations on their own, collaborative use can reveal more evidence related to user experiences and behaviors. When compared with traditional usability methods and single-sensor affect detection technologies, BMS technologies may provide more accurate data through a wide range of emotions and deliver insights for UX research. Particularly, BMS can (1) quantify and analyze visual attention, as well as reveal how, when, and what people see, with gaze position and pupil dilation quantified through eye tracking; (2) reveal demonstrations of underlying emotional states and identify universal basic emotions and valence through facial expressions analysis; (3) uncover perceptual, cognitive, and emotional processing, as well as measure motivation, engagement, and workload; (4) grasp insights into human arousal and stress, exposing emotional reactions of users and measuring the psycho-physiological arousal of a user through GSR (Imotions, 2015). It may be worth investing in BMS technologies to get meaningful research in UX and HCI.

Overview of the Present Study

The main goal of the study was to analyze the results of multi-sensor affect detection in UX evaluation systems by comparing and combining them with traditional UX methods (self-report). These insights may reveal why some products are preferred more than others, or which features make influence purchasing decisions. Based on the main aim, the present design concerns user behavioral and affective measurements in relation to their online shopping and decision-making processes.

The Design Chart

The study design has three main sections, namely “The User”, “User Decision” and “The Measurement of User Experience”. While “The User” covers human behaviors, thoughts, and feelings, “User decision” refers to the measurable outcomes which will be called “choice” in the following sections. “The Measurement of User Experience” measures human actions and feelings to predict choice. Figure 1 represents the relationship between the three main sections of the experiment.

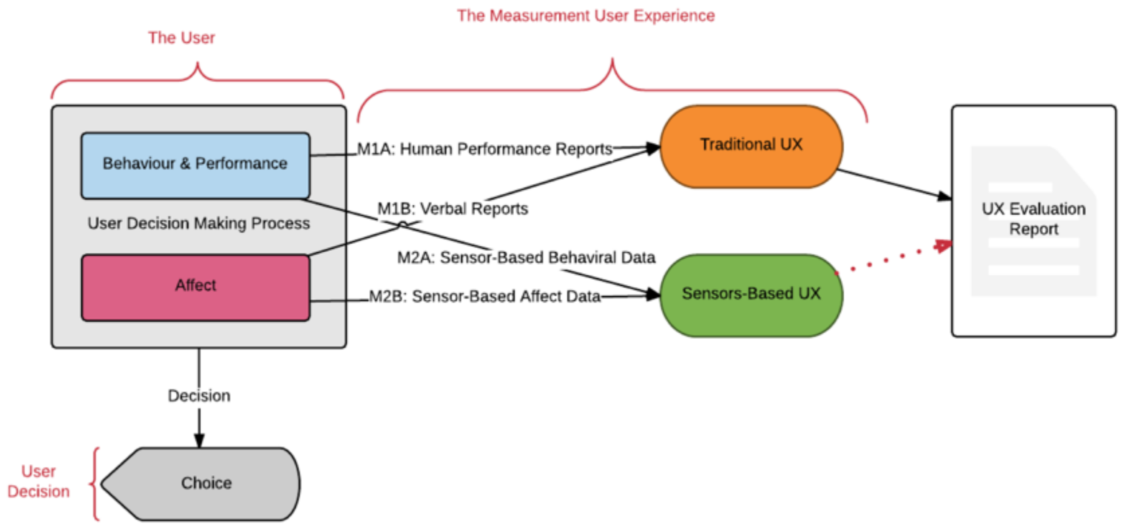


Figure 1: The Relationship of the Experimental Design Sections

In the proposed design, “The User” extends to human decision-making process. User behavioral performance and affective states are two main aspects of the user decision making process. The data relevant to user decision making process will be collected through three main UX methods. These are human performance measurements (M1A), verbal reports (M1B), and sensor-based UX methods (M2). While the first two methods can be seen as traditional UX methods (M1), the third one is an innovative UX method for questions about behavioral, cognitive and emotional user insights. This is a practical solution since it provides non-intrusive real-time data on human cognition, emotion, and actions.

The Research Questions

The research focused on three main research questions to enhance and enrich the user experience evaluations.

1. How closely can a sensor-based evaluation approach corroborate user experience identified by traditional UX methods during an online purchasing task?
2. Which model, whether sensor-based, traditional, or combined, can best be used to explain customer preferences for purchasing?
3. Which sensors, whether separately or integrated, can best be used to explain customer preferences for purchasing?

These questions work as general guidelines for detailed research and extended to sub-questions in the following sections, which explain the context of each question.

1. *RQ1: How closely can a sensor-based evaluation approach corroborate user experience identified by traditional UX methods during an online purchasing task?*
    a. Does a sensor-based UX evaluation confirm or elaborate on traditional UX methods during an online purchasing task?
    b. Does a BMS produce different results than traditional UX methods in an online purchasing task?

This literature review shows that the contributions of sensor-based affect detection systems to the UX studies can be categorized in two groups: First, several studies in the literature showed that they were used as a confirmatory method to validate the results obtained from the traditional usability methods in the UX. Second, sensor-based affect detection systems were used as a complementary method (Kula, Atkinson, Branaghan & Roscoe, 2018). They reveal remarkable evidence to show where traditional usability approaches fail.

The RQ1 examined how multi-sensor affect detection systems provide evidence on user experience. It was extended below for more detailed analysis and to clarify any mutually beneficial relationship between traditional UX measurements and sensor-based UX measurements. This allowed for estimates of user behavior as well as cognitive and affective states to reveal user experience. Regarding the RQ1, it is hypothesized that findings of sensor-based UX approach on user experience (UX) evaluation systems are equally or more reliable and precise than traditional UX methods alone during an online purchasing task.

2. *RQ2: Which model, whether sensor-based, traditional or combined, can be best used to explain customer preferences for purchasing?*
    a. How can a sensor-based UX evaluation approach predict user rankings during an online purchasing task?
    b. How can a traditional UX evaluation, consisting of verbal reports and human performance reports, predict user rankings during an online purchasing task?
    c. How can a combined model, including both sensor-based and traditional UX evaluation approaches, predict user rankings during an online purchasing task?

RQ2a inquiries whether a sensor-based UX evaluation is good enough to predict user choice. It aims to reveal that what sensor-based UX measurements are gathered from user behavior and performance (M2a), and how sensor-based UX measures reveal human insights data (M2b). Reversely, RQ2b asks whether a traditional UX evaluation consisting of verbal reports and human performance reports are good enough to predict the user rankings. RQ2b mainly aims to reveal if human performance reports yield clear results (M1a), and how efficient verbal reports are in getting human insights data (M1b). RQ2c asks whether the combined version (including both approaches) is most effective in explaining customer purchasing preferences.

In regard to the RQ2, it is hypothesized that a combination of sensor-based and traditional UX approaches provides more reliable and accurate UX reports than either method used separately to explain customer purchasing preferences

3. *RQ3: Which sensors, whether separately or integrated, can be best used to explain customer preferences for purchasing?*

Finally, RQ3 focuses on assessing the value from the BMS. The aim of RQ3 is to figure out the optimal sensor combinations for UX research, so that researchers can determine the necessity of each sensor type. The contributions of single-sensor and multi-sensor based UX methods on overall UX evaluations systems was analyzed. Further, the present research identified the most effective sensors or combinations of sensors, to characterize and predict customer preferences for purchasing. Variables from several sensors was evaluated to reveal the contribution of BMS to the overall UX evaluation system.

The hypothesis for the RQ3 is that all four sensors (eye tracking, EEG, GSR and facial expression analysis) are required to reveal holistic and precise user affect patterns. It is hypothesized that the efficacy of a multi-sensor approach was higher than single-sensor approach to explain customer preferences for purchasing. Table 6 summarizes all hypotheses and research questions.

Table 6: Hypotheses Related with Research Questions

RQ	Hypothesis
RQ1: How can a sensor-based UX evaluation approach reveal user experience identified by traditional UX methods during an online purchasing task?	**H1:** The impacts of sensor-based UX approach on user experience (UX) evaluation systems equal or greater than traditional UX methods, particularly human performance reports and verbal reports during an online purchasing task.
RQ2: Which model, whether sensor-based, traditional, or combined, can be best used to explain customer preferences for purchasing?	**H2:** A combination of sensor-based UX approach and traditional UX approaches provides more reliable and accurate UX reports than they used separately to explain customer preferences for purchasing.
RQ3: Which sensors, whether separate or integrated, can be best used to explain customer ranking preferences for purchasing?	**H3:** The efficacy of multi-sensor UX approach is higher than single sensor UX approach to explain customer preferences for purchasing?

In the following chapters, the methodology and analysis were provided to investigate these research questions and test the hypotheses.

CHAPTER 3

METHOD

This chapter consist of four main sections. Participant information and overall design is presented. Then, the chapter continues with the explanation of the detailed procedural flow. Third, proposed UX measures from both traditional and sensor based UX approaches are examined. The chapter ends with information about the elements of a proposed biometric sensor suite.

Participants and Design

The study was conducted in three different locations: the Consumer Behavior Lab in the Adidas Group in Portland, Oregon; the ASU iLUX Lab in Tempe, Arizona; and the SLATE Lab in Mesa, Arizona. Forty-eight participants from Portland and the Phoenix Metropolitan Area were recruited in exchange for a modest a financial incentive, which was a $25 gift card, for a 75-minute online experiment session. Participants were female adults ranging from 18 to 35 years old and were able to read and speak English fluently. Participants were randomly assigned to sessions and each session included only one participant and one researcher.

The study had three main conditions, Low-level Support, Medium-level Support, and High-level Support. Each support level included one Brand 1 bra and one Brand 2 bra, as seen in Table 7.

Table 7: The Products and Support Levels

Support Level	Brand 1: Adidas	Brand 2: Nike
Low	**Product 1:** Strappy (#11) 	**Product 2:** Indy Wipeout (#21) 
Medium	**Product 3:** Techfit (#12) 	**Product 4:** Pro Classic (#22) 
High	**Product 5:** Climachill (#13) 	**Product 6:** Pro Rival (#23) 

The order of presentation of these three support levels was counterbalanced to eliminate order as a potential confounding factor, which could negatively affect the results. In addition, the order of the choices in each condition was randomized. Using a within-subjects design, each participant experienced all conditions to decrease the amount of error that might result from variability among participants.

Procedure

The procedural steps were grouped in four sequential sections: (A) Informed Consent and Preparation, (B) Low-Level Support, (C) Mid-Level Support, (D) High-Level Support and (E) Closure. The steps of each procedure section are listed Table 8 in below.

After participants signed the consent form and completed the demographic and attitude survey, they put on the BMS with the guidance of the researcher. They then completed a baseline test. As the final step of section A, a sample webpage was provided to allow participants to familiarize themselves with the main elements of the webpage, such as choice name, price, ratings, comments, products specifications, and product images. Then subjects examined one product from each brand that has the same-support level. After rating each product through a rating survey, they ranked the two products of each support level by filling the decision survey. These steps were repeated until all three support levels were ranked. For the last decision response, subjects ranked all six products from most likely to buy to less likely to buy. Finally, the study was completed after a website satisfaction survey. While participants were examining the websites and responding to the surveys, the researcher recorded their affective and cognitive states, eye movements, facial expressions, and galvanic skin responses through biometrics measurement system consist of four main sensor-based tools.

Table 8: Procedure

Order	Section	Name	Time (Min)
1	A. Intro	Consent Form, Demographic and Attitude Surveys	5.0
2	A. Intro	Wearing Biometrics Tools and Impedance check	10.0
3	A. Intro	EEG Baseline Test	15.0
4	A. Intro	Orientation page	2.0
5	B. Low-Level Support	Web page to examine Adidas choice #1	1.5
6	B. Low-Level Support	Rating Survey	3.0
7	B. Low-Level Support	Web page to examine Competitor choice #1	1.5
8	B. Low-Level Support	Rating Survey	3.0
9	B. Low-Level Support	Decision Survey for Low-Support Condition	2.0
10	C. Mid-Level Support	Web page to examine Adidas choice #2	1.5
11	C. Mid-Level Support	Rating Survey	3.0
12	C. Mid-Level Support	Web page to examine Competitor choice #2	1.5
13	C. Mid-Level Support	Rating Survey	3.0
14	C. Mid-Level Support	Decision Survey for Mid-Support Condition	2.0
15	D. High-Level Support	Web page to examine Adidas choice #3	1.5
16	D. High-Level Support	Rating Survey	3.0
17	D. High-Level Support	Web page to examine Competitor choice #3	1.5
18	D. High-Level Support	Rating Survey	3.0
19	D. High-Level Support	Decision Survey for High-Support Condition	2.0
20	E. Closure	Ranking Survey	3.0
21	E. Closure	Website Satisfaction Survey	5.0
22	E. Closure	Closing the study.	2.0
		Total Exposure Time (min)	**75.0**

UX Measures

The study collected data from two main sources, traditional UX measures and sensor-based measures. Both were used for the inference of users' visual attention, emotional valence, emotional arousal, and affective and cognitive states, such as engagement, distraction, and workload. In addition, traditional sources allow researchers to collect user responses about product ratings, decision to buy one specific product, and product ranking. The following sections, first provide information about the types of traditional measures, then continue with the type of sensor-based measures and their variables.

Traditional UX Measures

Four traditional self-report surveys provided information about both users' behavioral response and their feelings to determine the most and the least preferred choice as well as essential features. These are the presurvey, postsurvey, user decision surveys, and ranking surveys. While the presurvey was composed of user demographics and attitudes, the postsurvey included participants' rating for the examined product, their feelings and their evaluations of the user interface sections after product examination. The user decision survey captured user final decision within each support group. Finally, after examination of all products, user ranked the products from most likely buy to less likely to buy in the ranking survey. The following paragraphs provide more info about each of these traditional UX measures as shown in Appendix A.

Presurvey-Demographics. The demographic survey asked about participants' background to help researchers assess participants' experience level as it relates to online product evaluation and shopping. The questions were about their age, gender, ethnicity and education level. Then the questions continued to get information about attitudes.

Presurvey-Attitudes. The attitude survey investigated user habits and attitudes about online shopping and their sports intensity.

Table 9: The Variables of Presurvey in the Traditional UX Analysis

#	Self-report	Variable	Description
1	Demographics	Everyday bra size	Everyday bra size
2	Demographics	Sports bra size	Sports bra size
3	Attitudes	Internet/day	The number hours using internet/day
4	Attitudes	Online shopping/month	The number of frequency for online shopping/month
5	Attitudes	Weekly sport practice	Frequency of sport practice in a week
6	Attitudes	Daily sport session	Frequency of sport session/day
7	Attitudes	Coach driven rate	Rate of coach driven activities
8	Attitudes	Internet/day	The number hours using internet/day
9	Attitudes	Sports bra frequency	Sports bra use while doing sports
10	Attitudes	Sports bras number	# of sports bras that subjects have
11	Attitudes	Money Theoretical	Money to be willing to spend on a sports bra
12	Attitudes	Money Practical	Money typically speny on a sports bra
13	Attitudes	Product Visuals Attitude	Importance of product visuals for online shopping
14	Attitudes	Descriptions Attitude	Importance of product description for online shopping
15	Attitudes	Ratings and Comments Attitude	Importance of product rating and comments for online shopping
16	Attitudes	Cost Attitude	Importance of product cost for online shopping

Particularly, the frequency of Internet use, the devices that they use for Internet, the significance of shopping website features, such as visuals, ratings, price and descriptions of the products, the frequency of online shopping, and the amount they would pay for these items were some of the examples in online shopping section of attitudes. It also included frequency of sport that participant actively performed, and their favorite sports to interpret their sport intensity. Table 9 below lists the variables and their descriptions that are used for the traditional UX analysis section of study.

Postsurvey-Rating. After participants examined the product, another survey was conducted to determine their behaviors, thoughts, and feelings. The main function of this Postsurvey was to collect overall evaluation of a given product from the participants' perspective. Participants rated the given product by using a slider range from 0 to 10, where 0 refers to minimum tendency to buy and 10 the maximum tendency to buy.

Postsurvey-Feeling. After overall evaluation, the survey focused on participants' feelings that they just experienced. Self-ratings of excitement, motivation, enjoyment, distraction, tiredness, and confusion were gathered to infer the overall emotional arousal, motivation, engagement, distraction, and workload as seen in Table 10.

Postsurvey-Evaluation of User Interface Sections. The third part of the rating survey aimed to identify the user experience and thoughts about the efficiency of web elements (visuals, rating and comments, and descriptions and prices in the given web stimuli). Participants recalled the user interface of the product web page and input the efficiency of these specific areas by using a 7-choice Likert scale changing from *extremely useless* to *extremely useful*.

Table 10: The Variables of Postsurvey in the Traditional UX Analysis

#	Self-report	Variable	Description
1	Feeling	Confusion	Rate of confusion that user felt
2	Feeling	Distraction	Rate of distraction that user felt
3	Feeling	Enjoyment	Rate of enjoyment that user felt
4	Feeling	Interest	Rate of interest that user felt
5	Feeling	Tiredness	Rate of tiredness that user felt
6	Feeling	Frustration	Rate of frustration that user felt
7	UIS	Product Visuals	Importance of product visuals on subjects' decisions
8	UIS	Ratings and Comment	Importance of rating and comments on subjects' decisions
9	UIS	Descriptions	Importance of descriptions on subjects' decisions
10	UIS	Cost	Importance of cost on subjects' decisions

User decision survey. After participants examined and rated both products in one of the support levels, a user-decision survey served to identify their last decision and confidence on the final selection in the given support level. The survey has three versions specialized for each support level category. In each version, users checked the summary of all two choices and then explained their preference to determine the final desired product choice for online purchase.

User experience ranking survey. After six postsurvey and three decision surveys, finally, a ranking survey was used to determine the participants' final comparison behaviors. Participants adjusted the ranking list according to their preference on how likely they were to buy or to eliminate the given products. While #1, or the first choice at the top of the list, referred to the choice they were most likely to buy, #6, or the last choice at the bottom of the list, meant the choice that they were less likely to buy. After adjusting this list, participants also reported their reasoning on this ranking in the open-ended section of the survey.

Traditional self reports conducted in the study to gather information about both participants' behavioral response and their feelings were used to predict the user ratings and ranking scores. The findings were compared and collaborated with the sensor data for overall UX evaluations.

Sensor-based UX Measures

Psychophysiological data was captured through BMS. This method of measuring participants' biometric responses provided non-intrusive real-time information to interpret the participants' cognition, emotion, and behaviors, which are relevant to comprehensive recognition of the winner and the loser choices. Table 11 to Table 14 summarize all inferences and their related variables and measurement sources. The following paragraphs provide more information about these variables.

In the analysis of biometrics section, pupil dilation and fixation variables are used. The description of 5 variables, seen in Table 11, are used to interpret the visual attention through eye tracking variables. Likewise, participants affective and cognitive dynamics are interpreted by using EEG variables listed in Table 12.

Table 11: The Eye Tracking Variables of Biometrics

#	Variable	Description
1	PupilLeft	Pupil size left eye in millimeters
2	PupilRight	Pupil size right eye in millimeters
3	FixationStart	The fixation start time in milliseconds
4	FixationDuration	The fixation duration in milliseconds
5	FixationAOI	The AOI that the fixation belongs to

Table 12: EEG Variables of Biometrics

#	Variable	Description
1	High Engagement	The engagement index is showing attentiveness and focus related to information-gathering, sustained attention, and visual scanning processes
2	Workload FBDS	The workload index during forward backward digit span tasks (i.e., working memory, planning, recall)
3	Workload BDS	The workload index during backward digit span tasks (i.e., recall)
4	Workload Average	The average between FBDS and BDS
5	Classification	The classification index includes all cognitive and affective states during information-gathering and visual scanning.
6	Distraction	The distraction index that indicates that classification index value is around 0.3.
7	Drowsy	The drowsiness index which is projected version of the continuous classification index once it is between 0.1 and 0.3

Besides eye tracking and EEG variables, five GSR variables were included in the analysis. These were peak count, number of peaks per minute, average amplitude, max amplitude level and calibrated GSR variables. Finally, the last biometric tool was FEA. Seven essential emotions (anger, sadness, disgust, joy, surprise, fear and contempt), burrow furrow, brow raise, lip corner depressor, smile, valence, and attention are included the biometrics analysis as listed in the Table 13.

Table 13: GSR Variables of Biometrics

#	Variable	Description
1	PeakCount	The number of peaks that the respondent had during this time period
2	PeakMin	The number of peaks the respondent had total during this stimulus or scene divided by the duration of the stimulus converted to peaks/min.
3	AveAmplitude	The average amplitude of the peak
4	MaxAmplitude	The max amplitude of the peak
5	Cal GSR	The skin conductance (unit: micro-Siemens).

Table 14: FEA Variables of Biometrics

#	Variable	Description
1	Brow.Furrow	An expression where both eyebrows moved lower and closer together
2	Brow.Raise	An expression where both eyebrows moved upward
3	Lip Corner Depressor	An expression where lip corners dropping downwards
4	Smile	An expression where lip corners pulling outwards and upwards towards ears
5	Valence	A measure of how positive or negative the expression is.
6	Attention	Measure of focus based on the head orientation
7	Anger	The index measuring the subjects' aggregated anger level
8	Sadness	The index measuring the subjects' aggregated sadness level
9	Disgust	The index measuring the subjects' aggregated disgust level
10	Joy	The index measuring the subjects' aggregated joy level
11	Surprise	The index measuring the subjects' aggregated surprise level
12	Fear	The index measuring the subjects' aggregated fear level
13	Contempt	The index measuring the subjects' contempt level

The exported sensor data of BMS produced 134 different variables from the four sources of biometrics tools. However, 30 variables—5 variables from eye tracking, 7 from EEG, 13 from FEA, and 5 from GSR—are included to simplify the analysis of the big data (12.5 GB).

The variables and constants. In the study, the data representing the user rating and ranking scores for each product was the dependent variable, while the rest of the sensor and traditional variables were independent variables. The specific products were shown through one retailer website, Amazon.com, to reduce confounding factors, such as the knowledge organization structure, interaction flow of the purchase process, visual and graphic style, overall user interface design of the website, and its navigational features and interactions. Additionally, the type of the device, operating system, and the software application that was used for online purchasing were kept constant for all data collection sessions.

Biometric Sensor Suite

The study was conducted by using iMotion 6.4, the Biometric Measurement System (BMS) software, a biometric sensor suite integrating several biosensors and synchronizing and visualizing eye tracking, facial expression analysis, electroencephalogram (EEG), Galvanic Skin Response (GSR), and surveys into one platform.

Table 15: BMS Equipment

BMS Equipment	Make	Model
Remote Eye Tracking – Lab Environment	Tobii	Pro X2-60
Gel-based EEG Headset	ABM	B-Alert X10
GSR	Shimmer3	GSR Device
Webcam	Logitech	C920
Monitor, mouse and keyboard for Participants		
Laptop for Researcher		
Biometric Platform Combining Sensors	iMotions	Core 6.4 and modules

This platform enables researchers to collect psycho-physiological data to measure participant biometric responses while engaged in specific tasks. Table 15 below provides the equipment list for the proposed BMS.

The data collection environment was composed of two sections as seen in Figure 2. The left side was designed for participants and the right side for the researcher. All participants used the same biometric sensor suite consisting of 4 sensor tools—a remote eye tracker attached to the bottom side of the 22-inch monitor, gel-based EEG headset, GSR device on their non-dominant hand, and a HD Webcam attached to the upper side of the monitor as seen in Figure 3. The researcher conducted the study and monitored participants' emotional and cognitive dynamics and behaviors through the display of the BMS as seen in Figure 4.

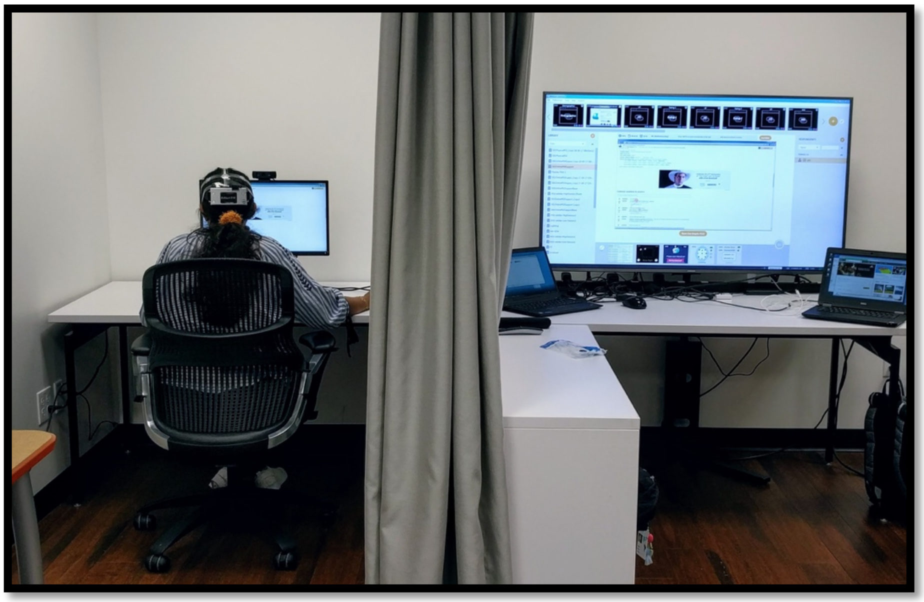


Figure 2: Data Collection Environment

Figure 3: A Participant Using Biometric Suite During Data Collection

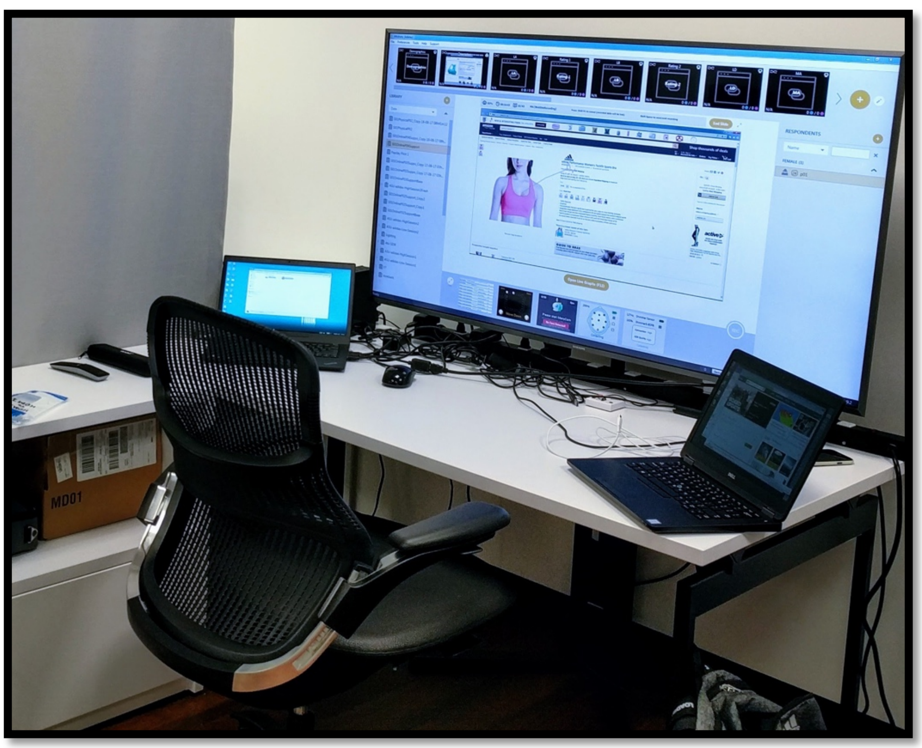


Figure 4: Researcher Side View of the Data Collection

Statistical Approach

This subsection aims to explains the statistical approach that was used in the analysis of the study. It consists of two main parts. The first part focuses on the information of decision trees about definition, description of tree formation, types of trees, benefits and risks of decision trees. The second part focuses on the practice of decision trees through a statistical software and its important features to run the decision trees.

Decision tree approaches, also known as classification tree and regression tree, have frequently been used in data mining to create prediction models for a dependent variable (Breiman, Friedman, Stone, & Olshen, 1984). Populations are divided into dichotomous branches, which form an upside-down-shaped tree structure starting from root nodes, through internal nodes to leaf nodes. In this way, recursive partition of the dataset into a two-way prediction model within each partition forms the decision tree (Loh, 2011). If the dependent variable consists of discrete values, the tree is called a classification tree. Prediction error is calculated by the rate of misclassification. However, if the dependent variable is continuous, the tree is called a regression tree and the squared difference between the observed and predicted values forms the prediction error (Breiman, Friedman, Stone, & Olshen, 1984).

Decision trees are non-parametric approaches without distributional assumptions, and they make the complex relationships between input variables and target variables easy to understand. In addition, they facilitate the interpretation of the data, since they illustrate the relation of predicted responses and their variables. However, decision trees are vulnerable to cause overfitting and underfitting, particularly when the sample size of the study is low (Veloso, Meira, & Zaki, 2006; Loh, 2014; Song & Ying, 2015).

In the study, both regression and classification trees were applied by using R, a free statistical software for statistical computing and data analysis. The R was run through RStudio, an integrated development environment (IDE) for R and consists of a console, syntax-highlighting editor for code execution, and some widgets for plotting, history, debugging and workspace management.

R has several prepackages for a wide variety of statistical computing needs. One of the R packages to practice the decision trees is Recursive Partitioning and Regression Trees, called rpart. The rpart algorithm runs by dividing the dataset recursively until the leaf node is formed in the tree structure (Therneau & Atkinson, 2018). If the desired analysis is a regression tree, the method setting is adjusted to "anova," otherwise it is set to "class." The implementations of these trees were explained in data analysis (4.2) of chapter 4.

The major task of the data analysis of the study was to develop a model predicting the user's rating scores and rating categories by using sensor variables, traditional variables and hybrid variables, a combination of sensors, and traditional surveys. In the study, first six regression tree analyses are run by using sensor-base data to analyze the rating scores, which are integers ranging from 0 to 10. Then, these rating scores were classified into three groups and 18 classification tree analysis. The rpart package was performed to develop a model predicting the user's rating classes by using sensor variables, traditional variables and hybrid variables. The results of analysis are presented and discussed in Chapters 4 and 5.

CHAPTER 4

ANALYSIS AND RESULT

This analysis of the study includes three sections: (1) Data Export and Cleaning; (2) Data Analysis; and (3) Tools and Variables. The following paragraphs provide detailed information related to each section.

Data Export and Cleaning

Predicting user ratings through collected sensor data and surveys requires some post-process after data collection. It was a relatively simple and quick process to export survey data through the Qualtrics online survey system. After logging in to the researcher's account, all survey results were downloaded (with some minor modifications: removing the users from the pilot sessions of the study and adjusting the recordings to present the variables in numeric format) the total data were transferred to the analysis file for further processing.

The processing of the sensor data was more complex and required additional attention to obtain precise and valid data from BMS. The first process was exporting the sensor data. Within the library section of the BMS software, the analysis section is formed first. Then sensor data were exported by choosing the required stimuli and users. In this case the web page stimuli included 6 different products and users. As the next step, the desired tool sources (Tobii X2-30 eye tracking, ABM B-Alert X10 EEG, Affectiva AFFDEX for facial expression data, and Shimmer 3 Sensor for galvanic skin response data)were selected to export the data. The BMS produced .txt files, consisting of

138 variables for each user as an output of sensor data export post processes for the next step in data cleaning.

The total size of the sensor data was around 13 GB; this big data was reduced and cleaned to form just the required variables for further statistical analysis. The data cleaning process had 4 major steps. First, the variables that were excluded from data analysis in the dataset were removed. Then specific variables, which are "ValidityLeft," "ValidityRight," "Number of faces," and "GSR Quality" were filtered to remove all low quality or invalid data. In this process, "ValidityLeft" and "ValidityRight" were set to "0," which indicates valid gaze point only. "Number of faces" was set to "1" to remove the unrecognized facial expression cells. Likewise, "GSR Quality" was set to "Valid" to remove the invalid data.

After these removals and filtrations, the scores of each required variable in the exported data set were aggregated to transform the value of variables in one cell per user, which allowed researchers to combine both traditional data and sensor-based data in a unique data set. Each row of this data set represented one participant. The aggregation was mostly done by measuring the mean of the variables, except for two, "FixationStart" and "FixationDuration." The min fixation score indicated the first fixation time as "FixationStart." The sum of fixation score to indicate the total fixation time is "FixationDuration." As a final step of sensor data cleaning, all aggregated data were organized by using specific data organization codes through a Python programming script. In the final step, the cleaned traditional survey data and sensor data were combined in one unique data set.

Data Analysis

The major goals of the data analysis of the study were to develop a model predicting the user's rating scores, rating class sets; to reveal the model specified variables, and to predict the user decisions within the same type of support level. For this purpose, four different analysis was reported in this section by using traditional, biometric, and hybrid variables. The first one was regression tree analysis to predict the user's rating scores through one product. Similarly, the second one was classification tree analysis to estimate the user's class sets. The error rate in cross-validation of rating classes and model accuracies was discussed to check the usefulness of the fitted model for the prediction of the user rating class sets. In the third analysis, the focus was the variables of the accurate classification trees to reveal the model specified variables for traditional, biometric, and hybrid UX approaches. Finally, the fourth analysis focused on predicting the user decisions within the same type of support level. The following subsections explain the detailed steps of each analysis and discuss the results.

Regression Tree Analysis

Rating scores were integers scaled from 0 to 10, thus a regression tree approach was applied on rpart. In the first part of the analysis, a data set, which included sensor-based UX measures related to product #1, was used and its histogram were formed and showed that the data took on the exponential shape shown in in Figure 5.

```
hist.default(RatingP1A1Mdata3$Rating_1, main = "Histogram of Rating Scores For P1",
             xlab = "Rating", ylab = "Frequency")
```

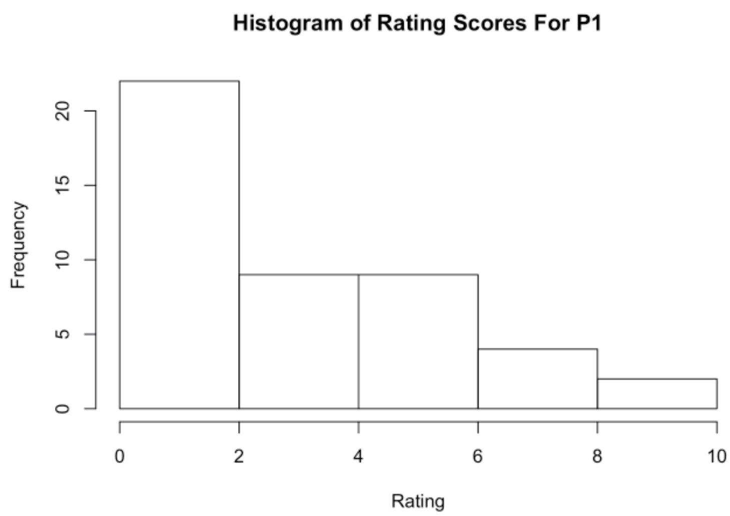


Figure 5: Product 1 Histogram of Rating Scores From Sensor Data

Next, the data set was divided into training and test sets. While the former was used to build the model, the latter was used to test it. The rpart algorithm set the train and test data frames by randomly selecting 80% of the data for the training set and assigning the remainder to the test data set as shown in Figure 6.

```
train <- sample (1:nrow(RatingP1A1Mdata3), size=0.8*nrow(RatingP1A1Mdata3)) # training row indices
RatingP1A1_train <- RatingP1A1Mdata3[train, ] # training data
RatingP1A1_test <- RatingP1A1Mdata3[-train, ] # test data
```

Figure 6: R Codes to Create a Training and Test Data Sets

Then, rpart was invoked as seen in Figure 7: Product 1 Regression Tree. It was seen in the syntax section that the *method* parameter set to "*anova*" stating *rpart* that the predicted variable is continuous.

```
#Classification Tree
formula=Rating_1 ~ PupilLeft_1 + PupilRight_1 + minFixationStart_1 + sumFixationDuration_1 + Classification_1 + H
ighEngagement_1 + LowEngagement_1 + Distraction_1.1 + Drowsy_1 + WorkloadFBDS_1 + WorkloadBDS_1 + WorkloadAverage
_1 + BrowFurrow_1 + BrowRaise_1 + LipCornerDepressor_1 + Smile_1 + Valence_1 + Attention_1 + Anger_1 + Sadness_1
 + Disgust_1 + Joy_1 + Surprise_1 + Fear_1 + Contempt_1 + PeakCount_1 + Peak.Min_1 + AveAmplitude_1 + MaxAmplitud
e_1 + GSR_CAL_1

RatingP1A1_regTree=rpart(formula,data=RatingP1A1_train,method="anova",control=rpart.control(minsplit=5,cp=0.001))
#plot(RegTree)
plot(RatingP1A1_regTree, uniform=TRUE,
     main="Regression Tree For P1 Rating")
text(RatingP1A1_regTree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number of observations at each node
```

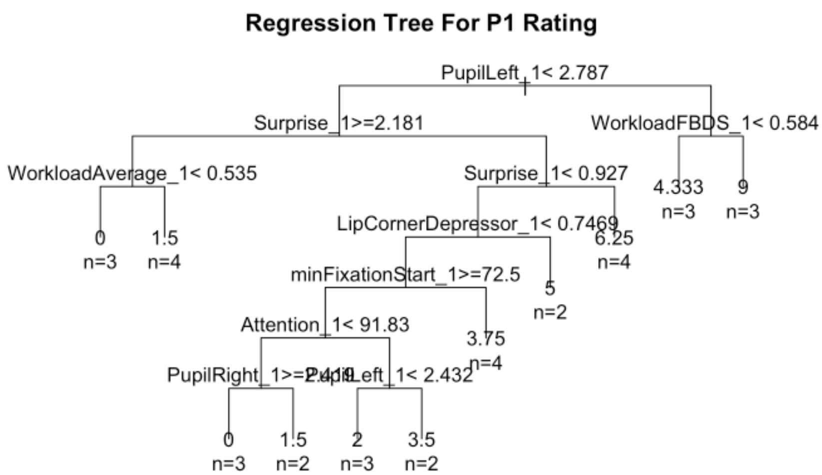


Figure 7: Product 1 Regression Tree

Afterward, the reliability of predictions was checked. Since the dependent variable is continuous, comparing the predictions directly against the test set was not the right way. In its place, the root mean squared error (rmse) and mean absolute error (mae) was used. By adding small scripts to R rpart syntax, the algorithm yielded the predictions as a vector, which is one prediction per record in the test dataset. Then rmse and mae scores were calculated by comparing this vector with the *rating* column in the test dataset as shown in Figure 8. Similarly, rmse and mae values were calculated as seen in Figure 9 in order to compare the two data frames.

```r
# Function that returns Root Mean Squared Error
rmse <- function(error_train)
{
  sqrt(mean(error_train^2))
}
# Function that returns Mean Absolute Error
mae <- function(error_train)
{
  mean(abs(error_train))
}
#mae(actual, predicted)
# Print rmse and mae in train data
rmse(error_train)
```

```
## [1] 0.763027
```

```r
mae(error_train)
```

```
## [1] 0.5334758
```

Figure 8: Functions Returning the Rmse and the Mae in Trained Data.

```r
# Function that returns Root Mean Squared Error
rmse <- function(error_test)
{
  sqrt(mean(error_test^2))
}
# Function that returns Mean Absolute Error
mae <- function(error_test)
{
  mean(abs(error_test))
}
#mae(actual, predicted)
# Print rmse and mae in test data
rmse(error_test)
```

```
## [1] 3.568963
```

```r
mae(error_test)
```

```
## [1] 2.883333
```

Figure 9: Functions Returning the Rmse and the Mae in Test Data

According to the scores in train data frame and in test data frame, although the model created in regression tree showed high performance in the training dataset resulting in low *rms*e and *mae (0.76 and 0.53 respectively)*, the model was unable to predict users' rating scores, since the *rmse* and *mae* scores are high *(3.57 and 0.2.89* *respectively)*, which means there is a huge difference between the actual scores and predicted scores.

Next, this analysis process was repeated for the rest of the five products by using surveys and then by using biometric data to compare the efficiency of these two major UX approaches. The *rmse* and *mae* scores are calculated in each condition. Overall, the rmse and mae scores are high in all six products of both survey and biometrics conditions, which means that models have some difficulty in predicting the rating scores in the 1-10 scale in Table 16 and Table 17.

Table 16: The Rmse Scores of the Two Major UX Data Source

Support Level	Pr ID	Root Mean Squared Error (Surveys)	Root Mean Squared Error (Biometrics)
Low	1	4.23	3.57
	2	3.78	4.13
Medium	3	3.26	3.77
	4	3.60	4.45
High	5	4.13	3.36
	6	3.17	4.36

Table 17: The Mae Scores of the Two Major UX Data Source

Support Level	Pr ID	Mean Absolute Error (Surveys)	Mean Absolute Error (Biometrics)
Low	1	3.07	2.88
	2	2.95	3.13
Medium	3	2.75	3.13
	4	2.76	3.78
High	5	3.83	2.85
	6	2.68	3.59

Classification Tree Analysis

Before examining the computational formation of the classification trees, it is worth discussing the logical background of this categorization in the study. The study aimed to determine the groups who tend to buy the product. In this case it is assumed that the subjects who rated the product as 9 or 10 could be a class who were most likely to buy the product. In the opposite direction, it is also assumed that the subjects who rated the product as 6 or lower could be a class who were less likely to buy the product. The group between these two opposite ends considered as another class. Therefore, the rating scores classified based on the 3 classes:

- **Class 1 (Scores 0-6):** Subjects unhappy with the product and would be most likely to not buy the product.
- **Class 2 (Scores 7-8):** Subjects satisfied with the product but product may not be their favorite.
- **Class 3 (Scores 9-10):** Subjects happy with the product and would be most likely to buy the product.

At a first glance, the categorized rating scores seems to be the same with the Net Promoter Score (NPS) classes developed by Reichheld in 2003, but in fact, they are not. Reichheld categorized the first group was "Promoters," consisting of the scores 9 and 10, which represent the users most likely to recommend the product their colleagues or friends. Then, he named "Passives" made up the second group, participants who rated the product 7 and 8. This means that users were satisfied with the product but were also vulnerable to competitive offerings. Finally, the last group was named as "Detractors," participants who rated the product from 0 to 6. This class indicated that users showed red alerts, they were unhappy with the product and would most likely not recommend the product to their colleagues or friends (Reichheld & Markey, 2011). Although the classification approach used in the study have some similar class sets, in fact, they are not the same categorization. While NPS asks how likely to recommend the product their colleagues or friends, the classifying system used in the study questioned how likely to buy the product. Thus, their psychometric affects are different.

After all the information above, it is time to explain the computational formation of the classification trees. The similar steps used in previous regression tree analysis was repeated in R syntax in order to perform the classification tree. The working directory was set, the required library packages, namely, rpart, ggplot2 and Hmisc, were loaded and then the new data set, including the three classes mentioned above, were loaded to the R. Next, the data set was divided into train and test data frames in the proportion of 80% and 20%, which was arbitrarily selected. The train data was used to build the model and the test data to test the fitted data. Afterward, rpart was invoked. Unlike from regression tree approach used in the previous analysis, in this classification tree approach, method parameter was set to "class" in the syntax, which means that the predicted variable is categorical as seen in Figure 10.

```
#Classification Tree
formula=NPSID_1 ~ PupilLeft_1 + PupilRight_1 + minFixationStart_1 + sumFixationDuration_1 + Classification_1 + Hi
ghEngagement_1 + LowEngagement_1 + Distraction_1.1 + Drowsy_1 + WorkloadFBDS_1 + WorkloadBDS_1 + WorkloadAverage_
1 + BrowFurrow_1 + BrowRaise_1 + LipCornerDepressor_1 + Smile_1 + Valence_1 + Attention_1 + Anger_1 + Sadness_1 +
 Disgust_1 + Joy_1 + Surprise_1 + Fear_1 + Contempt_1 + PeakCount_1 + Peak.Min_1 + AveAmplitude_1 + MaxAmplitude_
1 + GSR_CAL_1

RatingP1A1NPSdtree=rpart(formula,data=RatingP1A1NPS_train,method="class",control=rpart.control(minsplit=5,cp=0.00
1)) # build the model

#plot(dtree)
plot(RatingP1A1NPSdtree, uniform=TRUE,
     main="Classification Tree For P1 Rating")
text(RatingP1A1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number of observations at each node
```

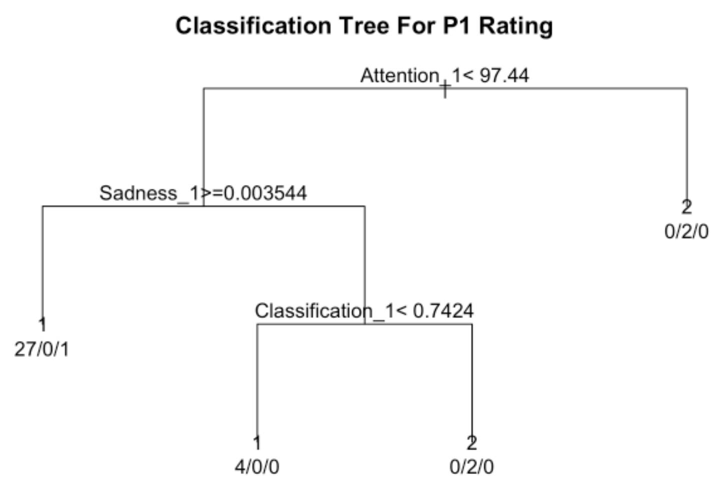


Figure 10: Product 1 Classification Tree

Next, the validity of the tree was checked through “printcp” output as seen in Figure 11. In this CP table, *CP* refers to complexity parameter, and help researchers to optimize size of tree. While small *CP* values result in larger trees and potential overfitting, large *CP* values result in small trees and potential underfitting. *Nsplit* means a number of split in the model. *Rel error* is a re-substituting error measure. Multiplication of *root node error* and *rel error* yield to the prediction error rates in training data. Finally, *xerror* refers to the cross-validation error estimate and multiplication of *root node error*, and *xerror* equals to the prediction error rates in cross-validation, which is absolute cross-validated error. The accuracy is gained by subtracting this value from 1.0.

```
printcp(RatingP1A1NPSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RatingP1A1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Attention_1      Classification_1 Sadness_1
##
## Root node error: 5/36 = 0.13889
##
## n= 36
##
##      CP nsplit rel error xerror    xstd
## 1 0.400      0       1.0    1.0 0.41500
## 2 0.200      1       0.6    1.4 0.47493
## 3 0.001      3       0.2    1.6 0.49889
```

Figure 11: The CP Table for Product 1Classes

The measurements of prediction error rate in cross-validation have been done for all products by using (1) surveys data only, (2) biometrics data only, and (3) combined version of surveys and biometrics data, called hybrid data. The results are shown in the following tables.

Table 18 showed that prediction error rate in cross-validation varied between 26% and 66% as a misclassification error, which means that there is 34-74% accuracy in prediction of user's rating class. Instead of interpreting this wide range between 6 different products, they are looked at closer based on rating classes. In the low support level, while the prediction error rate in cross-validation for Product #1 results in 33%, Product #2 was predicted a little bit less with an error rate of (26%). Overall, it can be stated that user rating classes of the low support level products were predicted with approximately 30% error rate in cross-validation. That means the classification tree model applied on low level support groups predicted the rating classes with 70% accuracy. In other words, the model predicted every 7 rating classes correctly out of 10. Similarly, in the high support level, while the prediction error rate in cross-validation for

Product #5 results in 45%, Product #6 was predicted with a smaller error rate (27%). However, the prediction results in the medium level were not as good as in high and low support levels. The highest prediction error rate in cross-validation score, which is 66%, was obtained for Product #3. Relatively, Product #4 had a 32% misclassification rate. Therefore, the prediction of the rating classes analysis through surveys provided worthwhile data for the prediction of user preferences in low support level, but medium and high support level scores were not as good as in low support, because of wide range between the products of medium and high support levels.

Table 18: Prediction of Rating Classes (Surveys Only)

Prediction of Rating Classes: Promoters (9-10), Passives (7-8) and Detractors (0-6) (Surveys Only)					
Pr ID	**Root node error**	**rel error**	**xerror**	**Prediction error rate in training data**	**Prediction error rate in cross-validation**
1	0.14	0.60	2.40	8%	33%
2	0.21	0.13	1.25	3%	26%
3	0.47	0.22	1.39	11%	66%
4	0.24	0.67	1.33	16%	32%
5	0.37	0.14	1.21	5%	45%
6	0.16	0.17	1.67	3%	27%

After survey-only data, analysis continued with the biometrics-only data frame. Table 19 showed that prediction error rate in cross-validation varied between 22% and 62% as a misclassification error in prediction of user’s rating class. At first glance, this range seems to be so close to the range of the surveys’ data, in fact, the range of the low support and high support levels are equal or small. In the low support level, the prediction error rate in cross-validation for both Product #1 and #2 results in 22%. Thus, it can be stated that user rating classes of the low support level products predicted with approximately 20% error rate in cross-validation, which was 30% error rate in the surveys. In other words, the model predicted every 8 rating classes correctly out of 10. Similarly, in the high support level, while the prediction error rate in cross-validation for Product #5 results in 24%, Product #6 was predicted with a slightly higher error rate (25%). However, the prediction results in medium level were much worse than low and high support levels. The highest prediction error rate in cross-validation score, which is 62%, was obtained for Product #3. It was 43% in Product #4. Therefore, the prediction of the rating classes analysis through biometrics provided useful results for the high and low support levels. However, the prediction results for the medium support level were not as good as in low and high support, because of high error rate in cross validation and also the higher range between the two products of the medium support level.

Table 19: Prediction of Rating Classes (Biometrics Only)

Prediction of Rating Classes: Promoters (9-10), Passives (7-8) and Detractors(0-6) (Biometrics Only)					
Pr ID	**Root node error**	**rel error**	**xerror**	**Prediction error rate in training data**	**Prediction error rate in cross-validation**
1	0.14	0.20	1.60	3%	22%
2	0.16	0.33	1.33	5%	22%
3	0.46	0.35	1.35	16%	62%
4	0.24	0.24	1.78	6%	43%
5	0.27	0.30	0.90	8%	24%
6	0.16	0.20	1.60	3%	25%

The analysis showed that biometrics-only data presented higher accuracy rates in both low and high supports level than in surveys-only data. Strikingly, prediction error rate in cross-validation decreased to 19% in Product #1 and 13% in Product #2 as shown in Table 20. Likewise, it reduced to 22% in Product #6, but it was kept in Product #5. Also, prediction error rate in cross-validation decreased to 53% in Product #3 and 35% in Product #6, once compared with the results of biometrics only condition.

Table 20: Prediction of Rating Classes (Surveys and Biometrics)

Prediction of Rating Classes: Promoters (9-10), Passives (7-8) and Detractors (0-6) (Biometrics+Surveys)					
Pr ID	**Root node error**	**rel error**	**xerror**	**Prediction error rate in training data**	**Prediction error rate in cross-validation**
1	0.14	0.60	1.40	8%	19%
2	0.16	0.17	0.83	3%	13%
3	0.53	0.20	1.00	11%	53%
4	0.22	0.25	1.63	5%	35%
5	0.24	0.00	1.00	0%	24%
6	0.16	0.33	1.33	5%	22%

As an overall comparison of three support groups, the mean of accuracy rates for each support level are calculated by using the two products that they included as shown in Figure 12. The study showed that the accuracy rate in cross-validation of rating classes varied between 51% and 70% once only surveys are used. Although 70% accuracy in low support can be interpreted as a meaningful prediction, the others were not helpful to predict the users' preferences. Thus, the performance of survey-only condition showed weaknesses in the prediction of user choices.

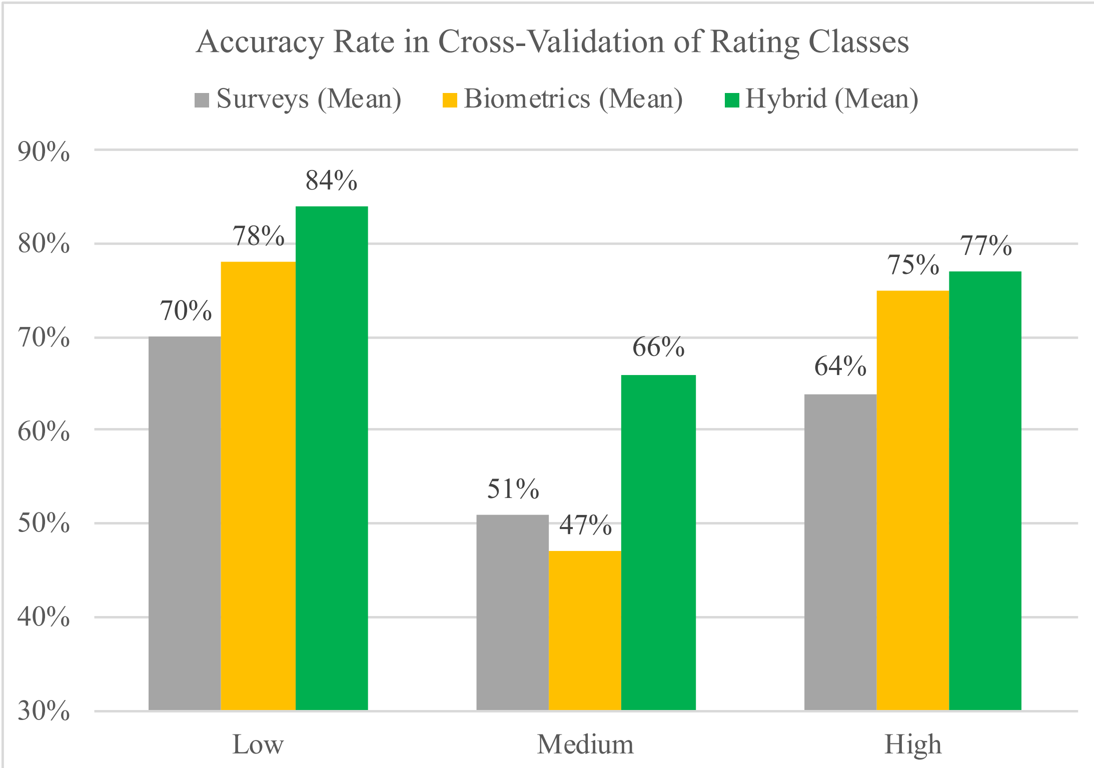


Figure 12: Accuracy Rate in Cross-Validation of Rating Classes

As shown in Figure 12, the use of biometrics showed higher accuracies, particularly in low and high support levels. Although the performance of biometrics in medium support was slightly lower than surveys, the remarkable accuracy increases in low and high support levels hit to the 75% or above and showed that they could be used to reduce the weakness of surveys-only conditions to predict the users' preferences.

Besides, when both systems are used together, the accuracy rate in cross-validation of rating classes reached the highest rate in all support levels, which is another essential piece of evidence that related to the second research question investigating which model (sensor-based, traditional, or combined) can best explain customer purchasing preferences. In the hybrid condition, the top score (84%) was observed in low support products in hybrid condition. However, the sharp increase in the accuracy rate in medium support level was worth to take attention more. Despite the fact that there were profoundly low accuracy scores both in the survey only (51%) and biometrics only (47%) conditions of the medium support level, the hybrid conditions escalated the accuracy rate in cross-validation of rating classes in medium support level to 66%. This means that although the classification tree model can predict accurately almost every 7 of 10 users' preferences in medium support level, based on this cross-validation of rating classes. This also can be interpreted as evidence that in the specific cases where traditional survey-based UX methodologies unable to predict user's preferences, integration of sensor-based UX approaches to the current traditional survey-based UX methodologies and the collaborative use of both may cause useful and efficient prediction of user's preferences.

Model Specified Variables

In the previous section of the analysis, the accuracy rates are analyzed to make sure that they are reliable and meaningful for UX researcher to clarify and optimize the most beneficial tool in further researches. This section of Chapter 4 focuses on revealing the variables which are used in the rating class prediction analyzes. In this way, the investigation clarifies which sensors, whether separately or integrated, can best be used to explain customer preferences for purchasing. In that purpose, first, variables, functioned in the decision tree constructions are determined, and then sensor tools including these variables are discussed based on their efficiency in the study to reveal the optimum source combination for the prediction of user rating classes.

```r
#Classification Tree
formula=NPSID_6 ~ PupilLeft_6 + PupilRight_6 + minFixationStart_6 + sumFixation
Duration_6 + Classification_6 + HighEngagement_6 + LowEngagement_6 + Distractio
n_6.1 + Drowsy_6 + WorkloadFBDS_6 + WorkloadBDS_6 + WorkloadAverage_6 + BrowFur
row_6 + BrowRaise_6 + LipCornerDepressor_6 + Smile_6 + Valence_6 + Attention_6
 + Anger_6 + Sadness_6 + Disgust_6 + Joy_6 + Surprise_6 + Fear_6 + Contempt_6 +
 PeakCount_6 + Peak.Min_6 + AveAmplitude_6 + MaxAmplitude_6 + GSR_CAL_6

RatingP6A1NPSdtree=rpart(formula,data=RatingP6A1NPS_train,method="class",contro
l=rpart.control(minsplit=5,cp=0.001)) # build the model

#plot(dtree)
plot(RatingP6A1NPSdtree, uniform=TRUE,
     main="Classification Tree For P6 Rating")
text(RatingP6A1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

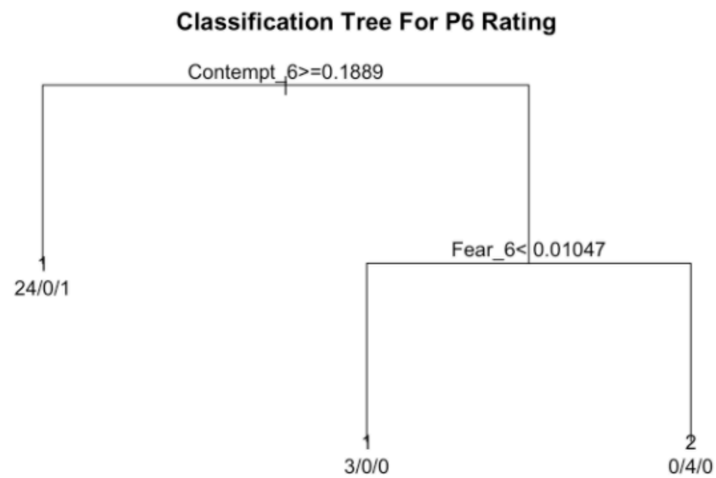


Figure 13: The Classification Tree for Product #6

```r
printcp(RatingP6A1NPSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RatingP6A1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Contempt_6 Fear_6
##
## Root node error: 5/32 = 0.15625
##
## n= 32
##
##       CP nsplit rel error xerror    xstd
## 1 0.400      0       1.0    1.0 0.41079
## 2 0.001      2       0.2    1.6 0.48990
```

Figure 14: The Output of the Printcp Script

Although several tools and variables were used as input in the decision tree models, a few combinations of tools were desirable for most of the prediction cases. For example, as shown in Figure 14, the classification tree includes just two different variables although the formula includes 30 variables. These two variables are printed as a function of “printcp” script and indicated as “Variables actually used in tree construction” in Figure 15. In the following subsections, first the variables and then tools were analyzed three conditions sequentially.

Tool Selection in Survey Only Condition

When the survey-only analysis is considered, it was noted that while “interest” was common in all products, the attitude survey variables dominated the used variable list in the classification tree survey only analysis. However, there were not so common variables within the all support levels. In fact, they are so diverse from each other. It is also revealed that user interface sections were the least used variable in this analysis, where the two elements, “product descriptions” and “photos”, are the only used variables. Also, “Cost in Overall” and “Everyday Bra Size” were the only variables used in the model from the demographic survey in surveys-only condition.

Besides model specified variables in surveys-only condition, the analysis of the used tool was informative to detect the important sources. The model specified variables categorized based on the sources in Table 21 showed that at least two different sources combined to create a classification tree model in surveys only conditions. These are the ones used to predict the rating classes of Products #1, #2, #4, and #6. The trees combining three different surveys were for Products #3 and #5. While the former was composed of attitude, Feelings, and UIS, the later consisted of demographics, attitude, and feeling surveys. It is noted that all four sources together were including in neither of the product rating class predictions. However, the attitude survey and feeling surveys are dominated the all six product rating class predictions in terms of used tools.

Table 21: Model Specified Variables and Their Tool of Survey Only

Prediction of Rating Classes (Surveys Only)						
Support	**Pr ID**	**Prediction error rate in cross-validation**	**Demographics**	**Attitude**	**Feeling**	**UIS**
Low	**1**	33%		Time Spent on Daily Sports Activities	Interest,	
	2	26%		Product Descriptions in Overall, Time Spent on Internet/Day	Interest	
Medium	**3**	66%		Money Spent on Sports Bra, Money Willing to Spent on Sports Bra, Time Spent on Daily Sports Activities	Frustration, Interest	Descriptions, Photos and Videos,
	4	32%		Time Spent on Internet/Day	Interest	
High	**5**	45%		Coach Driven, Sports Rate, Cost in Overall, Everyday Bra Size	Enjoyment, Interest	
	6	27%		Cost in Overall, Frequency of Sports Bra Use	Interest	

Thus, based on the survey-only conditions, it is recommended that attitude and feeling surveys may be worth taking into consideration for the optimum tool selection to predict the users' rating classes in the design of future studies.

Tool Selection in Biometrics Only Condition

Similar analyses were done for the biometrics only conditions. While eye tracking variables were used in half of the six products, EEG variables used four of the six products. The FEA variables were the most model specified variables since they are used in the prediction of five of the six products. In fact, the classification tree of Product #6 Rating classes consists of only FEA variables, “contempt” and “fear” namely. Unlike Product #6, the rating classes of the other four products predicted by combining two pairs from the sensors, such as FEA and EEG for Product #1 and #5, FEA and ET for Product #3, and EEG and Eye tracking for Product #2. Product 4, was the only one where triple combination observed, including FEA, EEG and eye tracking. In overall, FEA variables were not only used in most products but also, they are the top in the number of model specified variables for each prediction classes of the six products.

It is also noted that the tools used in the biometrics only analysis were ET, EEG, and FEA, but not GSR as seen in Table 22. Surprisingly, while FEA, composed of “contempt” and “fear” variables, was the only tool used in Product #6, the other products rating classes are predicted by using just two tools: EEG and FEA in Product #1, ET and EEG in Product #2, and FEA and EEG in Product #5. Except for Product #2, the FEA dominancy was observed in tools used in tree constructions once the biometrics only condition was applied. The FEA followed by EEG and then ET in terms of mostly used tools in biometrics only condition.

Table 22: Model Specified Variables and Their Tool of Biometric Only

Prediction of Rating Classes (Biometrics Only)						
Support	**Pr ID**	**Prediction error rate in cross-validation**	**Eye T**	**EEG**	**FEA**	**GSR**
Low	**1**	22%		Classification	Attention, Sadness	
	2	22%	Sum of Fixation Duration	Drowsy, WorkloadBDS		
Medium	**3**	62%	Min Fixation Start, PupilRight		Attention, Sadness, Smile	
	4	43%	PeakCount, PupilLeft,	Classification, Drowsy,	BrowFurrow, Valence	
High	**5**	24%		Low Engagement	Lip Corner Depressor, Valence	
	6	25%			Contempt, Fear	

Last but not least, the analysis of biometric only conditions revealed that attention-focused variables was the mostly used to determine the users rating classes. While attention variable from FEA, pupil dilation, fixation variables from eye tracker, which are the efficient indicators of the visual attention, drowsy and low engagement from EEG that might be an interpretation of low attention was appeared in the used variable list once the biometric only condition was used. Also, it is good to the recall that these biometrics provided better accuracies when compared with the survey-based variables.

Therefore, it is recommended for UX researchers who are searching the optimum tool combination within biometrics-only condition, to combine FEA and EEG in their experimental set-ups to predict the users' preferences during the online product evaluations. Adding eye tracker to FEA and EEG combination was also suggested for more precise and valid outcomes unless their budget is limited. However, GSR may not be necessary tool since it was not used in the current six classification trees of rating class predictions in biometrics only conditions.

Tool Selection in Hybrid Condition

As the last part of the analysis of tools and variables used in classification trees, the hybrid condition was analyzed. Once both UX approaches are combined, the model specified variables were also changed. It resulted in more heterogeneous sources and variables as seen in Table 23 and Table 24. Enjoyment and interest variables from surveys emotions performed collaborated work with valence, sadness and surprise from FEA, drowsy and workload from EEG, and pupil dilation from ET and resulted with the best accuracies which was slightly higher than biometrics only option. This combination predicted the negatively skewed user ratings better. It was also noted that while biometrics was used only for Product #1, survey-based variables used only in Product #6 during the hybrid version of the analysis.

While Product #1 rating class predicted by using ET and FEA, Product #2 required EEG and Feeling, which may be considered emotions and affective states only. Although both products are in low support groups and their accuracies are 81% and 87%, which are the top two high accuracies, the tool used in their decision tree lacks common variables, as do the common tools.

Table 23: Model Specified Variables and Their Tool of Hybrid (Part 1)

Prediction of Rating Classes (Biometrics + Surveys)						
Support	**Pr ID**	**Prediction error rate in cross-validation**	**Demographics**	**Attitude**	**Feeling**	**UIS**
Low	**1**	19%				
	2	13%			Enjoyment_2	
Medium	**3**	53%		MoneyPractice, SportsPerDay	Interest_3	
	4	35%			Enjoyment_4	
High	**5**	24%		OnlineShopPmont, SportsPerWeek	CoachDrivenS, Enjoyment_5	PhotosVideosA
	6	22%			Interest_6	Descriptions_6

Table 24: Model Specified Variables and Their Tool of Hybrid (Part 2)

Prediction of Rating Classes (Biometrics + Surveys)						
Support	**Pr ID**	**Prediction error rate in cross-validation**	**Eye T**	**EEG**	**FEA**	**GSR**
Low	**1**	19%	PupilLeft_1, sumFixationDuration_1		Sadness_1	
	2	13%		Classification_2, WorkloadBDS_2		
Medium	**3**	53%			LipCornerDepressor_3, Valence_3	GSR_CAL_3,
	4	35%	Surprise_4	Drowsy_4		
High	**5**	24%	PupilLeft_5			
	6	22%				

Meanwhile, Product #6 was the one that required surveys only, particularly Postsurvey only, to predict their rating classes with 24% accuracy. However, Product #5 required ET, attitude and feeling surveys and was the product that required the maximum number of variables and tools in hybrid analysis. “Frequency of coach-driven sport”, “frequency of sports activity/week”, “frequency of online shopping/month”, and “product photos in overall” from attitude survey, “enjoyment from feeling survey” and pupil dilation from ET were variables used in Product #5.

The analysis of tools and variables used in classification trees in the hybrid condition resulted in a little bit different that surveys only and biometrics only conditions. While dominance of attitude and feeling surveys variables in the surveys-only condition and the dominance of FEA and EEG combination in biometrics-only was not as apparent as hybrid conditions. The distribution of variables was more diverse, so does the variations source. In fact, the variable from GSR used in this condition only. The “classification”, “workload” and “drowsy” from EEG, “Pupil dilation” variables from Eye tracker, “Valence” from FEA were observed as common variables both in biometrics only and hybrid conditions. Likewise, “interest” and “enjoyment” from feeling survey, “photos” and “product descriptions” from UIS, “affordable amount money to buy”, and “frequency of sports done in a week” were the common variables used in both surveys only and hybrid conditions.

Predicting User’s First Choice

In the rating class analysis in section 4.2.2, the hybrid conditions resulted in the highest accuracy for the prediction of rating classes by using both traditional surveys and biometrics. The model specified variables of this analysis were selected and used for prediction of the users' first choices within same support levels, namely in low, medium and high levels supports. In each support level, there were two product choices, one from brand 1, the other from brand 2. The model was run to predict the user's first choice. It means which one of the products will be chosen by subjects. The results are in Figure 15 below.

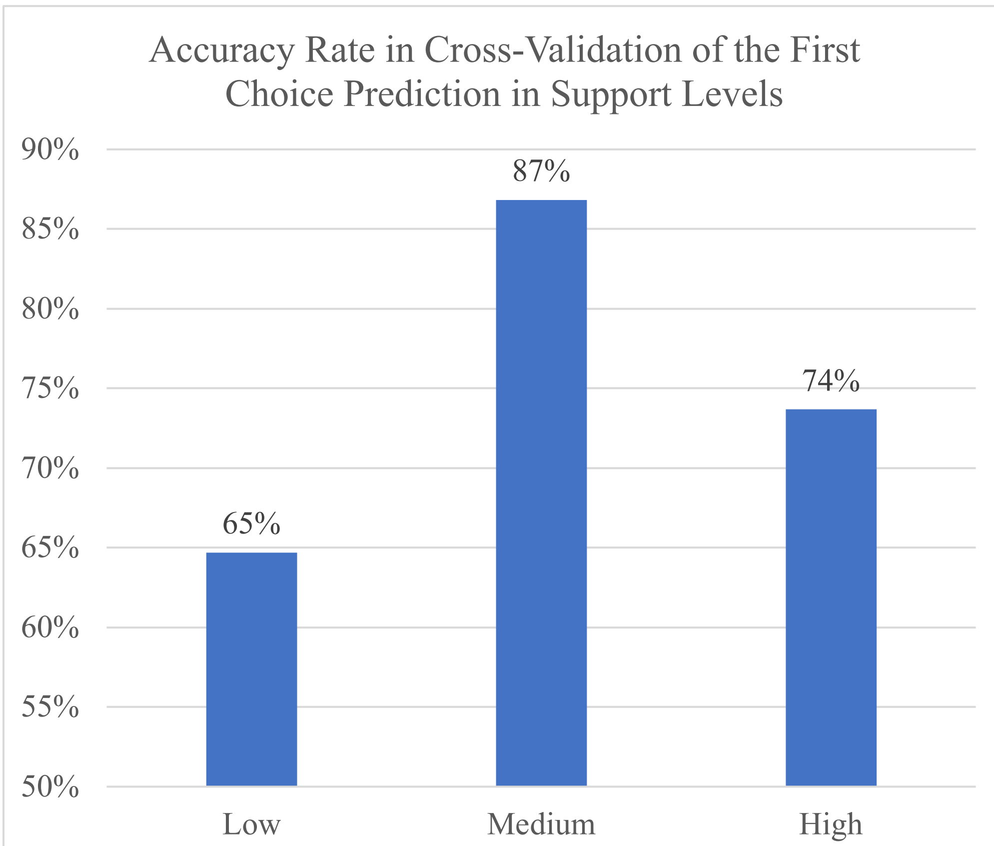


Figure 15: The Accuracy Rate of User's First Choice Prediction

The classification tree model applied in low support level showed 65% accuracy in prediction of user's first choice. Once the random prediction rate of this decision might be considered around 50%, it can be stated that the result of low support level provided more accurate prediction of the first choice. This modest accuracy improvement in low support product sharply transformed to a huge increase once the model was applied on high level support and resulted with 87% accuracy in predicting users first choice. In the high-level support, 74% accuracy was obtained, which means that the model predicted user's first choice correctly every seven of ten prediction attempts.

Once all levels were considered together, although the accuracy rates varied between 65% and 87%, all of them is higher than random guessing which is 50%. Therefore, it can be stated that the use of model specified variables in the prediction of first choices analysis would be beneficial for UX researchers and help them to project and optimize production satisfaction over product cost rate. In addition, once the mean of all three accuracy (75%) are taken together, researchers may have a chance to predict correctly every 8 of 10 times on the foreseen of user's first choice. This ratio also interpreted that it may be 60% more efficient than a random guess option, which is the possibility that half of them may be predicted correctly for each time.

Closure of the Analysis

After all of the processes of this chapter and their findings were recalled, user experience researchers who question the usefulness of multi-sensor affect detection on UX research studies can keep three outcomes in their minds as a summary of results and analysis of the study. First, the study showed that the best accuracy was obtained once both traditional surveys and biometrics conditions were used together to predict user rating classes. However, it is also noted that three current conditions failed to provide accurate results in the prediction of user's rating scores of each product.

Second, according to the results of model specified variables in classification trees of the biometrics only conditions, FEA and EEG were dominated the model specified variables list, Likewise, the results of the survey-only conditions highlighted that feeling surveys, particularly the interest variable, and attitude surveys dominated the used variable list. Therefore, the hybrid experimental setups combining FEA and EEG sensors with the support of traditional surveys might helpful for the higher efficacy of UX evaluation systems.

Third, it can be stated that the use of model specified variables in the prediction of first choices analysis would be beneficial for UX researchers and help them to project and optimize production satisfaction over product cost rate, since the accuracy rates of the classification tree models in 3 different support levels higher than random guessing which is 50%. Overall, the use of model specified variables in the prediction of first choices analysis may prove to be 60% more efficient than a random guess option, which is the possibility that over half may be predicted correctly for each time.

As a result, the study revealed that multi-sensor affect detection systems may take a major role in user experience evaluation systems to solve the current gaps or weaknesses of the traditional usability methods, particularly self-reports, in predictions of user choices since BMS offers complementary solutions to enhance the findings or provide more precise and valid evidence.

In Chapter 5, all these three brief outcomes are explained and discussed in a wider perspective. Investigation of the usefulness of multi-sensor affect detection on user experience through an application of biometric measurement systems on online purchasing is discussed to clarify the research questions in the conclusion section.

Limitations, future works, and overall efficacy of this dissertation are discussed to provide user experience researchers an enhancement of their knowledge and skills for the design of future UX studies and evaluations.

CHAPTER 5

CONCLUSION AND DISCUSSION

This dissertation was a pioneer study of the idea on increasing the efficiency of user experience reports by integrating emerging user insights methodologies and practices (Calvo, D'Mello, Gratch, & Kappas, 2014; Ghergulescu & Muntean, 2014; Yao et al., 2014, Cowley et al., 2016). To that purpose, integration of emotional and cognitive-based user insights into the current UX evaluation systems was experienced in the current study. The study claimed that biometric measurement systems may be vital to user experience evaluation systems to solve the current weaknesses of traditional usability methods in the prediction of user choices. In that light, the study investigated three major research questions: First, it explored how well a sensor-based evaluation approach can corroborate a user's experience as identified by traditional UX methods during an online purchasing task. Second, the study also investigated which model (sensor-based, traditional, or combined) can best predict customer purchasing preferences. Finally, the study attempted to identify which sensors (separately or integrated) most fully explain customer preferences for purchasing.

In the following sections of the final chapter, the conclusion, limitations, opportunities and future works are discussed to evaluate these research questions of the study.

Conclusion

This study revealed three essential pieces of evidence. First piece was regarding the first research question. That means the evidence was related to explain if sensor-based evaluation approach through biometrics tools and software could corroborate user experience identified by traditional UX methods during an online purchasing task and enhanced the efficacy of the UX evaluation systems..

The study also showed that the accuracy rate in cross-validation of rating classes was varied between 51% and 70% once only surveys are used. Although 70% accuracy in low support can be interpreted as a meaningful prediction, the others were not helpful to predict the users' preferences. Thus, the performance of survey-only condition showed weaknesses in the prediction of user choices. However, as shown in Figure 12 in Chapter 4, the use of biometrics results in higher accuracies, particularly in low and high support levels. Although the performance of biometrics in medium support was slightly lower than in the surveys, the accuracy increases in low and high support levels hit to the 75% or above and showed that they could be used to reduce the weakness of surveys-only conditions to predict user preferences. This finding also clarified the first research question and provided evidence that the biometrics-only condition may enhance the reliability and accuracy of UX evaluations and they can be considered as a useful approach, particularly once surveys-only condition showed weaknesses in the prediction of user rating classes during an online purchasing task.

When both systems are used together, the accuracy rate in cross-validation of rating classes reached the highest rate in all support levels, which is another essential evidence related to the second research question investigating which model (sensor-based, traditional, or hybrid) can best explain customer purchasing preferences. In the hybrid condition, the top score (84%) was observed in low support products. However, the sharp increase in the accuracy rate in medium support level was worth more attention. Despite the fact that there were profoundly low accuracy scores both in the survey only (51%) and biometrics only (47%) conditions of the medium support level, the hybrid conditions escalated the accuracy rate in cross-validation of rating classes in medium support level to 66%. This means that although the classification tree model can predict accurately almost every 7 of 10 user preferences in medium support level, based on this cross-validation of rating classes. This also can be interpreted that in the specific cases where traditional survey-based UX methodologies unable to predict user's preferences, integration of sensor-based UX approaches to the current traditional survey-based advance the efficacy of UX methodologies and the collaborative use of both may be useful and efficient in prediction of user preferences.

This study revealed that the use of biometrics during an online purchasing task enhanced the findings of traditional UX methods and provided more accurate evidence to predict consumer purchasing preferences, notably once both measures were used together. Therefore, the contribution of a sensor-based evaluation approach in this study was classified as "complementary" to traditional UX evaluation according to the contributions criteria stated by Kula, Atkinson, Branaghan, and Roscoe in 2017.

Therefore, once all of the above outputs and interpretations are recalled, the first two research questions can be clarified. It is stated that a sensor-based evaluation approach through biometrics tools and software could corroborate user experience identified by traditional UX methods during an online purchasing task and enhanced the efficacy of the UX evaluation systems. Also, the collaborative use of both sensor-based and traditional approaches can best be used to explain customer preferences for purchasing, due to synergetic effects between the two approaches.

User experience researchers who question the usefulness of multi-sensor affect detection on UX research studies can use this evidence to plan and design their UX experiments by including biometrics approaches to their UX evaluations systems. However, they also need to consider which sensors, whether separately or integrated, can best be used to explain customer preferences for purchasing. The study investigated this question and provided some evidence as well.

It was shown that a combination of biometrics reduced the prediction error rate in cross-validation in the classification tree analysis. However, this does not mean that the more these approaches are combined, then the higher accuracy in the prediction of user ratings. Surprisingly, the study showed that none of the variables from GSR were used in tree constructions in the biometrics-only condition as shown in Table 19. Overall, FEA variables are not only used in most products but also, they are the top in the number of model specified variables for each prediction classes of the six products. Therefore, the study clearly pointed out that FEA variables played a major role in the construction of classification trees for the prediction of user preferences, followed by EEG, and then the eye tracking variables.

Also, the study highlighted that combination of biometric measures was observed in most of the decision trees to predict user preferences. For instance, the combination of FEA and EEG either explained full branches of classification trees such as Product #1 and #5 or dominated the tree constructions. Thus, the last research question can be clarified and it can be stated that it is worth giving more attention to the use of FEA and EEG combination for the optimum efficacy of biometric approaches to predict customer preferences during an online purchasing task. FEA and eye tracking combination or EEG and eye tracking combination may also be considered as other alternatives. The further interpretations of the model specified variables were discussed in opportunities and discussions sections.

Besides, the implementation of the model specified variables in the prediction of first choices analysis output higher efficacy (75% in overall) than random guessing (50%). This means that researchers may have a chance to predict correctly every eight out of ten times on the foreseen of user's first choice. This ratio also interpreted that it may be 60% more efficient than a random guess option, which is the possibility that half may be predicted correctly for each time. Therefore, it can be stated that the use of model specified variables in the prediction of first choices analysis would be beneficial for UX researchers and help them to project and optimize production satisfaction over product cost rate.

In conclusion, the use of multi-sensor affect detection systems, particularly application of biometric measurement systems on online purchasing analysis can be vital for user experience evaluation systems to solve the current gaps or weaknesses of the traditional usability methods, particularly self-reports, in predictions of user choices. BMS offers complementary solutions to enhance the findings or provide more precise and valid evidence. Moreover, hybrid experimental setups combining FEA and EEG sensors with support of traditional surveys are recommended for the higher efficacy of UX evaluation systems.

Limitations

The analysis section the study aims to prediction of user's preferences, particularly projection of the products which consumers most likely to buy. In this light, this phasefocused on revealing model-specified variables, including the required tools from both surveys and biometric UX approaches to predict their preferences. According to the analysis of rating classes and ranking scores for their first choices, the current scope of the study was efficient to clarify the research questions and provide accurate evidence for researchers to enhance their UX evaluations' experimental designs for further research studies. Thus, the analysis did not include detailed comparisons of emotion dynamics that can be gathered through advanced statistical analysis such as time series approaches. Also, the reasoning of subjects explaining why they rate each product through self-reports was excluded in the study because of the time and cost limitations.

The study also has some more limitations that might be considered. First, participants might not fully represent the populations of interest. Participation was limited to females between 18 and 35 years old who were living in the Phoenix and Portland metropolitan areas. This might be a good sample of a typical western population in the United States. However, the sample might not fully represent the full customer population who does online purchasing. The sample may cause some bias or mismatches with the desired variety of population to generalize the results of the study. The number and variety of participants might be increased by conducting the experiments in different locations not only within the US, but also other parts of the world in future research. Thus, it represents better population for the accurate generalization of the results and their interpretations.

The second limitation was technical in nature. Research stimuli were displayed with a 22-inch extended monitor connected to a laptop. Therefore, the results were limited to the device hardware and software specifications, and the resolution and displays ratio and sizes of the selected websites might differ on other media devices such as other computers, tablets, and smartphones. In further research opportunities, different devices can also be included such as tablet, smartphone or laptop devices.

Third, there were some technical concerns that had to be considered during data collection. It is assumed that the broadband quality would not radically change, and the wireless Internet connection would be almost the same through all the data collection sessions of the study. However, an unexpected network drop might impact data collection quality, specifically for the BMS.

Fourth, experimental errors and the technical limitations or challenges of the BMS devices were important considerations regarding the accuracy and reliability of data. Artifacts and noise on EEG and GSR data, validation concerns in facial expression analysis, and lack of the exact moment response detection on GSR were the primary potential sources of experimental error. However, all these risks were minimized with efficient data quality measures through BMS and data cleaning techniques during data collection sessions. Then, a detailed data cleaning has been processed to reduce the invalid or noise data and only high-quality data were used for further statistical approaches.

Another limitation was using the application of a gel-based EEG headset, which made participants' hair temporarily untidy and "jelly." At the end of each session, a dry shampoo is offered in case participants might feel uncomfortable and even unhappy after the EEG sessions.

Finally, biometric research tools provided “big data,” which requires extra attention and expertise. However, unrecognized errors might still exist and any unexpected process during data mining could increase the risk of type II errors, so the researcher used extraordinary attention both during data collection and analysis to get reliable and valid results.

Discussions, Opportunities and Future Work

The conclusions and limitations of the study also opened discussions for new opportunities to improve UX evaluations and consumers insights. This study has the high potential for future innovations and methodologies to measure the effect of consumer emotions on shopping preferences. For example, this study could be used as an alternative tool for the replacement or improvement of the Net Promoter Score (NPS), which is a well-known but also well-criticized user satisfaction measurement through one question developed by Reichheld in 2003. Although NPS was highly innovative, particularly for corporations looking for agile consumer satisfaction measurements in the first decade of the millennium age (Reichheld & Markey, 2011), it seems that NPS is severely limited, since it is mostly focused on rational aspects of an experience and ignores emotional ones (Shaw, 2007, pp. 120-135).

In fact, there’s already a developed solution, called Net Emotional Value (NEV) to fill that gap. Colin Shaw, in 2007, developed a new measure which is a single number that represents a person’s emotional value. It is calculated by subtracting positive emotions from negative emotions. The higher number, then the more loyal relationships a business has. However, NEV is still conducted through self-reports and it includes the potential risk that is mentioned in the literature background chapter. Since consumer insights cannot always be captured through self-reports, neither NPS nor NEV alone may be a comprehensive solution. However, integration of multisensory-based affect detection systems to consumer satisfaction measurements may be a game changer for future designs, and they can be validated as an efficient tool to optimize the product satisfaction/cost ratio for both consumers and producers. The suitability of the rating class formation of the current study might provide another opportunity to combine NPS and NEV to create a mutually beneficial customer satisfaction measurement system.

The study was designed to compare the two products in three separate groups. As a next step, focusing on one support level with a multivariate product test would be recommended to investigate the user's first choices. In addition, it may reveal evidence more on how various user interface features related to each other.

Besides the potential of the current study on futures user satisfaction measures, the study can be advanced by additional design plans and experimental implementations. For example, the current study investigated the roles of 30 biometrics variables, but the total numbers of variables that BMS output is more than 130, most of which are captured through EEG. The used EEG variables in this study were seven affective state constructs which are automated output of ABM X-10 EEG device. However, it also allows researchers to obtain frontal asymmetry index, which is a marker of approach and avoidance. It is found that higher left-frontal activity compared to right is an indicator of approach-oriented, a sign of motivation and positive feelings, such as joy (Coan & Allen, 2003), while higher right-frontal activity is an indicator of motivation withdrawal and negative feeling, such as sadness and fear (Harmon-Jones, Gable, & Peterson, 2010).

Frontal asymmetry index allows researchers to measure how the brain responds to the new product or stimuli. The analysis of motivation dynamics of users allows researchers to validate their attitude towards given product and provide evidence if they would be likely to buy or not. This EEG variable can be crucial particularly once the traditional approaches fail such as the cases of biased surveys, mistreated samples or the existence of code error caused by researchers.

Also, the current output might be enhanced by performing an additional analysis comparing the affective and cognitive dynamics of the sensor variables. The limitations and issues of traditional approach were explained in background literature in Chapter 2. It has been stated that traditional UX approached such as surveys are unable to capture users' real-time cognitive and affective dynamics and also it might be a challenge to express users' thoughts and behaviors (Podsakoff et al., 2003; Vermeeren et al., 2010), their visual attention (Jarodzka, Scheiter, Gerjets & van Gog, 2010; Tsai, Hou, Lai, Liu, & Yang, 2012), cognitive workload (Dirican & Göktürk, 2011), and emotions (Boucsein, 1992; Cohn & De la Torre, 2014). However, multisensory based affect detection systems allow a researcher to explore real-time dynamics of users' emotional and cognitive states during the stimuli (Cowley et al., 2016). The output of this analysis may help to address the location of the specific changes and indicated significant differences between variables. However, the current analysis of the study provided required evidence to predict the users' preferences, and the additional contributions are not planned due to the time and cost limitations of the study.

There is no doubt that future work and studies would be mutually beneficial to validate the current outcomes and also advance the contributions of the usefulness of multi-sensor affect detection on user experience. Like the application of biometric measurement systems on online purchasing, similar experimental research can be designed by using real products and the results of physical product examination environment and online purchasing environment may be compared for this purpose. Thus, it would be a highly remarkable study to investigate if the proven complimentary impacts of multi-sensor based UX approaches would differ during the physical product evaluations.

Also, once the model specified variables in the classification trees and their sources are recalled, the most interesting question can be which tool or tools should be taken into consideration for better UX evaluations. The answer is not quick and easy, however, based on a budget of the study, expertise skills of the researchers, and scope of the target population and the research questions of the study may figure out the optimum tool combinations. Just for simplicity's sake, assume that all suitable conditions have been met and the study focuses on the sensor efficacy only. According to the results of model specified variables in classification trees of the biometrics-only conditions, FEA and EEG were dominated the model specified variables list. Likewise, the results of the survey only conditions highlighted that feeling surveys, particularly interest variable, and attitude surveys dominated the used variable list. In the hybrid condition, while sensor-only variables distributed results almost equally and so that it is a challenge to point out specific tools. However, measures of feeling survey and attitude survey was higher than the rest of the other tool sources in the hybrid survey. Therefore, it is recommended to take considerations of FEA and EEG combinations supported by attitude and feeling surveys to predict the user preferences during an online product examinations' task. However, in some limited cases, where research budgets would be an issue, the FEA would be a modest but also an effective solution according to the findings of this study.

As a result, this thesis investigated the usefulness of the of multi-sensor affect detection on user experience through an application of biometric measurement systems on online purchasing. The study revealed that multi sensor affect detection systems are vital on user experience evaluation systems to solve the current gaps or weaknesses of the traditional usability methods, particularly self-reports, in predictions of user choices since BMS offers complementary solutions to enhance the findings or provide more precise and valid evidence. Moreover, hybrid experimental setups combining FEA and EEG sensors with the support of traditional surveys are recommended for the higher efficacy of UX evaluation systems. The use of model specified variables in the prediction of first choice analysis would be beneficial for UX researchers and help them to projectile and optimize production satisfaction over product cost rate.

.

REFERENCES

Albert, W. & Tullis, T. (2013). Measuring the user experience: Collecting, analyzing, and presenting usability variables (2nd ed.). Burlington, MA: Morgan Kaufman.

Baker, R. S., Ocumpaugh, J., Gowda, S. M., Kamarainen, A. M., & Metcalf, S. J. (2014). Extending log-based affect detection to a multi-user virtual environment for science. In Dimitrova V., Kuflik T., Chin D., Ricci F., Dolog P., & Houben GJ. (eds.) User Modeling, Adaptation, and Personalization. UMAP 2014. Lecture Notes in Computer Science, 8538. Cham, Switzerland: Springer International Publishing.

Baxter, K., Courage, C., & Caine, K. (2015). Understanding Your Users: A Practical Guide to User Research Methods. Elsevier Science. Retrieved from https://books.google.com/books?id=l9-cBAAAQBAJ

Berka, C., Levendowski, D. J., Lumicao, M. N., Yau, A., Davis, G., Zivkovic, V. T., Olmstead, R.E., Craven, P. L., & Tremoulte, P. (2007). EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. Aviation, Space and Environmental Medicine, 78 (Supplement 1), B231-B244.

Beyer, H., & Holtzblatt, K. (1997). Contextual design: Defining customer-centered systems Interactive Technology). Amsterdam, The Netherlands: Elsevier.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

Bosch, N., Chen, Y., & D'Mello, S. (2014). It's written on your face: Detecting affective states from facial expressions while learning computer programming. In Trausan-Matu S., Boyer K.E., Crosby M., & Panourgia K. (eds.) Intelligent Tutoring Systems. ITS 2014. Lecture Notes in Computer Science, 8474. Cham, Switzerland: Springer International Publishing.

Brookings, J. B., Wilson, G. F., & Swain, C. R. (1996). Psychophysiological responses to changes in workload during simulated air traffic control. *Biological Psychology*, *42*(3), 361-377.

Coan, J. A., & Allen, J. J. (2003). Frontal EEG asymmetry and the behavioral activation and inhibition systems. *Psychophysiology, 40*(1), 106-114.

Calvo, R. A., D'Mello, S., Gratch, J., & Kappas, A. (Eds.). (2014). *The Oxford Handbook of Affective Computing.* Oxford, England: Oxford University Press.

Calvo, R. A. & D'Mello, S. (2010). Affect recognition: An interdisciplinary review of models, methods, and their applications. IEEE *Transactions on Affective Computing 1*(1), 18-37.

Calvo, R. A. & D’Mello, A. C. (2012). Frontiers of affect-aware learning technologies. *IEEE Intelligent Systems, 27*(6), 86-89.

Caroux, L., Le Bigot, L., & Vibert, N. (2013). Impact of the motion and visual complexity of the background on players' performance in video game-like displays. *Ergonomics*, *56*(12), 1863-1876.

D’Mello, S., Jackson, G., Craig, S., Morgan, B., Chip-man, P., White, H., Person, N., Kort, B., El Kaliouby, R, Picard, R., & Graesser, A. (2008,). AutoTutor detects and responds to learners [sic.] affective and cognitive states. In Workshop on emotional and cognitive issues at the international conference on intelligent tutoring systems. Montreal, Canada.

Dalmaijer, E. S., Mathôt, S., & Van der Stigchel, S. (2014). PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eye-tracking experiments. *Behavior research methods*, *46*(4), 913-921.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802-813.

Foglia, P., Prete, C. A., & Zanda, M. (2008). Relating GSR signals to traditional usability variables: Case study with an anthropomorphic web assistant. In IEEE Instrumentation and Measurement Technology Conference.

Gaver, W. W. & Martin, H. (2000). Alternatives: Exploring information appliances through conceptual design proposals. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. The Hague, The Netherlands.

Gevins, A., Smith, M. E., McEvoy, L., & Yu, D. (1997). High-resolution EEG mapping of cortical activation related to working memory: Effects of task difficulty, type of processing, and practice. *Cerebral Cortex*, *7*(4), 374-385.

Ghergulescu, I., & Muntean, C. H. (2014). A novel sensor-based methodology for learner’s [sic] motivation analysis in game-based learning. *Interacting with Computers*, *26*(4), 305-320.

Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Green, W., Dunn, G., & Hoonhout, J. (2008). Developing the scale adoption framework for evaluation (SAFE). In Proceedings of the International Workshop on Meaningful Measures, Reykjavik, Iceland.

Hanington, B., & Martin, B. (2012). Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions. Rockport Publishers.

Harmon-Jones, E., Gable, P. A., & Peterson, C. K. (2010). The role of asymmetric frontal cortical activity in emotion-related phenomena: A review and update. *Biological psychology, 84*(3), 451-462.

Hassenzahl, M., & Tractinsky, N. (2006). User experience–A research agenda. *Behavior & Information Technology*, *25*(2), 91–97.

Hassenzahl, M., Burmester, M., & Koller, F. (2003). AttrakDiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer qualität. In Swillus, G. & Ziegler, J. (eds.), *Mensch & Computer 2003.* Wiesbaden, Germany: Springer International Publishing.

He, W., Boesveldt, S., de Graaf, C., & de Wijk, R. A. (2016). The relation between continuous and discrete emotional responses to food odors with facial expressions and non-verbal reports. *Food Quality and Preference*, *48*(A), 130-137.

Healey, J. (2014). Physiological Sensing of Emotion. In Calvo, R., D'Mello, S., Gratch, J., & Kappas, A. (eds.), *The Oxford Handbook of Affective Computing*. Oxford, England: Oxford University Press.

Holtzblatt, K., Wendell, J. B., & Wood, S. (2004). *Rapid contextual design: A how-to guide to key techniques for user-centered design*. Amsterdam, The Netherlands: Elsevier.

Hoque, M.E., Mohammed, & Picard, R. W. (2011). Acted vs. natural frustration and delight: Many people smile in natural frustration. IEEE International Conference on Automatic Face and Gesture Recognition and Workshops.

iMotions A/S (2016). Human Behavior. Retrieved from https://imotions.com/guides/ on [03/03/2017].

ISO DIS 9241-210:2008. Ergonomics of human system interaction - Part 210: Human-centred design for interactive systems (formerly known as 13407). International Organization for Standardization (ISO). Switzerland.

Kaatiala, J., Yrttiaho, S., Forssman, L., Perdue, K., & Leppänen, J. (2014). A graphical user interface for infant ERP analysis. *Behavior Research Methods*, *46*(3), 745-757.

Kahneman, D. (1973). *Attention and effort.* Englewood Cliffs, NJ: Prentice-Hall.

Kula, I., Branaghan, R. J., Atkinson, R., K., & Roscoe, R. D. (2017). Assessing user experience via biometric sensor affect detection. In R. D. Roscoe, S. D. Craig, & I. Douglas (Eds.), End-user considerations in educational technology design (pp. 123-140). Hershey, PA: IGI Global.

Kuniavsky, M. (2010). *Smart things: Ubiquitous computing user experience design: Ubiquitous computing user experience design*. Amsterdam, The Netherlands: Elsevier.

Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In Symposium of the Austrian HCI and Usability Engineering Group. Berlin, Germany: Springer International Publishing.

Lenzner, T., Kaczmirek, L. & Galesic, M. (2014). Left feels right! A usability study on the position of answer boxes in web surveys. *Social Science Computer Review*, *32*(6), 743-764.

Loh, W. Y. (2011). Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1), 14-23.

Loh WY. Fifty years of classification and regression trees. Int Stat Rev. 2014;82(3): 329–348. doi: 10.1111/insr.12016.

Loyola, P., Martinez, G., Muñoz, K., Velásquez, J. D., Maldonado, P., & Couve, A. (2015). Combining eye tracking and pupillary dilation analysis to identify Website Key Objects. *Neurocomputing*, *168*, 179-189.

Makeig, S. (1993). Auditory event-related dynamics of the EEG spectrum and effects of exposure to tones. *Electroencephalography and clinical neurophysiology*, *86*(4), 283-293.

Makeig, S., & Jung, T. P. (1996). Tonic, phasic, and transient EEG correlates of auditory awareness in drowsiness. *Cognitive Brain Research*, *4*(1), 15-25.

Mandryk, R. L., Atkins, M. S., & Inkpen, K. M. (2006). A continuous and objective evaluation of emotional experience with interactive play environments. Proceedings of the SIGCI Conference on Human Factors in Computing Systems. Montreal, Canada.

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Ames, M., & Ledered, S. (2003). Heuristic evaluation of ambient displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Ft. Lauderdale, FL

Masaki, H., Ohira, M., Uwano, H., & Matsumoto, K. I. (2011). A quantitative evaluation on the software use experience with electroencephalogram. A. Marcus, A. (ed.)

Design, User Experience, and Usability. Theory, Methods, Tools and Practice. Berlin, Germany: Springer International Publishing.

Mirza-Babaei, P. & Nacke, L. (2015). Introducing the biometric storyboards tool for games user research. In Games Media Entertainment. IEEE Xplore. Fabric of Digital Life. https://fabricofdigitallife.com/index.php/Detail/objects/1194

Mirza-Babaei, Long, S., Foley, E., & McAllister, G. (2011). Understanding the contribution of biometrics to games user research. In Proceedins of DiGRA 2011 Conference: Think Design Play.

Monk, A. (2002). Noddy's guide to usability. *Interfaces*, *50*, 31-33.

Nielsen, J. (2012). How many test users in a usability study? *Nielsen Norman Group, 4*(06).

Norman, D. A. & Draper, S. W. (Eds.) (1986), *User centered system design: New perspectives on human-computer interaction.* Hillsdale, NJ: Erlbaum Associates.

Patterson, E. & Erturk, E. (2015) An inquiry into agile and innovative user experience (UX) design. (2015). Proceedings of the 6th Annual Conference of Computing and Information Technology Education and Research in New Zealand incorporating the 28th Annual Conference of the National Advisory Committee on Computing Qualifications, Queenstown, New Zealand.

Picard, R. W. (2010). Emotion research by the people, for the people. *Emotion Review*, *2*(3), 250-254. (1997). *Affective computing.* Cambridge, MA: MIT Press.

Pleydell-Pearce, C. W., Whitecross, S. E. & Dickson, B. T. (2003). Multivariate analysis of EEG: Predicting cognition on the basis of frequency decomposition, inter-electrode correlation, coherence, cross phase and cross power. *System Sciences*. 2003. Proceedings of the 36th Annual Hawaii International Conference. IEEE.

Ramsey, Z. T. (2014). Neuromarketing and consumer neuroscience: Current understanding and the way forward. *Decision*, *43*(4), 457-462.

Reichheld, F., & Markey, R. (2011). *The ultimate question 2.0 (Revised and expanded edition): How net promoter companies thrive in a customer-driven world.* Cambridge, Mass: Harvard Business Publishing.

Shaw, C. (2007). *The DNA of customer experience: How emotions drive value*. Springer.

Soloway, E., Guzdial, M., & Hay, K. E. (1994). Learner-centered design: The challenge for HCI in the 21st century. interactions, *1*(2), 36-48.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.

Sterman, M. B., & Mann, C. A. (1995). Concepts and applications of EEG analysis in aviation performance evaluation. *Biological Psychology*, *40*(1), 115-130.

Stikic, M., Johnson, R. R., Levendowski, D. J., Popovic, D. P., Olmstead, R. E., & Berka, C. (2011). EEG-derived estimators of present and future cognitive performance. *Frontiers in Human Neuroscience*, *5*(70).

Therneau, T. M., & Atkinson, E. J. (1997). An introduction to recursive partitioning using the RPART routines.

Toker, D., Steichen, B., Gingerich, M., Conati, C., & Carenini, G. (2014). Towards facilitating user skill acquisition: Identifying untrained visualization users through eye tracking. In Proceedings of the 19th international conference on Intelligent User Interfaces, Haifa, Israel.

Tory, M. & Kincaid, R. (2013). Comparing physical, overlay, and touch screen parameter controls. In Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces, St. Andrews, Scotland, UK.

Veloso, A., Meira Jr, W., & Zaki, M. J. (2006, December). Lazy associative classification. In Data Mining, 2006. ICDM'06. Sixth International Conference on (pp. 645-654). IEEE.

Vermeeren, A. P., Law, E. L. C., Roto, V., Obrist, M., Hoonhout, J., & Väänänen-Vainio-Mattila, K. (2010, October). User experience evaluation methods: current state and development needs. In Proceedings of the 6th Nordic Conference on Human-Computer Interaction. Reykjavik, Iceland.

Wang, X. W., Nie, D. & Lu, B. L. (2014). Emotional state classification from EEG data using machine learning approach. *Neurocomputing*, *129*, 94-106.

Ward, R. D. & Marsden, P. H. (2003). Physiological responses to different WEB page designs. *International Journal of Human-Computer Studies*, *59*(1), 199-212.

Whitehill, J., Serpell, Z., Foster, A., Lin, Y. C., Pearson, B., Bartlett, M., & Movellan, J. (2011). Towards an optimal affect-sensitive instructional system of cognitive skills. Proccedures of IEEE Confonference on Computer Vision and Pattern Recognition Workshop on Human Communicative Behavior.

Wolpin, S. E., Halpenny, B., Whitman, G., McReynolds, J., Stewart, M., Lober, W. B., & Berry, D. L. (2014). Development and usability testing of a web-based cancer symptom and quality-of-life support intervention. *Health Education Journal*, 21(1), 10-23.

Yao, L., Liu, Y., Li, W., Zhou, L., Ge, Y., Chai, J., & Sun, X. (2014). Using Physiological Measures to Evaluate User Experience of Mobile Applications. In Harris, D. (ed.) *Engineering Psychology and Cognitive Ergonomics.* New York, NY: Springer International Publishing.

Yayici, Emrah (2014). *UX Design and Usability Mentor Book : With Best Practice Business Analysis and User Interface Design Tips and Techniques*. Kindle Edition.

Zaharias, P. (2004). Usability and e-learning: The road towards integration. *eLearn*, (*6)*, 4.

Zaharias, P., & Poylymenakou, A. (2009). Developing a usability evaluation method for e-learning applications: Beyond functional usability. *International Journal of Human–Computer Interaction*, *25*(1), 75-98.

APPENDIX A

DEMOGRAPHIC AND ATTITUDE SURVEY

In this survey, we ask questions about your demographic background and experience relates to online shopping. Please remember that these data are collected anonymously and will not be linked to your name or other identifying details.

Please answer honestly. There are no right or wrong answers. When you are ready to begin, type your study ID below and start your responses.

Participant ID: .... *(Type your participant ID given by the researcher)*

1. Age - *Please type the year that you were born.*
   - ………

2. Gender - *Type your gender.*
   - ………

3. Ethnicity origin (or Race) - *Please specify your ethnicity.*
   - White
   - Hispanic or Latino
   - Black or African American
   - Native American or American Indian
   - Asian / Pacific Islander
   - If the options above do not fit well for you, please describe your ethnicity: ………….
   - Decline to answer

4. Highest Level of Education You Have Currently Enrolled - *Mark only one oval.*
   - Freshman
   - Sophomore
   - Junior
   - Senior
   - Bachelor's Degree
   - Master's Degree
   - Ph.D.
   - Decline to answer

In the following 4 questions, we would like to hear your online shopping habits or preferences. In that purpose, you will see questions asking the frequency of your Internet usage per day, an occurrence of your online shopping activities per month, the device options that you usually use for online shopping and the websites that you preferred for online shopping.

Please answer all these questions very honestly. This information will not be used in the individual base. They will be aggregated to describe overall behaviors of customers who do online shopping.

5. How often do you use the Internet in a day? - *Use the slider to select the appropriate number. 0 is Never, 24 is 24 hours in a day*
   - Never 0 3 6 9 12 15 18 21 24 *(Slider range from 0 to 24)*

6. How often do you do online shopping? - *Use the slider to select the appropriate number. 0 is Not at all likely, 10+ is ten times or more in a month.*
   - Not at all (0) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10+) Ten times or more in a month

7. How often do you usually use the following devices for online shopping? - *Use the slider to select the appropriate number. 0 is Not at all likely, 10+ is ten times or more in a month.*
   - Desktop Computer
     - Not at all likely (0) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10+) Ten hours or more in a month
   - Laptop Computer
     - Not at all likely (0) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10+) Ten hours or more in a month
   - Tablet
     - Not at all likely (0) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10+) Ten hours or more in a month
   - Smartphone
     - Not at all likely (0) (1) (2) (3) (4) (5) (6) (7) (8) (9) (10+) Ten hours or more in a month

8. What are your top three websites for online shopping? - *Write your top 3 websites for online shopping.*
   1. ……………….
   2. ……………….
   3. ……………….

In the last part of this survey, we aim to learn your stances about a specific product groups, namely sports bras. Though the product group has a lot of varieties and models, we would like to learn your general attitudes related to your frequency of use, the money that you can pay and the parts of websites for online shopping. Please help us by taking a few minutes to tell us about your attitudes on them so far. We appreciate your participation in the study.

When you are ready to begin, type your study ID below and start your responses.

Study ID: .... *(Type your study ID given by the researcher)*

1. How often do you typically use sports bra? - *Mark only one oval.*
- Everyday
- 3 or less in a week.
- 3 or less in a month
- 1-3 times in a year
- Never

2. How much can you pay for a sports bra?
- I can pay for a sports bra max $ .... *(Type the amount of money)*

3. When you are shopping online how important are the following elements of the shopping website for your decision to buy sports bras?

Elements of Website	**How important the elements of the online shopping website to buy sports bras?** (0: Not at all likely and 10: Extremely important)
Product photos and videos	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Product descriptions	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Customer ratings and comments	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
User comments	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Product Cost	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*

APPENDIX B

USER EXPERIENCE RATING SURVEY

Thank you for examining one of the product choices. Now, please share your thoughts and feelings about if you are likely to buy or eliminate it.

As it was mentioned in the consent form, your data will only be reported in aggregate or summarized form and your responses are anonymous. Please answer honestly. We appreciate your participation in the study. When you are ready to begin, type your participant ID below and start your responses.

Participant ID: .... *(Type your participant ID given by the researcher)*

Your Product Ratings to Buy or Eliminate It:
The name of the product: .... *(Select from the drop-down list)*

1. How likely are you to buy it? *(Slider range from 0 to 10. 0 is Not at all likely to buy and 10 is Extremely likely to buy)*
   - The overall rating for this product is 0 1 2 3 4 5 6 7 8 9 10

2. Could you please explain why you preferred this rating? *(Open-ended short paragraph)*
   ………………………………

Your Ratings on the Usefulness of the Website Parts:

3. How much do you feel useful or useless with the following elements of the website? *(Slider range from 0 to 10)*

The Elements of the Website	**How much do you feel useful or useless with the following elements of the website? (0: Strongly useless and 10: Strongly useful)**
Product photos and videos	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Customer ratings and comments	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Product descriptions	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*
Prices	0 1 2 3 4 5 6 7 8 9 10 *(Slider range from 0 to 10)*

Your Feelings During the Product Examination:

4. While examining the specific product choice, how strongly did you feel the following emotions? *(Slider range from 0 to 10)*

Feelings	**Rating of Your Feeling: While examining the examining the specific product choice, how strongly did you feel the following emotions?**

	(0: Not at all likely and 10: Strongly felt)										
Confusion	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*
Distraction	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*
Enjoyment	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*
Interest	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*
Tiredness	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*
Frustration	0	1	2	3	4	5	6	7	8	9	10 *(Slider range from 0 to 10)*

Note that you will not be allowed to back to this evaluation again. So please make sure that you are highly confident with your current responses. If not, please go back to the questions, think again on them and revise your responses to be confident.

SUBMIT

You have now completed the User Experience Rating Survey of the study. Thank you very much for your participation!

We greatly appreciate the time and effort you have put into this study.

APPENDIX C

USER EXPERIENCE RANKING SURVEY

Now, it is time to decide which product you would like to buy. In the given table, I put the sample photo and name of each product to help you for recalling them from your memory. Please think carefully about all sports bras that you have just examined and rated, compare them in your mind, and then respond the following three questions to reflect your final decisions to buy or eliminate them. We appreciate your participation in the study. When you are ready to begin, type your study ID below and start your responses.

Participant ID: .... *(Type your participant ID given by the researcher)*

1. Could you please adjust the following ranking list according to your decision preference to buy one of them? (Note that, while #1 or the first choice at the top of the list refers to the choice which you most likely to buy, the #6 or the last choice at the bottom of the list means the choice which you less likely to buy.) *(Sorting list*)
    - Strappy
    - Techfit
    - Climachill
    - Indy Wipeout
    - Pro Classic
    - Pro Rival

2. How much do you agree or disagree with the following statement about your decision?
   Statement: I made the best decision to select.

    a) Strongly agree
    b) Agree
    c) Somewhat agree
    d) Neither agree nor disagree
    e) Somewhat disagree
    f) Disagree
    g) Strongly disagree

3. Could you please explain why did you prefer this ranking? *(Open-ended short paragraph)*
   ………………………………

APPENDIX D

PREDICTION OF RATING FOR PRODUCT 1

RatingP1A1RT3.R
irfan
Thu Mar 1 23:52:05 2018

```
#UoB Rating Analysis
#RatingP1A1RT-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)

# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P1/A1/Model")

# Upload the data
RatingP1A1Mdata <- read.csv("UoB.v12.csv", header=TRUE)

#Display header names
#names(RatingP1A1Mdata)

# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP1A1Mdata2 <- RatingP1A1Mdata[c(1,47,48,151:180)]
names(RatingP1A1Mdata2)
```

```
##  [1] "Subject"              "ProductID_1"
##  [3] "Rating_1"             "PupilLeft_1"
##  [5] "PupilRight_1"         "minFixationStart_1"
```

```
## [7] "sumFixationDuration_1" "Classification_1"
## [9] "HighEngagement_1"      "LowEngagement_1"
## [11] "Distraction_1.1"      "Drowsy_1"
## [13] "WorkloadFBDS_1"       "WorkloadBDS_1"
## [15] "WorkloadAverage_1"    "BrowFurrow_1"
## [17] "BrowRaise_1"          "LipCornerDepressor_1"
## [19] "Smile_1"              "Valence_1"
## [21] "Attention_1"          "Anger_1"
## [23] "Sadness_1"            "Disgust_1"
## [25] "Joy_1"                "Surprise_1"
## [27] "Fear_1"               "Contempt_1"
## [29] "PeakCount_1"          "Peak.Min_1"
## [31] "AveAmplitude_1"       "MaxAmplitude_1"
## [33] "GSR_CAL_1"
#If Rating_1 have NAs, omit these rows that contain NA values
#RatingP1A1Mdata3 <- RatingP1A1Mdata2[!is.na(RatingP1A1Mdata2$PeakCount_1),]
RatingP1A1Mdata3 <- RatingP1A1Mdata2[!is.na(RatingP1A1Mdata2$Rating_1),]

plot.default(RatingP1A1Mdata3$Rating_1, main = "Rating Scores For P1",
       xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP1A1Mdata3$Rating_1, main = "Histogram of Rating Scores For P1",
       xlab = "Rating", ylab = "Frequency")
```

```
#Create training and test data
str(RatingP1A1Mdata3)
## 'data.frame':    46 obs. of  33 variables:
##  $ Subject             : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
## $ ProductID_1         : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Rating_1            : int  10 7 2 4 5 1 8 3 3 2 ...
## $ PupilLeft_1         : num  2.84 2.45 3.31 2.45 2.56 ...
## $ PupilRight_1        : num  2.8 2.43 3.4 2.61 2.49 ...
## $ minFixationStart_1   : int  251 328 153 134 233 90 0 230 22 1041 ...
## $ sumFixationDuration_1: int  7901952 5121529 13311952 12329417 9887573 1178
4199 5910785 10168494 8764840 4599975 ...
## $ Classification_1     : num  0.607 0.7 0.802 0.76 0.784 ...
## $ HighEngagement_1     : num  0.271 0.392 0.695 0.547 0.638 ...
## $ LowEngagement_1      : num  0.414 0.591 0.268 0.429 0.338 ...
## $ Distraction_1.1      : num  0.26805 0.01703 0 0.02395 0.00835 ...
## $ Drowsy_1             : num  0.04756 0.000199 0.036427 0 0.015909 ...
## $ WorkloadFBDS_1       : num  0.828 0.511 0.565 0.491 0.624 ...
## $ WorkloadBDS_1        : num  0.716 0.458 0.519 0.457 0.561 ...
## $ WorkloadAverage_1    : num  0.772 0.484 0.542 0.474 0.593 ...
## $ BrowFurrow_1         : num  2.51e-05 8.81e-05 6.52e-06 4.90e-05 1.10e-06 ...
## $ BrowRaise_1          : num  7.493 15.913 1.07 0.363 4.495 ...
## $ LipCornerDepressor_1 : num  0.00203 0.00866 0.36181 0.04757 0.97588 ...
## $ Smile_1              : num  5.98e-02 4.20e-05 1.13e-05 4.97e-11 2.56e-06 ...
## $ Valence_1            : num  0 0 -0.632 0 -0.344 ...
## $ Attention_1          : num  93.5 98.5 94.9 92.7 96.5 ...
## $ Anger_1              : num  0.00252 0.02918 0.00819 0.00193 0.00173 ...
## $ Sadness_1            : num  0.013209 0.000679 0.019041 0.024512 0.022348 ...
## $ Disgust_1            : num  0.498 0.427 0.599 0.439 0.432 ...
## $ Joy_1                : num  0.00169 0.00122 0.00253 0.00181 0.00165 ...
## $ Surprise_1           : num  0.505 1.816 0.672 0.202 0.335 ...
## $ Fear_1               : num  0.0284 32.02929 0.00402 0.00444 0.00406 ...
## $ Contempt_1           : num  0.198 0.192 3.536 0.193 0.193 ...
## $ PeakCount_1          : int  2 4 13 9 5 9 10 17 0 17 ...
## $ Peak.Min_1           : num  1.59 2.7 8.9 6.19 3.41 ...
```

```
##  $ AveAmplitude_1       : num  0.1492 0.0664 0.14 0.1117 0.0463 ...
##  $ MaxAmplitude_1       : num  0.192 0.1935 0.432 0.36 0.0825 ...
##  $ GSR_CAL_1            : num  2.64 2.04 6.05 2.9 1.56 ...
```

```
train <- sample (1:nrow(RatingP1A1Mdata3), size=0.8*nrow(RatingP1A1Mdata3)) # tra
ining row indices
RatingP1A1_train <- RatingP1A1Mdata3[train, ] # training data
RatingP1A1_test <- RatingP1A1Mdata3[-train, ] # test data
hist.default(RatingP1A1_train$Rating_1, main = "Histogram of Rating Scores in Train D
ata For P1",
          xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP1A1_test$Rating_1, main = "Histogram of Rating Scores in Test Dat
a For P1",
          xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree
formula=Rating_1 ~ PupilLeft_1 + PupilRight_1 + minFixationStart_1 + sumFixationDu
ration_1 + Classification_1 + HighEngagement_1 + LowEngagement_1 + Distraction_1.
1 + Drowsy_1 + WorkloadFBDS_1 + WorkloadBDS_1 + WorkloadAverage_1 + BrowF
urrow_1 + BrowRaise_1 + LipCornerDepressor_1 + Smile_1 + Valence_1 + Attention_1
+ Anger_1 + Sadness_1 + Disgust_1 + Joy_1 + Surprise_1 + Fear_1 + Contempt_1 + Pea
kCount_1 + Peak.Min_1 + AveAmplitude_1 + MaxAmplitude_1 + GSR_CAL_1


RatingP1A1_regTree=rpart(formula,data=RatingP1A1_train,method="anova",control=rp
art.control(minsplit=5,cp=0.001))
#plot(RegTree)
plot(RatingP1A1_regTree, uniform=TRUE,
    main="Regression Tree For P1 Rating")
text(RatingP1A1_regTree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number of
observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP1A1_regTree)
##
## Regression tree:
## rpart(formula = formula, data = RatingP1A1_train, method = "anova",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Attention_1          LipCornerDepressor_1 minFixationStart_1
## [4] PupilLeft_1          PupilRight_1         Surprise_1
## [7] WorkloadAverage_1    WorkloadFBDS_1
##
## Root node error: 264.22/36 = 7.3395
##
## n= 36
##
##          CP nsplit rel error xerror    xstd
## 1  0.323297      0  1.000000 1.0556 0.22608
## 2  0.142756      1  0.676703 1.2658 0.28726
## 3  0.123633      3  0.391192 1.7114 0.41856
## 4  0.058587      4  0.267558 1.9930 0.45250
## 5  0.056621      5  0.208972 1.8359 0.43998
## 6  0.040991      6  0.152351 1.9232 0.44576
## 7  0.014598      7  0.111360 2.1072 0.45060
## 8  0.010219      8  0.096762 2.0523 0.45832
## 9  0.010219      9  0.086543 2.0464 0.45897
## 10 0.001000     10  0.076325 2.0322 0.45374
plotcp(RatingP1A1_regTree)
```

```
#summary(RatingP1A1_regTree)
RatingP1A1_regTree
## n= 36
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 36 264.222200 3.2222220
##     2) PupilLeft_1< 2.787365 30 135.466700 2.5333330
##       4) Surprise_1>=2.180666 7   6.857143 0.8571429
##         8) WorkloadAverage_1< 0.5350445 3   0.000000 0.0000000 *
##         9) WorkloadAverage_1>=0.5350445 4   3.000000 1.5000000 *
##       5) Surprise_1< 2.180666 23 102.956500 3.0434780
##        10) Surprise_1< 0.9270395 19  50.421050 2.3684210
##          20) LipCornerDepressor_1< 0.7469331 17  34.941180 2.0588240
##            40) minFixationStart_1>=72.5 13  17.230770 1.5384620
##              80) Attention_1< 91.82903 5   3.200000 0.6000000
##               160) PupilRight_1>=2.418548 3   0.000000 0.0000000 *
##               161) PupilRight_1< 2.418548 2   0.500000 1.5000000 *
##              81) Attention_1>=91.82903 5   3.200000 2.6000000
##               162) PupilLeft_1< 2.431911 3   0.000000 2.0000000 *
##               163) PupilLeft_1>=2.431911 2   0.500000 3.5000000 *
##            41) minFixationStart_1< 72.5 4   2.750000 3.7500000 *
##          21) LipCornerDepressor_1>=0.7469331 2   0.000000 5.0000000 *
##        11) Surprise_1>=0.9270395 4   2.750000 6.2500000 *
##     3) PupilLeft_1>=2.787365 6  43.333330 6.6666670
##       6) WorkloadFBDS_1< 0.584792 3   8.666667 4.3333330 *
##       7) WorkloadFBDS_1>=0.584792 3   2.000000 9.0000000 *
#Model responses in train data
```

```
ModelResponse_train = predict(RatingP1A1_regTree, RatingP1A1_train)

df_train = data.frame("actual" =RatingP1A1_train$Rating_1 , "predicted" = ModelRespo
nse_train)

plot(df_train$actual, df_train$predicted, main = "Actual vs Predicted Rating Scores For P
1",
    xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Train Data")

lines(lowess(df_train$actual, df_train$predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot
#print.data.frame(df_train)


# Calculate and print error in Train data
error_train <- df_train$actual - df_train$predicted
#print(error_train)
print(RatingP1A1Mdelta<-as.data.frame(cbind(df_train,error_train)))
```

```
##    actual predicted error_train
## 9       3  3.750000  -0.7500000
## 28      0  0.000000   0.0000000
## 43      2  2.000000   0.0000000
## 2       7  6.250000   0.7500000
## 21      9  9.000000   0.0000000
## 16      1  1.500000  -0.5000000
## 22      3  1.500000   1.5000000
## 4       4  3.500000   0.5000000
## 12      0  0.000000   0.0000000
## 30      0  0.000000   0.0000000
## 39      5  3.750000   1.2500000
## 19      2  2.000000   0.0000000
## 17      6  6.250000  -0.2500000
## 45      2  1.538462   0.4615385
```

```
## 25      1  1.500000  -0.5000000
## 33      5  6.250000  -1.2500000
## 10      2  1.500000   0.5000000
## 27      1  1.500000  -0.5000000
## 31      0  0.000000   0.0000000
## 13      7  6.250000   0.7500000
## 29      1  1.500000  -0.5000000
## 14      0  0.000000   0.0000000
## 1      10  9.000000   1.0000000
## 35      5  4.333333   0.6666667
## 15      1  1.538462  -0.5384615
## 3       2  4.333333  -2.3333333
## 5       5  5.000000   0.0000000
## 34      5  5.000000   0.0000000
## 26      1  1.538462  -0.5384615
## 44      3  3.500000  -0.5000000
## 11      0  0.000000   0.0000000
## 7       8  9.000000  -1.0000000
## 20      6  4.333333   1.6666667
## 32      4  3.750000   0.2500000
## 18      3  3.750000  -0.7500000
## 37      2  2.000000   0.0000000
```

```
write.csv(RatingP1A1Mdelta,file="Rating P1A1 RegTree Train Delta.csv",row.names=T
RUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P1/A1/Model"
```

```
# Function that returns Root Mean Squared Error
rmse <- function(error_train)
{
  sqrt(mean(error_train^2))
```

```
}
# Function that returns Mean Absolute Error
mae <- function(error_train)
{
  mean(abs(error_train))
}
#mae(actual, predicted)
# Print rmse and mae in train data
rmse(error_train)
```

```
## [1] 0.763027
```

```
mae(error_train)
```

```
## [1] 0.5334758
```

```
# Model Testing
ModelResponse_test = predict(RatingP1A1_regTree, RatingP1A1_test)
df_test = data.frame("actualinTest" =RatingP1A1_test$Rating_1 , "predictedinTest" = M
odelResponse_test)
plot(df_test$actualinTest, df_test$predictedinTest, main = "Actual vs Predicted Rating Sc
ores For P1",
     xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test Data")
par(pty='s') #produces a square plot
lines(lowess(df_test$actualinTest, df_test$predictedinTest), col = "blue") # Add loess fit
```

```
#print.data.frame(df_test)


# Calculate and print error in test data
error_test <- df_test$actual - df_test$predicted
#print(error_train)
(RatingP1A1Mdelta<-as.data.frame(cbind(df_test,error_test)))
```

```
##    actualinTest predictedinTest error_test
## 6            1        3.500000 -2.5000000
```

```
## 8            3        4.333333 -1.3333333
## 23           4        1.500000  2.5000000
## 24           5        9.000000 -4.0000000
## 36           0        4.333333 -4.3333333
## 38           1        1.500000 -0.5000000
## 40           2        3.750000 -1.7500000
## 41           8        0.000000  8.0000000
## 42           3        6.250000 -3.2500000
## 48           5        4.333333  0.6666667
```

```
write.csv(RatingP1A1Mdelta,file="Rating P1A1 RegTree Test Delta.csv",row.names=T
RUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P1/A1/Model"
```

```
# Function that returns Root Mean Squared Error
rmse <- function(error_test)
{
  sqrt(mean(error_test^2))
}
# Function that returns Mean Absolute Error
mae <- function(error_test)
{
  mean(abs(error_test))
}
#mae(actual, predicted)
# Print rmse and mae in test data
rmse(error_test)
```

```
## [1] 3.568963
```

```
mae(error_test)
```

```
## [1] 2.883333
```

APPENDIX E

PREDICTION OF RATING CLASS FOR PRODUCT 1

RatingP2H1NPS3.R
irfan
Tue Mar 6 14:54:46 2018

```
#UoB Rating Analysis
#RatingP2H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS")

# Upload the data
RatingP2H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP2H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP2H1NPSdata2 <- RatingP2H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,61,63,
65:74,182:211)]
names(RatingP2H1NPSdata2)
## [1] "Subject"             "InternetPerDay"
## [3] "OnlineShopPmont"     "PhotosVideosA"
## [5] "DescriptionsA"       "RatingsCommentsA"
## [7] "CostA"               "SportsPerWeek"
## [9] "SportsPerDay"        "CoachDrivenS"
## [11] "EBraSize"           "SportsBraSize"
## [13] "SportsBraUse"       "SportsbrasN"
## [15] "MoneyWishes"        "MoneyPractice"
## [17] "ProductID_2"        "NPSID_2"
## [19] "ProductVideos_2"    "RatingsComments_2"
## [21] "Descriptions_2"     "Cost_2"
## [23] "Confusion_2"        "Distraction_2"
## [25] "Enjoyment_2"        "Interest_2"
## [27] "Tiredness_2"        "Frustration_2"
## [29] "PupilLeft_2"        "PupilRight_2"
## [31] "minFixationStart_2" "sumFixationDuration_2"
## [33] "Classification_2"   "HighEngagement_2"
## [35] "LowEngagement_2"    "Distraction_2.1"
## [37] "Drowsy_2"           "WorkloadFBDS_2"
## [39] "WorkloadBDS_2"      "WorkloadAverage_2"
## [41] "BrowFurrow_2"       "BrowRaise_2"
## [43] "LipCornerDepressor_2" "Smile_2"
## [45] "Valence_2"          "Attention_2"
## [47] "Anger_2"            "Sadness_2"
```

```
## [49] "Disgust_2"            "Joy_2"
## [51] "Surprise_2"           "Fear_2"
## [53] "Contempt_2"           "PeakCount_2"
## [55] "Peak.Min_2"           "AveAmplitude_2"
## [57] "MaxAmplitude_2"       "GSR_CAL_2"
```

```
#If NPSID_2 have NAs, omit these rows that contain NA values
#RatingP2H1NPSdata3 <- RatingP2H1NPSdata2[!is.na(RatingP2H1NPSdata2$PeakCo
unt_2),]
RatingP2H1NPSdata3 <- RatingP2H1NPSdata2[!is.na(RatingP2H1NPSdata2$NPSID_2)
,]
#str(RatingP2H1NPSdata3)


#Descriptives
describe(RatingP2H1NPSdata3$NPSID_2)
```

```
## RatingP2H1NPSdata3$NPSID_2
##        n  missing distinct     Info     Mean      Gmd
##       48        0        3     0.42     1.25   0.4397
##
## Value          1     2     3
## Frequency     40     4     4
## Proportion 0.833 0.083 0.083
```

```
plot.default(RatingP2H1NPSdata3$NPSID_2, main = "Rating Scores For P2",
             xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP2H1NPSdata3$NPSID_2, main = "Histogram of Rating Scores For P
2",
             xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP2H1NPSdata3$NPSID_2) # returns the density data
```

```
plot(d) # plots the results
```

```
#Create training and test data
str(RatingP2H1NPSdata3)
```

```
## 'data.frame':    48 obs. of  58 variables:
##  $ Subject            : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay      : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont      : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA       : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA       : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA     : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA               : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek       : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay        : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS        : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize            : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize       : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse        : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN         : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes         : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice       : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ ProductID_2         : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ NPSID_2             : int  3 1 2 1 1 1 1 1 1 2 ...
##  $ ProductVideos_2      : int  7 6 7 6 3 2 7 5 5 6 ...
##  $ RatingsComments_2    : int  6 1 1 7 1 1 7 1 4 1 ...
##  $ Descriptions_2       : int  7 3 6 6 2 3 6 3 7 2 ...
##  $ Cost_2              : int  7 3 6 5 6 3 6 4 6 5 ...
##  $ Confusion_2         : int  NA 5 0 8 0 1 3 6 5 3 ...
##  $ Distraction_2       : int  NA 5 0 5 1 2 6 9 2 7 ...
```

```
##  $ Enjoyment_2          : int  NA 5 8 6 4 3 5 1 3 4 ...
##  $ Interest_2           : int  NA 3 8 4 4 3 2 2 2 3 ...
##  $ Tiredness_2          : int  NA 3 5 3 0 4 3 5 0 2 ...
##  $ Frustration_2        : int  NA 0 0 5 0 4 6 0 2 2 ...
##  $ PupilLeft_2          : num  2.82 2.46 3.38 2.55 2.61 ...
##  $ PupilRight_2         : num  2.77 2.43 3.47 2.57 2.65 ...
##  $ minFixationStart_2   : int  230 0 602 180 440 61 0 62 0 967 ...
##  $ sumFixationDuration_2: int  8968155 3934942 12616682 13368035 8177879 7519
561 4164101 7952600 9011590 7117524 ...
##  $ Classification_2     : num  0.664 0.685 0.82 0.761 0.737 ...
##  $ HighEngagement_2     : num  0.433 0.369 0.713 0.584 0.487 ...
##  $ LowEngagement_2      : num  0.325 0.581 0.282 0.386 0.453 ...
##  $ Distraction_2.1      : num  2.37e-01 4.92e-02 1.00e-08 3.00e-02 5.84e-02 ...
##  $ Drowsy_2             : num  5.51e-03 1.37e-05 5.32e-03 0.00 1.32e-03 ...
##  $ WorkloadFBDS_2       : num  0.752 0.439 0.614 0.446 0.605 ...
##  $ WorkloadBDS_2        : num  0.642 0.385 0.56 0.426 0.532 ...
##  $ WorkloadAverage_2    : num  0.697 0.412 0.587 0.436 0.569 ...
##  $ BrowFurrow_2         : num  3.66e-07 1.76e-03 3.11e-05 2.61e-05 9.05e-05 ...
##  $ BrowRaise_2          : num  3.904 6.83 0.537 0.44 4.854 ...
##  $ LipCornerDepressor_2 : num  0.11183 0.00599 0.13015 0.70968 1.09123 ...
##  $ Smile_2              : num  6.91e-03 2.51e-05 3.05e-05 3.26e-10 1.64e-04 ...
##  $ Valence_2            : num  0 0 -2.645 -0.316 -2.316 ...
##  $ Attention_2          : num  96.7 98.5 91.9 95.5 90.1 ...
##  $ Anger_2              : num  0.00283 0.03413 0.11906 0.00243 0.00207 ...
##  $ Sadness_2            : num  0.01643 0.00103 0.02064 0.02443 0.02267 ...
##  $ Disgust_2            : num  0.549 0.427 0.449 0.431 0.518 ...
##  $ Joy_2                : num  0.00207 0.00154 0.00192 0.00182 0.00163 ...
##  $ Surprise_2           : num  0.417 0.976 0.231 0.215 0.329 ...
##  $ Fear_2               : num  0.00515 30.53138 0.00443 0.00415 0.01252 ...
##  $ Contempt_2           : num  0.194 0.193 0.193 0.198 6.39 ...
```

```
##  $ PeakCount_2         : int  2 1 14 4 2 9 0 11 2 26 ...
##  $ Peak.Min_2          : num  1.35 0.68 9.43 2.76 1.36 ...
##  $ AveAmplitude_2      : num  0.0187 0.0105 0.0571 0.1144 0.0377 ...
##  $ MaxAmplitude_2      : num  0.024 0.0105 0.1944 0.336 0.0645 ...
##  $ GSR_CAL_2           : num  2.76 1.95 6.2 2.74 1.46 ...
```

```
train <- sample (1:nrow(RatingP2H1NPSdata3), size=0.8*nrow(RatingP2H1NPSdata3)) 
# training row indices

RatingP2H1NPS_train <- RatingP2H1NPSdata3[train, ] # training data

RatingP2H1NPS_test <- RatingP2H1NPSdata3[-train, ] # test data


hist.default(RatingP2H1NPS_train$NPSID_2, main = "Histogram of Rating Scores in Tr
ain Data For P2",

            xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP2H1NPS_test$NPSID_2, main = "Histogram of Rating Scores in Test
Data For P2",

            xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree

formula=NPSID_2 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA
+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize
+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVi
deos_2+RatingsComments_2+Descriptions_2+Cost_2+Confusion_2+Distraction_2+Enjo
yment_2+Interest_2+Tiredness_2+Frustration_2 + PupilLeft_2 + PupilRight_2 + minFix
ationStart_2 + sumFixationDuration_2 + Classification_2 + HighEngagement_2 + LowE
ngagement_2 + Distraction_2.1 + Drowsy_2 + WorkloadFBDS_2 + WorkloadBDS_2 +
WorkloadAverage_2 + BrowFurrow_2 + BrowRaise_2 + LipCornerDepressor_2 + Smile
_2 + Valence_2 + Attention_2 + Anger_2 + Sadness_2 + Disgust_2 + Joy_2 + Surprise_2
+ Fear_2 + Contempt_2 + PeakCount_2 + Peak.Min_2 + AveAmplitude_2 + MaxAmplit
ude_2 + GSR_CAL_2


RatingP2H1NPSdtree=rpart(formula,data=RatingP2H1NPS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model
```

```
#plot(dtree)
plot(RatingP2H1NPSdtree, uniform=TRUE,
    main="Classification Tree For P2 Rating")
text(RatingP2H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP2H1NPSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RatingP2H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Classification_2 Enjoyment_2      WorkloadBDS_2
##
## Root node error: 6/38 = 0.15789
##
## n= 38
##
##        CP nsplit rel error  xerror    xstd
## 1 0.50000      0   1.00000 1.00000 0.37463
## 2 0.16667      1   0.50000 0.83333 0.34730
## 3 0.00100      3   0.16667 0.83333 0.34730
```

```
plotcp(RatingP2H1NPSdtree)
```

```
#summary(RatingP2H1NPSdtree)
RatingP2H1NPSdtree
```

```
## n= 38
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 38 6 1 (0.84210526 0.10526316 0.05263158)
##    2) Enjoyment_2< 7.5 35 3 1 (0.91428571 0.02857143 0.05714286)
##      4) WorkloadBDS_2< 0.5960118 30 1 1 (0.96666667 0.03333333 0.00000000) *
##      5) WorkloadBDS_2>=0.5960118 5 2 1 (0.60000000 0.00000000 0.40000000)
##       10) Classification_2>=0.6866548 3 0 1 (1.00000000 0.00000000 0.00000000) *
##       11) Classification_2< 0.6866548 2 0 3 (0.00000000 0.00000000 1.00000000) *
##    3) Enjoyment_2>=7.5 3 0 2 (0.00000000 1.00000000 0.00000000) *
```

```
RatingP2H1NPS_train$NPSID_2
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1
## [36] 1 1 1
```

```
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP2H1NPSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1
## [36] 1 1 1
```

```
length(train_response_predicted)
```

```
## [1] 38
```

```
train_response_actual<- RatingP2H1NPS_train$NPSID_2 # actuals
train_response_actual
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1
## [36] 1 1 1
```

```
length(train_response_actual)
```

```
## [1] 38
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9736842
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.02631579
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating Scores For P2 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP2H1NPS_train$NPSID_2, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      1  2  3
##   1 32  0  0
##   2  1  3  0
##   3  0  0  2
#create a Actual vs Predicted Table from Decision Tree
(RatingP2H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predic
ted)))
##    train_response_actual train_response_predicted
## 1                      3                        3
## 2                      1                        1
## 3                      1                        1
## 4                      1                        1
## 5                      1                        1
```

```
## 6           1         1
## 7           1         1
## 8           2         2
## 9           1         1
## 10          1         1
## 11          1         1
## 12          1         1
## 13          1         1
## 14          1         1
## 15          1         1
## 16          1         1
## 17          1         1
## 18          1         1
## 19          3         3
## 20          1         1
## 21          1         1
## 22          1         1
## 23          1         1
## 24          1         1
## 25          2         2
## 26          1         1
## 27          1         1
## 28          1         1
## 29          1         1
## 30          2         2
## 31          1         1
## 32          1         1
## 33          1         1
## 34          2         1
## 35          1         1
```

```
## 36                 1                  1
## 37                 1                  1
## 38                 1                  1
```

```
names(RatingP2H1NPSdelta)<-c("RatingP2H1NPSactual","RatingP2H1NPSpredicted")
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted
## 1                  3                      3
## 2                  1                      1
## 3                  1                      1
## 4                  1                      1
## 5                  1                      1
## 6                  1                      1
```

```
RatingP2H1NPSdelta$Match<-ifelse(RatingP2H1NPSdelta$RatingP2H1NPSactual==Ra
tingP2H1NPSdelta$RatingP2H1NPSpredicted,1,0)
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 1                  3                      3     1
## 2                  1                      1     1
## 3                  1                      1     1
## 4                  1                      1     1
## 5                  1                      1     1
## 6                  1                      1     1
```

```
tail(RatingP2H1NPSdelta)
```

```
##    RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 33                   1                      1     1
## 34                   2                      1     0
## 35                   1                      1     1
## 36                   1                      1     1
## 37                   1                      1     1
## 38                   1                      1     1
```

```
write.csv(RatingP2H1NPSdelta,file="RatingP2H1NPSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS"
#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9736842
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.02631579
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP2H1NPSdtree, RatingP2H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
##  [1] 1 1 1 1 2 1 1 1 3 1
length(test_response_predicted)
## [1] 10
test_response_actual<- RatingP2H1NPS_test$NPSID_2 # actuals
test_response_actual
##  [1] 1 3 1 1 3 1 1 1 1 1
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.7
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.3
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating S
cores For P2 3 Grouped",
    xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
```

```
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP2H1NPS_test$NPSID_2, test_response_predicted)
print(confusion.matrix)
```

```
##    test_response_predicted
##     1 2 3
##   1 7 0 1
##   3 1 1 0
```

```
#create a Actual vs Predicted Table from Decision Tree
(RatingP2H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicte
d)))
```

```
##    test_response_actual test_response_predicted
## 1                     1                       1
## 2                     3                       1
## 3                     1                       1
## 4                     1                       1
## 5                     3                       2
## 6                     1                       1
## 7                     1                       1
## 8                     1                       1
## 9                     1                       3
## 10                    1                       1
```

```
names(RatingP2H1NPSdelta)<-c("RatingP2H1NPSactual","RatingP2H1NPSpredicted")
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted
## 1                   1                      1
```

```
## 2              3              1
## 3              1              1
## 4              1              1
## 5              3              2
## 6              1              1
```

```
RatingP2H1NPSdelta$Match<-ifelse(RatingP2H1NPSdelta$RatingP2H1NPSactual==RatingP2H1NPSdelta$RatingP2H1NPSpredicted,1,0)
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 1              1              1     1
## 2              3              1     0
## 3              1              1     1
## 4              1              1     1
## 5              3              2     0
## 6              1              1     1
```

```
tail(RatingP2H1NPSdelta)
```

```
##    RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 5               3              2     0
## 6               1              1     1
## 7               1              1     1
## 8               1              1     1
## 9               1              3     0
## 10              1              1     1
```

```
write.csv(RatingP2H1NPSdelta,file="RatingP2H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
```

```
## [1] 0.7
```

```
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.3
```

APPENDIX F

PREDICTION OF RATING CLASS FOR PRODUCT 2

RatingP2H1NPS3.R
irfan
Tue Mar 6 14:54:46 2018

```
#UoB Rating Analysis
#RatingP2H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS")

# Upload the data
RatingP2H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP2H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP2H1NPSdata2 <- RatingP2H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,61,63,
65:74,182:211)]
names(RatingP2H1NPSdata2)
```

```
##  [1] "Subject"             "InternetPerDay"
##  [3] "OnlineShopPmont"     "PhotosVideosA"
##  [5] "DescriptionsA"       "RatingsCommentsA"
##  [7] "CostA"               "SportsPerWeek"
##  [9] "SportsPerDay"        "CoachDrivenS"
## [11] "EBraSize"            "SportsBraSize"
## [13] "SportsBraUse"        "SportsbrasN"
## [15] "MoneyWishes"         "MoneyPractice"
## [17] "ProductID_2"         "NPSID_2"
## [19] "ProductVideos_2"     "RatingsComments_2"
## [21] "Descriptions_2"      "Cost_2"
## [23] "Confusion_2"         "Distraction_2"
## [25] "Enjoyment_2"         "Interest_2"
## [27] "Tiredness_2"         "Frustration_2"
## [29] "PupilLeft_2"         "PupilRight_2"
## [31] "minFixationStart_2"  "sumFixationDuration_2"
## [33] "Classification_2"    "HighEngagement_2"
## [35] "LowEngagement_2"     "Distraction_2.1"
## [37] "Drowsy_2"            "WorkloadFBDS_2"
## [39] "WorkloadBDS_2"       "WorkloadAverage_2"
## [41] "BrowFurrow_2"        "BrowRaise_2"
## [43] "LipCornerDepressor_2" "Smile_2"
## [45] "Valence_2"           "Attention_2"
## [47] "Anger_2"             "Sadness_2"
```

```
## [49] "Disgust_2"            "Joy_2"
## [51] "Surprise_2"           "Fear_2"
## [53] "Contempt_2"           "PeakCount_2"
## [55] "Peak.Min_2"           "AveAmplitude_2"
## [57] "MaxAmplitude_2"       "GSR_CAL_2"
```

```
#If NPSID_2 have NAs, omit these rows that contain NA values
#RatingP2H1NPSdata3 <- RatingP2H1NPSdata2[!is.na(RatingP2H1NPSdata2$PeakCo
unt_2),]
RatingP2H1NPSdata3 <- RatingP2H1NPSdata2[!is.na(RatingP2H1NPSdata2$NPSID_2)
,]
#str(RatingP2H1NPSdata3)


#Descriptives
describe(RatingP2H1NPSdata3$NPSID_2)
```

```
## RatingP2H1NPSdata3$NPSID_2
##        n  missing distinct     Info     Mean      Gmd
##       48        0        3     0.42     1.25   0.4397
##
## Value          1     2     3
## Frequency     40     4     4
## Proportion 0.833 0.083 0.083
```

```
plot.default(RatingP2H1NPSdata3$NPSID_2, main = "Rating Scores For P2",
             xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP2H1NPSdata3$NPSID_2, main = "Histogram of Rating Scores For P
2",
             xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP2H1NPSdata3$NPSID_2) # returns the density data
```

```r
plot(d) # plots the results
```

```r
#Create training and test data
str(RatingP2H1NPSdata3)
## 'data.frame':    48 obs. of  58 variables:
##  $ Subject            : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay     : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont    : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA      : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA      : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA   : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA              : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek      : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay       : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS       : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize           : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize      : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse       : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN        : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes        : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice      : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ ProductID_2        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ NPSID_2            : int  3 1 2 1 1 1 1 1 1 2 ...
##  $ ProductVideos_2    : int  7 6 7 6 3 2 7 5 5 6 ...
##  $ RatingsComments_2  : int  6 1 1 7 1 1 7 1 4 1 ...
##  $ Descriptions_2     : int  7 3 6 6 2 3 6 3 7 2 ...
##  $ Cost_2             : int  7 3 6 5 6 3 6 4 6 5 ...
##  $ Confusion_2        : int  NA 5 0 8 0 1 3 6 5 3 ...
##  $ Distraction_2      : int  NA 5 0 5 1 2 6 9 2 7 ...
```

```
##  $ Enjoyment_2         : int  NA 5 8 6 4 3 5 1 3 4 ...
##  $ Interest_2          : int  NA 3 8 4 4 3 2 2 2 3 ...
##  $ Tiredness_2         : int  NA 3 5 3 0 4 3 5 0 2 ...
##  $ Frustration_2       : int  NA 0 0 5 0 4 6 0 2 2 ...
##  $ PupilLeft_2         : num  2.82 2.46 3.38 2.55 2.61 ...
##  $ PupilRight_2        : num  2.77 2.43 3.47 2.57 2.65 ...
##  $ minFixationStart_2  : int  230 0 602 180 440 61 0 62 0 967 ...
##  $ sumFixationDuration_2: int  8968155 3934942 12616682 13368035 8177879 7519
561 4164101 7952600 9011590 7117524 ...
##  $ Classification_2    : num  0.664 0.685 0.82 0.761 0.737 ...
##  $ HighEngagement_2    : num  0.433 0.369 0.713 0.584 0.487 ...
##  $ LowEngagement_2     : num  0.325 0.581 0.282 0.386 0.453 ...
##  $ Distraction_2.1     : num  2.37e-01 4.92e-02 1.00e-08 3.00e-02 5.84e-02 ...
##  $ Drowsy_2            : num  5.51e-03 1.37e-05 5.32e-03 0.00 1.32e-03 ...
##  $ WorkloadFBDS_2      : num  0.752 0.439 0.614 0.446 0.605 ...
##  $ WorkloadBDS_2       : num  0.642 0.385 0.56 0.426 0.532 ...
##  $ WorkloadAverage_2   : num  0.697 0.412 0.587 0.436 0.569 ...
##  $ BrowFurrow_2        : num  3.66e-07 1.76e-03 3.11e-05 2.61e-05 9.05e-05 ...
##  $ BrowRaise_2         : num  3.904 6.83 0.537 0.44 4.854 ...
##  $ LipCornerDepressor_2 : num  0.11183 0.00599 0.13015 0.70968 1.09123 ...
##  $ Smile_2             : num  6.91e-03 2.51e-05 3.05e-05 3.26e-10 1.64e-04 ...
##  $ Valence_2           : num  0 0 -2.645 -0.316 -2.316 ...
##  $ Attention_2         : num  96.7 98.5 91.9 95.5 90.1 ...
##  $ Anger_2             : num  0.00283 0.03413 0.11906 0.00243 0.00207 ...
##  $ Sadness_2           : num  0.01643 0.00103 0.02064 0.02443 0.02267 ...
##  $ Disgust_2           : num  0.549 0.427 0.449 0.431 0.518 ...
##  $ Joy_2               : num  0.00207 0.00154 0.00192 0.00182 0.00163 ...
##  $ Surprise_2          : num  0.417 0.976 0.231 0.215 0.329 ...
##  $ Fear_2              : num  0.00515 30.53138 0.00443 0.00415 0.01252 ...
##  $ Contempt_2          : num  0.194 0.193 0.193 0.198 6.39 ...
```

```
## $ PeakCount_2          : int  2 1 14 4 2 9 0 11 2 26 ...
## $ Peak.Min_2           : num  1.35 0.68 9.43 2.76 1.36 ...
## $ AveAmplitude_2       : num  0.0187 0.0105 0.0571 0.1144 0.0377 ...
## $ MaxAmplitude_2       : num  0.024 0.0105 0.1944 0.336 0.0645 ...
## $ GSR_CAL_2            : num  2.76 1.95 6.2 2.74 1.46 ...
```

```
train <- sample (1:nrow(RatingP2H1NPSdata3), size=0.8*nrow(RatingP2H1NPSdata3))
# training row indices

RatingP2H1NPS_train <- RatingP2H1NPSdata3[train, ] # training data

RatingP2H1NPS_test <- RatingP2H1NPSdata3[-train, ] # test data


hist.default(RatingP2H1NPS_train$NPSID_2, main = "Histogram of Rating Scores in Tr
ain Data For P2",

           xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP2H1NPS_test$NPSID_2, main = "Histogram of Rating Scores in Test
Data For P2",

           xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree

formula=NPSID_2 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA
+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize
+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVi
deos_2+RatingsComments_2+Descriptions_2+Cost_2+Confusion_2+Distraction_2+Enjo
yment_2+Interest_2+Tiredness_2+Frustration_2 + PupilLeft_2 + PupilRight_2 + minFix
ationStart_2 + sumFixationDuration_2 + Classification_2 + HighEngagement_2 + LowE
ngagement_2 + Distraction_2.1 + Drowsy_2 + WorkloadFBDS_2 + WorkloadBDS_2 +
WorkloadAverage_2 + BrowFurrow_2 + BrowRaise_2 + LipCornerDepressor_2 + Smile
_2 + Valence_2 + Attention_2 + Anger_2 + Sadness_2 + Disgust_2 + Joy_2 + Surprise_2
+ Fear_2 + Contempt_2 + PeakCount_2 + Peak.Min_2 + AveAmplitude_2 + MaxAmplit
ude_2 + GSR_CAL_2


RatingP2H1NPSdtree=rpart(formula,data=RatingP2H1NPS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model
```

```
#plot(dtree)
plot(RatingP2H1NPSdtree, uniform=TRUE,
    main="Classification Tree For P2 Rating")
text(RatingP2H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP2H1NPSdtree)
##
## Classification tree:
## rpart(formula = formula, data = RatingP2H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Classification_2 Enjoyment_2      WorkloadBDS_2
##
## Root node error: 6/38 = 0.15789
##
## n= 38
##
##        CP nsplit rel error  xerror    xstd
## 1 0.50000      0   1.00000 1.00000 0.37463
## 2 0.16667      1   0.50000 0.83333 0.34730
## 3 0.00100      3   0.16667 0.83333 0.34730
plotcp(RatingP2H1NPSdtree)
```

```
#summary(RatingP2H1NPSdtree)
RatingP2H1NPSdtree
```

```
## n= 38
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 38 6 1 (0.84210526 0.10526316 0.05263158)
##    2) Enjoyment_2< 7.5 35 3 1 (0.91428571 0.02857143 0.05714286)
##      4) WorkloadBDS_2< 0.5960118 30 1 1 (0.96666667 0.03333333 0.00000000) *
##      5) WorkloadBDS_2>=0.5960118 5 2 1 (0.60000000 0.00000000 0.40000000)
##       10) Classification_2>=0.6866548 3 0 1 (1.00000000 0.00000000 0.00000000) *
##       11) Classification_2< 0.6866548 2 0 3 (0.00000000 0.00000000 1.00000000) *
##    3) Enjoyment_2>=7.5 3 0 2 (0.00000000 1.00000000 0.00000000) *
```

```
RatingP2H1NPS_train$NPSID_2
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1
## [36] 1 1 1
```

```
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP2H1NPSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1
## [36] 1 1 1
```

```
length(train_response_predicted)
```

```
## [1] 38
```

```
train_response_actual<- RatingP2H1NPS_train$NPSID_2 # actuals
train_response_actual
```

```
##  [1] 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1
## [36] 1 1 1
```

```
length(train_response_actual)
```

```
## [1] 38
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9736842
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.02631579
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating Scores For P2 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP2H1NPS_train$NPSID_2, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      1  2  3
##   1 32  0  0
##   2  1  3  0
##   3  0  0  2
#create a Actual vs Predicted Table from Decision Tree
(RatingP2H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predic
ted)))
##    train_response_actual train_response_predicted
## 1                      3                        3
## 2                      1                        1
## 3                      1                        1
## 4                      1                        1
## 5                      1                        1
```

```
## 6          1          1
## 7          1          1
## 8          2          2
## 9          1          1
## 10         1          1
## 11         1          1
## 12         1          1
## 13         1          1
## 14         1          1
## 15         1          1
## 16         1          1
## 17         1          1
## 18         1          1
## 19         3          3
## 20         1          1
## 21         1          1
## 22         1          1
## 23         1          1
## 24         1          1
## 25         2          2
## 26         1          1
## 27         1          1
## 28         1          1
## 29         1          1
## 30         2          2
## 31         1          1
## 32         1          1
## 33         1          1
## 34         2          1
## 35         1          1
```

```
## 36                  1                  1
## 37                  1                  1
## 38                  1                  1
```

```
names(RatingP2H1NPSdelta)<-c("RatingP2H1NPSactual","RatingP2H1NPSpredicted")
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted
## 1                3                  3
## 2                1                  1
## 3                1                  1
## 4                1                  1
## 5                1                  1
## 6                1                  1
```

```
RatingP2H1NPSdelta$Match<-ifelse(RatingP2H1NPSdelta$RatingP2H1NPSactual==Ra
tingP2H1NPSdelta$RatingP2H1NPSpredicted,1,0)
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 1                3                  3    1
## 2                1                  1    1
## 3                1                  1    1
## 4                1                  1    1
## 5                1                  1    1
## 6                1                  1    1
```

```
tail(RatingP2H1NPSdelta)
```

```
##    RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 33                1                  1    1
## 34                2                  1    0
## 35                1                  1    1
## 36                1                  1    1
## 37                1                  1    1
## 38                1                  1    1
```

```
write.csv(RatingP2H1NPSdelta,file="RatingP2H1NPSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS"
#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9736842
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.02631579
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP2H1NPSdtree, RatingP2H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
##  [1] 1 1 1 1 2 1 1 1 3 1
length(test_response_predicted)
## [1] 10
test_response_actual<- RatingP2H1NPS_test$NPSID_2 # actuals
test_response_actual
##  [1] 1 3 1 1 3 1 1 1 1 1
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.7
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.3
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating S
cores For P2 3 Grouped",
     xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
```

```
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP2H1NPS_test$NPSID_2, test_response_predicted)
print(confusion.matrix)
##    test_response_predicted
##     1 2 3
##   1 7 0 1
##   3 1 1 0
#create a Actual vs Predicted Table from Decision Tree
(RatingP2H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicte
d)))
##    test_response_actual test_response_predicted
## 1                     1                       1
## 2                     3                       1
## 3                     1                       1
## 4                     1                       1
## 5                     3                       2
## 6                     1                       1
## 7                     1                       1
## 8                     1                       1
## 9                     1                       3
## 10                    1                       1
names(RatingP2H1NPSdelta)<-c("RatingP2H1NPSactual","RatingP2H1NPSpredicted")
head(RatingP2H1NPSdelta)
##   RatingP2H1NPSactual RatingP2H1NPSpredicted
## 1                   1                      1
```

```
## 2                3                1
## 3                1                1
## 4                1                1
## 5                3                2
## 6                1                1
```

```
RatingP2H1NPSdelta$Match<-ifelse(RatingP2H1NPSdelta$RatingP2H1NPSactual==RatingP2H1NPSdelta$RatingP2H1NPSpredicted,1,0)
head(RatingP2H1NPSdelta)
```

```
##   RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 1                1                1     1
## 2                3                1     0
## 3                1                1     1
## 4                1                1     1
## 5                3                2     0
## 6                1                1     1
```

```
tail(RatingP2H1NPSdelta)
```

```
##    RatingP2H1NPSactual RatingP2H1NPSpredicted Match
## 5                 3                2     0
## 6                 1                1     1
## 7                 1                1     1
## 8                 1                1     1
## 9                 1                3     0
## 10                1                1     1
```

```
write.csv(RatingP2H1NPSdelta,file="RatingP2H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P2/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
```

```
## [1] 0.7
```

```
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.3
```

APPENDIX G

PREDICTION OF RATING CLASS FOR PRODUCT 3

RatingP3H1NPS3.R
irfan
Tue Mar 6 14:20:59 2018

```
#UoB Rating Analysis
#RatingP3H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P3/H1NPS")

# Upload the data
RatingP3H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP3H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP3H1NPSdata2 <- RatingP3H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,80:91,
213:242)]
names(RatingP3H1NPSdata2)
```

```
##  [1] "Subject"              "InternetPerDay"
##  [3] "OnlineShopPmont"      "PhotosVideosA"
##  [5] "DescriptionsA"        "RatingsCommentsA"
##  [7] "CostA"                "SportsPerWeek"
##  [9] "SportsPerDay"         "CoachDrivenS"
## [11] "EBraSize"             "SportsBraSize"
## [13] "SportsBraUse"         "SportsbrasN"
## [15] "MoneyWishes"          "MoneyPractice"
## [17] "NPSID_3"              "RatingReason_3"
## [19] "ProductVideos_3"      "RatingsComments_3"
## [21] "Descriptions_3"       "Cost_3"
## [23] "Confusion_3"          "Distraction_3"
## [25] "Enjoyment_3"          "Interest_3"
## [27] "Tiredness_3"          "Frustration_3"
## [29] "PupilLeft_3"          "PupilRight_3"
## [31] "minFixationStart_3"   "sumFixationDuration_3"
## [33] "Classification_3"     "HighEngagement_3"
## [35] "LowEngagement_3"      "Distraction_3.1"
## [37] "Drowsy_3"             "WorkloadFBDS_3"
## [39] "WorkloadBDS_3"        "WorkloadAverage_3"
## [41] "BrowFurrow_3"         "BrowRaise_3"
## [43] "LipCornerDepressor_3" "Smile_3"
## [45] "Valence_3"            "Attention_3"
## [47] "Anger_3"              "Sadness_3"
```

```
## [49] "Disgust_3"            "Joy_3"
## [51] "Surprise_3"           "Fear_3"
## [53] "Contempt_3"           "PeakCount_3"
## [55] "Peak.Min_3"           "AveAmplitude_3"
## [57] "MaxAmplitude_3"       "GSR_CAL_3"
```

```
#If NPSID_3 have NAs, omit these rows that contain NA values
#RatingP3H1NPSdata3 <- RatingP3H1NPSdata2[!is.na(RatingP3H1NPSdata2$PeakCo
unt_3),]
RatingP3H1NPSdata3 <- RatingP3H1NPSdata2[!is.na(RatingP3H1NPSdata2$NPSID_3)
,]
#str(RatingP3H1NPSdata3)


#Descriptives
describe(RatingP3H1NPSdata3$NPSID_3)
```

```
## RatingP3H1NPSdata3$NPSID_3
##        n  missing distinct     Info     Mean      Gmd
##       48        0        3    0.808    1.604   0.7615
##
## Value          1     2     3
## Frequency     26    15     7
## Proportion 0.542 0.312 0.146
```

```
plot.default(RatingP3H1NPSdata3$NPSID_3, main = "Rating Scores For P3",
          xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP3H1NPSdata3$NPSID_3, main = "Histogram of Rating Scores For P
3",
          xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP3H1NPSdata3$NPSID_3) # returns the density data
```

```
plot(d) # plots the results
```

```
#Create training and test data
str(RatingP3H1NPSdata3)
```

```
## 'data.frame':    48 obs. of  58 variables:
##  $ Subject            : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay      : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont     : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA       : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA       : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA    : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA               : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek       : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay        : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS        : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize            : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize       : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse        : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN         : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes         : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice       : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ NPSID_3             : int  1 3 2 1 1 2 3 1 1 1 ...
##  $ RatingReason_3      : Factor w/ 47 levels "","Expensive",..: 12 24 34 19 31 6 22 27
13 32 ...
##  $ ProductVideos_3     : int  7 6 6 5 5 1 7 7 5 7 ...
##  $ RatingsComments_3   : int  7 7 7 5 6 6 7 7 6 7 ...
##  $ Descriptions_3      : int  7 6 6 3 7 6 7 7 5 7 ...
##  $ Cost_3              : int  7 6 6 7 7 1 6 4 6 3 ...
##  $ Confusion_3         : int  NA 6 0 8 1 2 2 6 0 7 ...
```

```
##  $ Distraction_3        : int  NA 2 0 6 1 2 2 0 3 2 ...
##  $ Enjoyment_3          : int  NA 7 6 8 6 4 6 1 2 4 ...
##  $ Interest_3           : int  NA 9 6 5 6 4 9 3 3 7 ...
##  $ Tiredness_3          : int  NA 2 4 4 0 5 0 3 0 2 ...
##  $ Frustration_3        : int  NA 2 0 6 1 3 1 1 0 6 ...
##  $ PupilLeft_3          : num  2.71 2.41 3.25 2.57 2.51 ...
##  $ PupilRight_3         : num  2.68 2.39 3.29 2.67 2.52 ...
##  $ minFixationStart_3   : int  2 1132 0 23 0 83 0 31 699 192 ...
##  $ sumFixationDuration_3: int  3579466 5590881 7654130 16365530 11380692 7569
150 10329412 8312908 3084551 6572442 ...
##  $ Classification_3     : num  0.679 0.769 0.829 0.782 0.655 ...
##  $ HighEngagement_3     : num  0.467 0.581 0.723 0.601 0.423 ...
##  $ LowEngagement_3      : num  0.304 0.404 0.246 0.383 0.373 ...
##  $ Distraction_3.1      : num  0.19228 0.00971 0 0.00641 0.19649 ...
##  $ Drowsy_3             : num  0.03692 0.00541 0.03101 0.00936 0.00805 ...
##  $ WorkloadFBDS_3       : num  0.774 0.524 0.606 0.456 0.591 ...
##  $ WorkloadBDS_3        : num  0.666 0.462 0.564 0.438 0.539 ...
##  $ WorkloadAverage_3    : num  0.72 0.493 0.585 0.447 0.565 ...
##  $ BrowFurrow_3         : num  4.50e-07 5.13e-05 5.19e-06 4.82e-05 1.28e-07 ...
##  $ BrowRaise_3          : num  14.392 18.991 1.193 0.183 3.108 ...
##  $ LipCornerDepressor_3 : num  0.02419 0.00552 0.01532 1.05246 0.0301 ...
##  $ Smile_3              : num  1.51e-03 9.10e-06 7.18e-09 2.22e-09 1.49e-06 ...
##  $ Valence_3            : num  0.0873 0 0 -0.1971 0 ...
##  $ Attention_3          : num  94.5 93.3 88.3 94.2 92.1 ...
##  $ Anger_3              : num  0.00217 0.02233 0.00198 0.00259 0.00182 ...
##  $ Sadness_3            : num  0.012474 0.000966 0.023233 0.025577 0.021448 ...
##  $ Disgust_3            : num  0.528 0.394 0.449 0.427 0.436 ...
##  $ Joy_3                : num  0.00154 0.0011 0.00176 0.00182 0.00172 ...
##  $ Surprise_3           : num  1.443 2.218 0.232 0.201 0.264 ...
##  $ Fear_3               : num  0.03541 22.29218 0.00425 0.00442 0.00428 ...
```

```
## $ Contempt_3          : num  0.195 0.189 0.194 0.194 0.193 ...
## $ PeakCount_3         : int  1 0 15 5 2 7 0 13 0 7 ...
## $ Peak.Min_3          : num  0.78 0 10.97 3.41 1.38 ...
## $ AveAmplitude_3      : num  0.0105 NA 0.0659 0.0666 0.0342 ...
## $ MaxAmplitude_3      : num  0.0105 NA 0.1728 0.1395 0.0629 ...
## $ GSR_CAL_3           : num  2.97 1.98 6.09 2.27 1.21 ...
```

```
train <- sample (1:nrow(RatingP3H1NPSdata3), size=0.8*nrow(RatingP3H1NPSdata3))
# training row indices
RatingP3H1NPS_train <- RatingP3H1NPSdata3[train, ] # training data
RatingP3H1NPS_test <- RatingP3H1NPSdata3[-train, ] # test data


hist.default(RatingP3H1NPS_train$NPSID_3, main = "Histogram of Rating Scores in Tr
ain Data For P3",
        xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP3H1NPS_test$NPSID_3, main = "Histogram of Rating Scores in Test
Data For P3",
        xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree
formula=NPSID_3 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA
+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize
+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVi
deos_3+RatingsComments_3+Descriptions_3+Cost_3+Confusion_3+Distraction_3+Enjo
yment_3+Interest_3+Tiredness_3+Frustration_3 + PupilLeft_3 + PupilRight_3 + minFix
ationStart_3 + sumFixationDuration_3 + Classification_3 + HighEngagement_3 + LowE
ngagement_3 + Distraction_3.1 + Drowsy_3 + WorkloadFBDS_3 + WorkloadBDS_3 +
WorkloadAverage_3 + BrowFurrow_3 + BrowRaise_3 + LipCornerDepressor_3 + Smile
_3 + Valence_3 + Attention_3 + Anger_3 + Sadness_3 + Disgust_3 + Joy_3 + Surprise_3
+ Fear_3 + Contempt_3 + PeakCount_3 + Peak.Min_3 + AveAmplitude_3 + MaxAmplit
ude_3 + GSR_CAL_3
```

```
RatingP3H1NPSdtree=rpart(formula,data=RatingP3H1NPS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model


#plot(dtree)

plot(RatingP3H1NPSdtree, uniform=TRUE,

   main="Classification Tree For P3 Rating")

text(RatingP3H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot

printcp(RatingP3H1NPSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RatingP3H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] GSR_CAL_3           Interest_3          LipCornerDepressor_3
## [4] MoneyPractice       SportsPerDay        Valence_3
##
## Root node error: 20/38 = 0.52632
##
## n= 38
##
##      CP nsplit rel error xerror    xstd
## 1 0.300      0       1.0   1.00 0.15390
## 2 0.150      1       0.7   1.15 0.15066
## 3 0.100      3       0.4   1.10 0.15218
## 4 0.050      4       0.3   1.05 0.15325
## 5 0.001      6       0.2   1.00 0.15390
```

```
plotcp(RatingP3H1NPSdtree)
```

```
#summary(RatingP3H1NPSdtree)
RatingP3H1NPSdtree
```

```
## n= 38
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 38 20 1 (0.47368421 0.34210526 0.18421053)
##    2) SportsPerDay< 45.5 14  2 1 (0.85714286 0.07142857 0.07142857)
##      4) Valence_3< 0.01135465 12  0 1 (1.00000000 0.00000000 0.00000000) *
##      5) Valence_3>=0.01135465 2  1 2 (0.00000000 0.50000000 0.50000000) *
##    3) SportsPerDay>=45.5 24 12 2 (0.25000000 0.50000000 0.25000000)
##      6) GSR_CAL_3< 0.4538561 5  1 1 (0.80000000 0.20000000 0.00000000) *
##      7) GSR_CAL_3>=0.4538561 19  8 2 (0.10526316 0.57894737 0.31578947)
##       14) LipCornerDepressor_3>=0.007129086 16  5 2 (0.12500000 0.68750000 0.187
50000)
##         28) Interest_3< 9.5 14  3 2 (0.14285714 0.78571429 0.07142857)
##           56) MoneyPractice>=17.5 12  1 2 (0.08333333 0.91666667 0.00000000) *
##           57) MoneyPractice< 17.5 2  1 1 (0.50000000 0.00000000 0.50000000) *
##         29) Interest_3>=9.5 2  0 3 (0.00000000 0.00000000 1.00000000) *
##       15) LipCornerDepressor_3< 0.007129086 3  0 3 (0.00000000 0.00000000 1.00000
000) *
```

```
RatingP3H1NPS_train$NPSID_3
```

```
##  [1] 1 1 2 1 2 1 1 2 3 3 1 1 1 2 1 2 2 1 1 3 3 1 2 2 2 2 2 2 1 1 3 1 1 3 1
## [36] 3 2 1
```

```
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP3H1NPSdtree)
```

```
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
##  [1] 1 1 1 1 2 1 2 2 2 3 1 1 1 2 1 2 2 1 1 3 3 1 2 2 2 2 2 2 1 1 3 1 1 3 1
## [36] 1 2 1
length(train_response_predicted)
## [1] 38
train_response_actual<- RatingP3H1NPS_train$NPSID_3 # actuals
train_response_actual
##  [1] 1 1 2 1 2 1 1 2 3 3 1 1 1 2 1 2 2 1 1 3 3 1 2 2 2 2 2 2 1 1 3 1 1 3 1
## [36] 3 2 1
length(train_response_actual)
## [1] 38
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.8947368
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.1052632
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating
Scores For P3 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
```

```
par(pty='s') #produces a square plot

#Obtain a confusion matrix
confusion.matrix <- table(RatingP3H1NPS_train$NPSID_3, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      1  2  3
```

```
##   1 17  1  0
##   2  1 12  0
##   3  1  1  5
```

```
#create a Actual vs Predicted Table from Decision Tree
(RatingP3H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predicted)))
```

```
##    train_response_actual train_response_predicted
## 1                      1                        1
## 2                      1                        1
## 3                      2                        1
## 4                      1                        1
## 5                      2                        2
## 6                      1                        1
## 7                      1                        2
## 8                      2                        2
## 9                      3                        2
## 10                     3                        3
## 11                     1                        1
## 12                     1                        1
## 13                     1                        1
## 14                     2                        2
## 15                     1                        1
## 16                     2                        2
## 17                     2                        2
## 18                     1                        1
## 19                     1                        1
## 20                     3                        3
## 21                     3                        3
## 22                     1                        1
## 23                     2                        2
```

```
## 24            2            2
## 25            2            2
## 26            2            2
## 27            2            2
## 28            2            2
## 29            1            1
## 30            1            1
## 31            3            3
## 32            1            1
## 33            1            1
## 34            3            3
## 35            1            1
## 36            3            1
## 37            2            2
## 38            1            1
```

```
names(RatingP3H1NPSdelta)<-c("RatingP3H1NPSactual","RatingP3H1NPSpredicted")
head(RatingP3H1NPSdelta)
```

```
##   RatingP3H1NPSactual RatingP3H1NPSpredicted
## 1          1            1
## 2          1            1
## 3          2            1
## 4          1            1
## 5          2            2
## 6          1            1
```

```
RatingP3H1NPSdelta$Match<-ifelse(RatingP3H1NPSdelta$RatingP3H1NPSactual==Ra
tingP3H1NPSdelta$RatingP3H1NPSpredicted,1,0)
head(RatingP3H1NPSdelta)
```

```
##   RatingP3H1NPSactual RatingP3H1NPSpredicted Match
## 1          1            1   1
## 2          1            1   1
```

```
## 3                  2                   1     0
## 4                  1                   1     1
## 5                  2                   2     1
## 6                  1                   1     1
tail(RatingP3H1NPSdelta)
##    RatingP3H1NPSactual RatingP3H1NPSpredicted Match
## 33                  1                   1     1
## 34                  3                   3     1
## 35                  1                   1     1
## 36                  3                   1     0
## 37                  2                   2     1
## 38                  1                   1     1
write.csv(RatingP3H1NPSdelta,file="RatingP3H1NPSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P3/H1NPS"
#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.8947368
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.1052632
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP3H1NPSdtree, RatingP3H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
##  [1] 2 1 3 1 1 1 1 1 2 2
length(test_response_predicted)
## [1] 10
```

```
test_response_actual<- RatingP3H1NPS_test$NPSID_3 # actuals
test_response_actual
##  [1] 1 1 1 2 1 1 1 1 2 1
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating S
cores For P3 3 Grouped",
    xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP3H1NPS_test$NPSID_3, test_response_predicted)
print(confusion.matrix)
##    test_response_predicted
##     1 2 3
##   1 5 2 1
##   2 1 1 0
#create a Actual vs Predicted Table from Decision Tree
(RatingP3H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicte
d)))
##    test_response_actual test_response_predicted
## 1                     1                       2
## 2                     1                       1
## 3                     1                       3
```

```
## 4         2         1
## 5         1         1
## 6         1         1
## 7         1         1
## 8         1         1
## 9         2         2
## 10        1         2
```

```
names(RatingP3H1NPSdelta)<-c("RatingP3H1NPSactual","RatingP3H1NPSpredicted")
head(RatingP3H1NPSdelta)
```

```
##   RatingP3H1NPSactual RatingP3H1NPSpredicted
## 1         1         2
## 2         1         1
## 3         1         3
## 4         2         1
## 5         1         1
## 6         1         1
```

```
RatingP3H1NPSdelta$Match<-ifelse(RatingP3H1NPSdelta$RatingP3H1NPSactual==RatingP3H1NPSdelta$RatingP3H1NPSpredicted,1,0)
head(RatingP3H1NPSdelta)
```

```
##   RatingP3H1NPSactual RatingP3H1NPSpredicted Match
## 1         1         2     0
## 2         1         1     1
## 3         1         3     0
## 4         2         1     0
## 5         1         1     1
## 6         1         1     1
```

```
tail(RatingP3H1NPSdelta)
```

```
##    RatingP3H1NPSactual RatingP3H1NPSpredicted Match
## 5         1         1     1
## 6         1         1     1
```

```
## 7                  1                   1     1
## 8                  1                   1     1
## 9                  2                   2     1
## 10                 1                   2     0
write.csv(RatingP3H1NPSdelta,file="RatingP3H1NPSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P3/H1NPS"
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
```

APPENDIX H

PREDICTION OF RATING CLASS FOR PRODUCT 4

RatingP4H1NPS3.R
irfan
Tue Mar 6 17:06:54 2018

```
#UoB Rating Analysis
#RatingP1H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P4/H1NPS")

# Upload the data
RatingP1H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP1H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP1H1NPSdata2 <- RatingP1H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,92, 94:105,244:273)]
names(RatingP1H1NPSdata2)
```

```
##  [1] "Subject"              "InternetPerDay"
##  [3] "OnlineShopPmont"      "PhotosVideosA"
##  [5] "DescriptionsA"        "RatingsCommentsA"
##  [7] "CostA"                "SportsPerWeek"
##  [9] "SportsPerDay"         "CoachDrivenS"
## [11] "EBraSize"             "SportsBraSize"
## [13] "SportsBraUse"         "SportsbrasN"
## [15] "MoneyWishes"          "MoneyPractice"
## [17] "ProductID_4"          "NPSID_4"
## [19] "RatingReason_4"       "ProductVideos_4"
## [21] "RatingsComments_4"    "Descriptions_4"
## [23] "Cost_4"               "Confusion_4"
## [25] "Distraction_4"        "Enjoyment_4"
## [27] "Interest_4"           "Tiredness_4"
## [29] "Frustration_4"        "PupilLeft_4"
## [31] "PupilRight_4"         "minFixationStart_4"
## [33] "sumFixationDuration_4" "Classification_4"
## [35] "HighEngagement_4"     "LowEngagement_4"
## [37] "Distraction_4.1"      "Drowsy_4"
## [39] "WorkloadFBDS_4"       "WorkloadBDS_4"
## [41] "WorkloadAverage_4"    "BrowFurrow_4"
## [43] "BrowRaise_4"          "LipCornerDepressor_4"
## [45] "Smile_4"              "Valence_4"
## [47] "Attention_4"          "Anger_4"
```

```
## [49] "Sadness_4"          "Disgust_4"
## [51] "Joy_4"              "Surprise_4"
## [53] "Fear_4"             "Contempt_4"
## [55] "PeakCount_4"        "Peak.Min_4"
## [57] "AveAmplitude_4"     "MaxAmplitude_4"
## [59] "GSR_CAL_4"
```

```
#If NPSID_4 have NAs, omit these rows that contain NA values
#RatingP1H1NPSdata3 <- RatingP1H1NPSdata2[!is.na(RatingP1H1NPSdata2$PeakCo
unt_4),]
RatingP1H1NPSdata3 <- RatingP1H1NPSdata2[!is.na(RatingP1H1NPSdata2$NPSID_4)
,]
#str(RatingP1H1NPSdata3)


#Descriptives
describe(RatingP1H1NPSdata3$NPSID_4)
```

```
## RatingP1H1NPSdata3$NPSID_4
##        n  missing distinct     Info     Mean      Gmd
##       47        0        3    0.509    1.277   0.4644
##
## Value          1     2     3
## Frequency     37     7     3
## Proportion 0.787 0.149 0.064
```

```
plot.default(RatingP1H1NPSdata3$NPSID_4, main = "Rating Scores For P1",
             xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP1H1NPSdata3$NPSID_4, main = "Histogram of Rating Scores For P
1",
             xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP1H1NPSdata3$NPSID_4) # returns the density data
plot(d) # plots the results
```

```
#Create training and test data
str(RatingP1H1NPSdata3)
## 'data.frame':    47 obs. of  59 variables:
##  $ Subject              : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay       : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont      : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA        : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA        : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA     : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA                : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek        : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay         : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS         : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize             : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize        : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse         : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN          : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes          : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice        : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ ProductID_4          : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ NPSID_4              : int  1 1 1 1 1 1 1 1 2 1 ...
##  $ RatingReason_4       : Factor w/ 45 levels "","138 reviews with a 4.3 star rating. I de
finitely would trust this product. Not a lot of sizes in stock but so m"| __truncated__,..: 15
33 27 28 5 35 30 31 16 7 ...
##  $ ProductVideos_4      : int  6 6 7 6 6 2 7 3 6 5 ...
##  $ RatingsComments_4    : int  7 7 4 5 7 6 7 7 6 7 ...
```

```
## $ Descriptions_4       : int  7 5 1 3 5 3 2 5 5 2 ...
## $ Cost_4               : int  7 4 7 7 7 6 4 6 6 6 ...
## $ Confusion_4          : int  NA 5 0 2 0 2 7 1 5 6 ...
## $ Distraction_4        : int  NA 4 5 7 0 2 8 4 3 8 ...
## $ Enjoyment_4          : int  NA 2 4 6 5 4 2 2 5 3 ...
## $ Interest_4           : int  NA 1 4 5 5 4 0 2 7 2 ...
## $ Tiredness_4          : int  NA 2 3 4 0 5 8 2 0 5 ...
## $ Frustration_4        : int  NA 2 0 2 1 1 9 0 3 5 ...
## $ PupilLeft_4          : num  2.64 2.35 3.46 2.56 2.57 ...
## $ PupilRight_4         : num  2.52 2.37 3.53 2.65 2.57 ...
## $ minFixationStart_4   : int  735 0 0 545 0 58 NA 678 0 2273 ...
## $ sumFixationDuration_4: int  3022499 4173224 13854115 7444840 11912835 9480
603 NA 8059042 9394230 2267260 ...
## $ Classification_4     : num  0.71 0.744 0.805 0.767 0.748 ...
## $ HighEngagement_4     : num  0.533 0.554 0.724 0.566 0.567 ...
## $ LowEngagement_4      : num  0.262 0.376 0.258 0.432 0.308 ...
## $ Distraction_4.1      : num  0.1485 0.04386 0 0.00245 0.12419 ...
## $ Drowsy_4             : num  0.056444 0.025307 0.017073 0 0.000692 ...
## $ WorkloadFBDS_4       : num  0.758 0.503 0.652 0.491 0.577 ...
## $ WorkloadBDS_4        : num  0.65 0.47 0.604 0.468 0.524 ...
## $ WorkloadAverage_4    : num  0.704 0.487 0.628 0.479 0.551 ...
## $ BrowFurrow_4         : num  1.50e-06 2.46e-05 1.73e-06 1.21e-03 9.71e-06 ...
## $ BrowRaise_4          : num  3.529 19.279 1.882 0.109 8.477 ...
## $ LipCornerDepressor_4 : num  0.01317 0.00635 0.09995 1.39055 4.37737 ...
## $ Smile_4              : num  3.54e-03 4.62e-02 2.81e-05 3.11e-05 4.06e-03 ...
## $ Valence_4            : num  0 -0.540035 -0.000616 -1.840559 -2.377205 ...
## $ Attention_4          : num  64.7 74.6 95 96.4 87.5 ...
## $ Anger_4              : num  0.00349 0.02002 0.00712 0.02863 0.00454 ...
## $ Sadness_4            : num  0.013057 0.000747 0.01695 0.0233 0.019768 ...
## $ Disgust_4            : num  0.461 3.227 0.661 0.418 0.53 ...
```

```
##  $ Joy_4              : num  0.00165 0.00218 0.00281 0.00194 0.0018 ...
##  $ Surprise_4          : num  0.353 3.394 0.739 0.254 1.032 ...
##  $ Fear_4              : num  0.30963 20.5329 0.00393 0.0043 0.00368 ...
##  $ Contempt_4          : num  29.058 0.199 0.193 0.792 0.193 ...
##  $ PeakCount_4         : int  0 1 12 8 3 12 0 12 0 8 ...
##  $ Peak.Min_4          : num  0 0.68 8.25 5.56 2.07 8.33 0 8.24 0 5.48 ...
##  $ AveAmplitude_4      : num  NA 0.0135 0.185 0.0536 0.058 ...
##  $ MaxAmplitude_4      : num  NA 0.0135 0.4158 0.1785 0.114 ...
##  $ GSR_CAL_4           : num  2.31 2.4 5.45 2.54 1.72 ...
```

```
train <- sample (1:nrow(RatingP1H1NPSdata3), size=0.8*nrow(RatingP1H1NPSdata3)) # training row indices

RatingP1H1NPS_train <- RatingP1H1NPSdata3[train, ] # training data

RatingP1H1NPS_test <- RatingP1H1NPSdata3[-train, ] # test data


hist.default(RatingP1H1NPS_train$NPSID_4, main = "Histogram of Rating Scores in Train Data For P1",
         xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP1H1NPS_test$NPSID_4, main = "Histogram of Rating Scores in Test Data For P1",
         xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree

formula=NPSID_4 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVideos_4+RatingsComments_4+Descriptions_4+Cost_4+Confusion_4+Distraction_4+Enjoyment_4+Interest_4+Tiredness_4+Frustration_4 + PupilLeft_4 + PupilRight_4 + minFixationStart_4 + sumFixationDuration_4 + Classification_4 + HighEngagement_4 + LowEngagement_4 + Distraction_4.1 + Drowsy_4 + WorkloadFBDS_4 + WorkloadBDS_4 + WorkloadAverage_4 + BrowFurrow_4 + BrowRaise_4 + LipCornerDepressor_4 + Smile_4 + Valence_4 + Attention_4 + Anger_4 + Sadness_4 + Disgust_4 + Joy_4 + Surprise_4
```

```
+ Fear_4 + Contempt_4 + PeakCount_4 + Peak.Min_4 + AveAmplitude_4 + MaxAmplit
ude_4 + GSR_CAL_4


RatingP1H1NPSdtree=rpart(formula,data=RatingP1H1NPS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model


#plot(dtree)
plot(RatingP1H1NPSdtree, uniform=TRUE,
   main="Classification Tree For P1 Rating")
text(RatingP1H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP1H1NPSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RatingP1H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Drowsy_4    Enjoyment_4 Surprise_4
##
## Root node error: 8/37 = 0.21622
##
## n= 37
##
##       CP nsplit rel error xerror    xstd
## 1 0.3125      0     1.000  1.000 0.31301
## 2 0.1250      2     0.375  1.375 0.34753
## 3 0.0010      3     0.250  1.625 0.36298
```

```
plotcp(RatingP1H1NPSdtree)
```

```
#summary(RatingP1H1NPSdtree)
RatingP1H1NPSdtree
## n= 37
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 37 8 1 (0.78378378 0.16216216 0.05405405)
##    2) Enjoyment_4< 4.5 20 1 1 (0.95000000 0.00000000 0.05000000) *
##    3) Enjoyment_4>=4.5 17 7 1 (0.58823529 0.35294118 0.05882353)
##      6) Surprise_4>=1.512639 9 0 1 (1.00000000 0.00000000 0.00000000) *
##      7) Surprise_4< 1.512639 8 2 2 (0.12500000 0.75000000 0.12500000)
##       14) Drowsy_4>=0.03386563 2 1 1 (0.50000000 0.00000000 0.50000000) *
##       15) Drowsy_4< 0.03386563 6 0 2 (0.00000000 1.00000000 0.00000000) *
RatingP1H1NPS_train$NPSID_4
##  [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 3 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 3 1 1
## [36] 1 2
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP1H1NPSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
##  [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 3 3 1 1
## [36] 1 2
length(train_response_predicted)
## [1] 37
```

```
train_response_actual<- RatingP1H1NPS_train$NPSID_4 # actuals
train_response_actual
## [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 3 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 3 1 1
## [36] 1 2
length(train_response_actual)
## [1] 37
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9459459
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.05405405
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating Scores For P1 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP1H1NPS_train$NPSID_4, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      1  2  3
##   1 28  0  1
##   2  0  6  0
##   3  1  0  1
#create a Actual vs Predicted Table from Decision Tree
(RatingP1H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predic
ted)))
##    train_response_actual train_response_predicted
```

```
## 1     1     1
## 2     1     1
## 3     1     1
## 4     1     1
## 5     2     2
## 6     1     1
## 7     1     1
## 8     1     1
## 9     1     1
## 10    1     1
## 11    1     1
## 12    1     1
## 13    2     2
## 14    2     2
## 15    1     1
## 16    3     1
## 17    2     2
## 18    1     1
## 19    1     1
## 20    1     1
## 21    1     1
## 22    1     1
## 23    1     1
## 24    1     1
## 25    2     2
## 26    1     1
## 27    1     1
## 28    1     1
## 29    1     1
## 30    1     1
```

```
## 31                1                1
## 32                1                3
## 33                3                3
## 34                1                1
## 35                1                1
## 36                1                1
## 37                2                2
```

```
names(RatingP1H1NPSdelta)<-c("RatingP1H1NPSactual","RatingP1H1NPSpredicted")
head(RatingP1H1NPSdelta)
```

```
##   RatingP1H1NPSactual RatingP1H1NPSpredicted
## 1               1                1
## 2               1                1
## 3               1                1
## 4               1                1
## 5               2                2
## 6               1                1
```

```
RatingP1H1NPSdelta$Match<-ifelse(RatingP1H1NPSdelta$RatingP1H1NPSactual==RatingP1H1NPSdelta$RatingP1H1NPSpredicted,1,0)
head(RatingP1H1NPSdelta)
```

```
##   RatingP1H1NPSactual RatingP1H1NPSpredicted Match
## 1               1                1     1
## 2               1                1     1
## 3               1                1     1
## 4               1                1     1
## 5               2                2     1
## 6               1                1     1
```

```
tail(RatingP1H1NPSdelta)
```

```
##    RatingP1H1NPSactual RatingP1H1NPSpredicted Match
## 32               1                3     0
## 33               3                3     1
```

```
## 34                1                   1     1
## 35                1                   1     1
## 36                1                   1     1
## 37                2                   2     1
```

```
write.csv(RatingP1H1NPSdelta,file="RatingP1H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P4/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
```

```
## [1] 0.9459459
```

```
mean (train_response_actual != train_response_predicted) # Misclassification %
```

```
## [1] 0.05405405
```

```
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP1H1NPSdtree, RatingP1H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
```

```
##  [1] 2 2 1 1 1 1 1 1 1 1
```

```
length(test_response_predicted)
```

```
## [1] 10
```

```
test_response_actual<- RatingP1H1NPS_test$NPSID_4 # actuals
test_response_actual
```

```
##  [1] 1 1 1 3 1 1 1 1 1 2
```

```
length(test_response_actual)
```

```
## [1] 10
```

```
mean (test_response_actual == test_response_predicted) # Accuracy %
```

```
## [1] 0.6
```

```
mean (test_response_actual != test_response_predicted) # Misclassification %
```

```
## [1] 0.4
```

```
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating Scores For P1 3 Grouped",
     xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP1H1NPS_test$NPSID_4, test_response_predicted)
print(confusion.matrix)
```

```
##    test_response_predicted
##     1 2
##   1 6 2
##   2 1 0
##   3 1 0
```

```
#create a Actual vs Predicted Table from Decision Tree
(RatingP1H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted)))
```

```
##    test_response_actual test_response_predicted
## 1                     1                       2
## 2                     1                       2
## 3                     1                       1
## 4                     3                       1
## 5                     1                       1
## 6                     1                       1
## 7                     1                       1
## 8                     1                       1
## 9                     1                       1
## 10                    2                       1
```

```
names(RatingP1H1NPSdelta)<-c("RatingP1H1NPSactual","RatingP1H1NPSpredicted")
head(RatingP1H1NPSdelta)
```

```
##   RatingP1H1NPSactual RatingP1H1NPSpredicted
## 1                   1                      2
## 2                   1                      2
## 3                   1                      1
## 4                   3                      1
## 5                   1                      1
## 6                   1                      1
```

```
RatingP1H1NPSdelta$Match<-ifelse(RatingP1H1NPSdelta$RatingP1H1NPSactual==RatingP1H1NPSdelta$RatingP1H1NPSpredicted,1,0)
head(RatingP1H1NPSdelta)
```

```
##   RatingP1H1NPSactual RatingP1H1NPSpredicted Match
## 1                   1                      2     0
## 2                   1                      2     0
## 3                   1                      1     1
## 4                   3                      1     0
## 5                   1                      1     1
## 6                   1                      1     1
```

```
tail(RatingP1H1NPSdelta)
```

```
##    RatingP1H1NPSactual RatingP1H1NPSpredicted Match
## 5                    1                      1     1
## 6                    1                      1     1
## 7                    1                      1     1
## 8                    1                      1     1
## 9                    1                      1     1
## 10                   2                      1     0
```

```
write.csv(RatingP1H1NPSdelta,file="RatingP1H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P4/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
```

]

APPENDIX I

PREDICTION OF RATING CLASS FOR PRODUCT 5

RatingP5H1NPS3.R
irfan
Tue Mar 6 18:03:00 2018

```
#UoB Rating Analysis
#RatingP5H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P5/H1NPS")

# Upload the data
RatingP5H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP5H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP5H1NPSdata2 <- RatingP5H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,111,11
3:122,275:304)]
names(RatingP5H1NPSdata2)
## [1] "Subject"              "InternetPerDay"
## [3] "OnlineShopPmont"      "PhotosVideosA"
## [5] "DescriptionsA"        "RatingsCommentsA"
## [7] "CostA"                "SportsPerWeek"
## [9] "SportsPerDay"         "CoachDrivenS"
## [11] "EBraSize"             "SportsBraSize"
## [13] "SportsBraUse"         "SportsbrasN"
## [15] "MoneyWishes"          "MoneyPractice"
## [17] "NPSID_5"              "ProductVideos_5"
## [19] "RatingsComments_5"    "Descriptions_5"
## [21] "Cost_5"               "Confusion_5"
## [23] "Distraction_5"        "Enjoyment_5"
## [25] "Interest_5"           "Tiredness_5"
## [27] "Frustration_5"        "PupilLeft_5"
## [29] "PupilRight_5"         "minFixationStart_5"
## [31] "sumFixationDuration_5" "Classification_5"
## [33] "HighEngagement_5"     "LowEngagement_5"
## [35] "Distraction_5.1"      "Drowsy_5"
## [37] "WorkloadFBDS_5"       "WorkloadBDS_5"
## [39] "WorkloadAverage_5"    "BrowFurrow_5"
## [41] "BrowRaise_5"          "LipCornerDepressor_5"
## [43] "Smile_5"              "Valence_5"
## [45] "Attention_5"          "Anger_5"
## [47] "Sadness_5"            "Disgust_5"
```

```
## [49] "Joy_5"               "Surprise_5"
## [51] "Fear_5"              "Contempt_5"
## [53] "PeakCount_5"         "Peak.Min_5"
## [55] "AveAmplitude_5"      "MaxAmplitude_5"
## [57] "GSR_CAL_5"
```

```
#If NPSID_5 have NAs, omit these rows that contain NA values
#RatingP5H1NPSdata3 <- RatingP5H1NPSdata2[!is.na(RatingP5H1NPSdata2$PeakCo
unt_5),]
RatingP5H1NPSdata3 <- RatingP5H1NPSdata2[!is.na(RatingP5H1NPSdata2$NPSID_5)
,]
#str(RatingP5H1NPSdata3)


#Descriptives
describe(RatingP5H1NPSdata3$NPSID_5)
```

```
## RatingP5H1NPSdata3$NPSID_5
##        n  missing distinct     Info     Mean      Gmd
##       48        0        3    0.667    1.438   0.6622
##
## Value          1     2     3
## Frequency     33     9     6
## Proportion 0.688 0.188 0.125
```

```
plot.default(RatingP5H1NPSdata3$NPSID_5, main = "Rating Scores For P5",
          xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP5H1NPSdata3$NPSID_5, main = "Histogram of Rating Scores For P
5",
          xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP5H1NPSdata3$NPSID_5) # returns the density data
```

```
plot(d) # plots the results
```

```
#Create training and test data
str(RatingP5H1NPSdata3)
## 'data.frame':    48 obs. of  57 variables:
##  $ Subject            : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay      : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont     : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA       : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA       : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA    : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA               : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek       : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay        : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS        : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize            : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize       : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse        : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN         : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes         : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice       : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ NPSID_5             : int  1 3 2 2 2 1 1 1 3 1 ...
##  $ ProductVideos_5     : int  7 7 7 7 6 5 7 5 7 6 ...
##  $ RatingsComments_5   : int  7 6 1 7 7 5 1 5 6 7 ...
##  $ Descriptions_5      : int  7 7 4 6 7 1 1 6 7 3 ...
##  $ Cost_5              : int  7 7 6 6 6 3 5 6 6 4 ...
##  $ Confusion_5         : int  NA 2 2 3 0 3 8 0 0 2 ...
##  $ Distraction_5       : int  NA 2 0 3 0 5 8 1 0 6 ...
##  $ Enjoyment_5         : int  NA 9 8 8 8 4 6 1 8 3 ...
```

```
##  $ Interest_5           : int  NA 10 8 9 9 4 5 3 8 4 ...
##  $ Tiredness_5          : int  NA 1 4 2 0 6 7 3 0 2 ...
##  $ Frustration_5        : int  NA 1 0 3 0 1 9 0 0 2 ...
##  $ PupilLeft_5          : num  2.58 2.38 3.11 2.57 2.15 ...
##  $ PupilRight_5         : num  2.53 2.3 3.2 2.62 2.67 ...
##  $ minFixationStart_5   : int  2100 952 0 484 44 47 217 NA 347 257 ...
##  $ sumFixationDuration_5: int  659492 5022918 8701748 14199265 9481004 114295
13 2055384 NA 5872100 4595927 ...
##  $ Classification_5     : num  0.689 0.728 0.824 0.798 0.755 ...
##  $ HighEngagement_5     : num  0.514 0.459 0.721 0.611 0.587 ...
##  $ LowEngagement_5      : num  0.263 0.529 0.273 0.388 0.339 ...
##  $ Distraction_5.1      : num  2.23e-01 3.38e-03 4.47e-06 6.35e-04 5.01e-02 ...
##  $ Drowsy_5             : num  5.02e-06 8.99e-03 5.96e-03 0.00 2.36e-02 ...
##  $ WorkloadFBDS_5       : num  0.85 0.552 0.562 0.399 0.625 ...
##  $ WorkloadBDS_5        : num  0.669 0.503 0.531 0.388 0.545 ...
##  $ WorkloadAverage_5    : num  0.759 0.527 0.547 0.393 0.585 ...
##  $ BrowFurrow_5         : num  8.74e-07 2.61e-06 2.52e-05 7.17e-05 8.53e-07 ...
##  $ BrowRaise_5          : num  4.833 10.253 2.899 0.475 2.585 ...
##  $ LipCornerDepressor_5 : num  0.20152 0.00644 0.19835 0.98534 5.73515 ...
##  $ Smile_5              : num  1.30e-04 6.72e-04 4.93e-06 1.96e-06 3.50e-05 ...
##  $ Valence_5            : num  0 0 -3.8 -1.05 -2.12 ...
##  $ Attention_5          : num  87 86.8 83.4 89.1 92.4 ...
##  $ Anger_5              : num  0.0027 0.01673 0.08317 0.01268 0.00307 ...
##  $ Sadness_5            : num  0.01535 0.00419 0.01823 0.01817 0.02557 ...
##  $ Disgust_5            : num  0.455 0.554 0.476 0.684 0.425 ...
##  $ Joy_5                : num  0.00166 0.00183 0.0021 0.00296 0.00171 ...
##  $ Surprise_5           : num  0.315 0.98 0.475 1.42 0.232 ...
##  $ Fear_5               : num  0.08444 12.10382 0.00471 0.00323 0.00446 ...
##  $ Contempt_5           : num  0.915 0.195 0.193 2.85 0.193 ...
##  $ PeakCount_5          : int  7 1 5 13 1 13 0 14 0 14 ...
```

```
## $ Peak.Min_5          : num  7.86 0.7 3.71 9.58 0.76 8.86 0 9.57 0 9.74 ...
## $ AveAmplitude_5       : num  0.0848 0.0135 0.0766 0.1131 0.0538 ...
## $ MaxAmplitude_5       : num  0.2175 0.0135 0.3942 0.513 0.0538 ...
## $ GSR_CAL_5           : num  2.58 2.13 4.67 2.98 1.15 ...
```

```
train <- sample (1:nrow(RatingP5H1NPSdata3), size=0.8*nrow(RatingP5H1NPSdata3)) # training row indices

RatingP5H1NPS_train <- RatingP5H1NPSdata3[train, ] # training data

RatingP5H1NPS_test <- RatingP5H1NPSdata3[-train, ] # test data


hist.default(RatingP5H1NPS_train$NPSID_5, main = "Histogram of Rating Scores in Train Data For P5",
             xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP5H1NPS_test$NPSID_5, main = "Histogram of Rating Scores in Test Data For P5",
             xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree

formula=NPSID_5 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVideos_5+RatingsComments_5+Descriptions_5+Cost_5+Confusion_5+Distraction_5+Enjoyment_5+Interest_5+Tiredness_5+Frustration_5 + PupilLeft_5 + PupilRight_5 + minFixationStart_5 + sumFixationDuration_5 + Classification_5 + HighEngagement_5 + LowEngagement_5 + Distraction_5.1 + Drowsy_5 + WorkloadFBDS_5 + WorkloadBDS_5 + WorkloadAverage_5 + BrowFurrow_5 + BrowRaise_5 + LipCornerDepressor_5 + Smile_5 + Valence_5 + Attention_5 + Anger_5 + Sadness_5 + Disgust_5 + Joy_5 + Surprise_5 + Fear_5 + Contempt_5 + PeakCount_5 + Peak.Min_5 + AveAmplitude_5 + MaxAmplitude_5 + GSR_CAL_5


RatingP5H1NPSdtree=rpart(formula,data=RatingP5H1NPS_train,method="class",control=rpart.control(minsplit=2,cp=0.001)) # build the model
```

```
#plot(dtree)
plot(RatingP5H1NPSdtree, uniform=TRUE,
    main="Classification Tree For P5 Rating")
text(RatingP5H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP5H1NPSdtree)
##
## Classification tree:
## rpart(formula = formula, data = RatingP5H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 2, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] CoachDrivenS    Enjoyment_5     OnlineShopPmont PhotosVideosA
## [5] PupilLeft_5     SportsPerWeek
##
## Root node error: 9/38 = 0.23684
##
## n= 38
##
##         CP nsplit rel error  xerror    xstd
## 1 0.444444      0   1.00000 1.00000 0.29120
## 2 0.222222      1   0.55556 0.55556 0.23153
## 3 0.111111      2   0.33333 0.55556 0.23153
## 4 0.055556      4   0.11111 0.66667 0.24976
## 5 0.001000      6   0.00000 1.00000 0.29120
plotcp(RatingP5H1NPSdtree)
```

```
#summary(RatingP5H1NPSdtree)
RatingP5H1NPSdtree
## n= 38
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 38 9 1 (0.76315789 0.15789474 0.07894737)
##    2) Enjoyment_5< 6.5 30 2 1 (0.93333333 0.03333333 0.03333333)
##      4) PhotosVideosA>=3 29 1 1 (0.96551724 0.03448276 0.00000000)
##        8) PupilLeft_5>=2.339682 26 0 1 (1.00000000 0.00000000 0.00000000) *
##        9) PupilLeft_5< 2.339682 3 1 1 (0.66666667 0.33333333 0.00000000)
##         18) OnlineShopPmont< 3 2 0 1 (1.00000000 0.00000000 0.00000000) *
##         19) OnlineShopPmont>=3 1 0 2 (0.00000000 1.00000000 0.00000000) *
##      5) PhotosVideosA< 3 1 0 3 (0.00000000 0.00000000 1.00000000) *
##    3) Enjoyment_5>=6.5 8 3 2 (0.12500000 0.62500000 0.25000000)
##      6) CoachDrivenS< 60.5 5 0 2 (0.00000000 1.00000000 0.00000000) *
##      7) CoachDrivenS>=60.5 3 1 3 (0.33333333 0.00000000 0.66666667)
##       14) SportsPerWeek< 2.5 1 0 1 (1.00000000 0.00000000 0.00000000) *
##       15) SportsPerWeek>=2.5 2 0 3 (0.00000000 0.00000000 1.00000000) *
RatingP5H1NPS_train$NPSID_5
##  [1] 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 2 1 1 1 2 1 3 1 1 1 3 1 1 1 1 1 2 1 1
## [36] 1 1 3
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP5H1NPSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
##  [1] 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 2 1 1 1 2 1 3 1 1 1 3 1 1 1 1 1 2 1 1
```

```
## [36] 1 1 3
length(train_response_predicted)
## [1] 38
train_response_actual<- RatingP5H1NPS_train$NPSID_5 # actuals
train_response_actual
##  [1] 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 2 1 1 1 2 1 3 1 1 1 3 1 1 1 1 1 2 1 1
## [36] 1 1 3
length(train_response_actual)
## [1] 38
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 1
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating Scores For P5 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP5H1NPS_train$NPSID_5, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      1  2  3
##   1 29  0  0
##   2  0  6  0
##   3  0  0  3
#create a Actual vs Predicted Table from Decision Tree
```

```
(RatingP5H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predicted)))
```

```
##    train_response_actual train_response_predicted
## 1                      1                        1
## 2                      1                        1
## 3                      1                        1
## 4                      1                        1
## 5                      1                        1
## 6                      1                        1
## 7                      1                        1
## 8                      2                        2
## 9                      1                        1
## 10                     1                        1
## 11                     1                        1
## 12                     2                        2
## 13                     2                        2
## 14                     1                        1
## 15                     1                        1
## 16                     1                        1
## 17                     2                        2
## 18                     1                        1
## 19                     1                        1
## 20                     1                        1
## 21                     2                        2
## 22                     1                        1
## 23                     3                        3
## 24                     1                        1
## 25                     1                        1
## 26                     1                        1
## 27                     3                        3
```

```
## 28          1          1
## 29          1          1
## 30          1          1
## 31          1          1
## 32          1          1
## 33          2          2
## 34          1          1
## 35          1          1
## 36          1          1
## 37          1          1
## 38          3          3
```

```
names(RatingP5H1NPSdelta)<-c("RatingP5H1NPSactual","RatingP5H1NPSpredicted")
head(RatingP5H1NPSdelta)
```

```
##   RatingP5H1NPSactual RatingP5H1NPSpredicted
## 1          1          1
## 2          1          1
## 3          1          1
## 4          1          1
## 5          1          1
## 6          1          1
```

```
RatingP5H1NPSdelta$Match<-ifelse(RatingP5H1NPSdelta$RatingP5H1NPSactual==RatingP5H1NPSdelta$RatingP5H1NPSpredicted,1,0)
head(RatingP5H1NPSdelta)
```

```
##   RatingP5H1NPSactual RatingP5H1NPSpredicted Match
## 1          1          1    1
## 2          1          1    1
## 3          1          1    1
## 4          1          1    1
## 5          1          1    1
## 6          1          1    1
```

```
tail(RatingP5H1NPSdelta)
```

```
##    RatingP5H1NPSactual RatingP5H1NPSpredicted Match
## 33                   2                      2     1
## 34                   1                      1     1
## 35                   1                      1     1
## 36                   1                      1     1
## 37                   1                      1     1
## 38                   3                      3     1
```

```
write.csv(RatingP5H1NPSdelta,file="RatingP5H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P5/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
```

```
## [1] 1
```

```
mean (train_response_actual != train_response_predicted) # Misclassification %
```

```
## [1] 0
```

```
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP5H1NPSdtree, RatingP5H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
```

```
##  [1] 2 2 3 3 1 1 2 1 1 1
```

```
length(test_response_predicted)
```

```
## [1] 10
```

```
test_response_actual<- RatingP5H1NPS_test$NPSID_5 # actuals
test_response_actual
```

```
##  [1] 2 2 3 2 1 3 1 1 3 1
```

```
length(test_response_actual)
```

```
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating Scores For P5 3 Grouped",
     xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP5H1NPS_test$NPSID_5, test_response_predicted)
print(confusion.matrix)
##    test_response_predicted
##     1 2 3
##   1 3 1 0
##   2 0 2 1
##   3 2 0 1
#create a Actual vs Predicted Table from Decision Tree
(RatingP5H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted)))
##    test_response_actual test_response_predicted
## 1                     2                       2
## 2                     2                       2
## 3                     3                       3
## 4                     2                       3
## 5                     1                       1
## 6                     3                       1
```

```
## 7                  1                    2
## 8                  1                    1
## 9                  3                    1
## 10                 1                    1
```

```
names(RatingP5H1NPSdelta)<-c("RatingP5H1NPSactual","RatingP5H1NPSpredicted")
head(RatingP5H1NPSdelta)
```

```
##   RatingP5H1NPSactual RatingP5H1NPSpredicted
## 1                 2                    2
## 2                 2                    2
## 3                 3                    3
## 4                 2                    3
## 5                 1                    1
## 6                 3                    1
```

```
RatingP5H1NPSdelta$Match<-ifelse(RatingP5H1NPSdelta$RatingP5H1NPSactual==Ra
tingP5H1NPSdelta$RatingP5H1NPSpredicted,1,0)
head(RatingP5H1NPSdelta)
```

```
##   RatingP5H1NPSactual RatingP5H1NPSpredicted Match
## 1                 2                    2     1
## 2                 2                    2     1
## 3                 3                    3     1
## 4                 2                    3     0
## 5                 1                    1     1
## 6                 3                    1     0
```

```
tail(RatingP5H1NPSdelta)
```

```
##    RatingP5H1NPSactual RatingP5H1NPSpredicted Match
## 5                  1                    1     1
## 6                  3                    1     0
## 7                  1                    2     0
## 8                  1                    1     1
## 9                  3                    1     0
```

```
## 10                  1                     1     1
write.csv(RatingP5H1NPSdelta,file="RatingP5H1NPSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P5/H1NPS"
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
```

APPENDIX J

PREDICTION OF RATING CLASS FOR PRODUCT 6

RatingP6H1NPS3.R
irfan
Tue Mar 6 17:48:02 2018

```
#UoB Rating Analysis
#RatingP6H1NPS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P6/H1NPS")

# Upload the data
RatingP6H1NPSdata <- read.csv("UoB.v13.csv", header=TRUE)

#Display header names
```

```
#names(RatingP6H1NPSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RatingP6H1NPSdata2 <- RatingP6H1NPSdata[c(1,12,13,18:21,25,26,30:34,36,37,125,127:136,306:335)]
names(RatingP6H1NPSdata2)
```

```
##  [1] "Subject"              "InternetPerDay"
##  [3] "OnlineShopPmont"      "PhotosVideosA"
##  [5] "DescriptionsA"        "RatingsCommentsA"
##  [7] "CostA"                "SportsPerWeek"
##  [9] "SportsPerDay"         "CoachDrivenS"
## [11] "EBraSize"             "SportsBraSize"
## [13] "SportsBraUse"         "SportsbrasN"
## [15] "MoneyWishes"          "MoneyPractice"
## [17] "NPSID_6"              "ProductVideos_6"
## [19] "RatingsComments_6"    "Descriptions_6"
## [21] "Cost_6"               "Confusion_6"
## [23] "Distraction_6"        "Enjoyment_6"
## [25] "Interest_6"           "Tiredness_6"
## [27] "Frustration_6"        "PupilLeft_6"
## [29] "PupilRight_6"         "minFixationStart_6"
## [31] "sumFixationDuration_6" "Classification_6"
## [33] "HighEngagement_6"     "LowEngagement_6"
## [35] "Distraction_6.1"      "Drowsy_6"
## [37] "WorkloadFBDS_6"       "WorkloadBDS_6"
## [39] "WorkloadAverage_6"    "BrowFurrow_6"
## [41] "BrowRaise_6"          "LipCornerDepressor_6"
## [43] "Smile_6"              "Valence_6"
## [45] "Attention_6"          "Anger_6"
## [47] "Sadness_6"            "Disgust_6"
```

```
## [49] "Joy_6"                "Surprise_6"
## [51] "Fear_6"               "Contempt_6"
## [53] "PeakCount_6"          "Peak.Min_6"
## [55] "AveAmplitude_6"       "MaxAmplitude_6"
## [57] "GSR_CAL_6"
```

```
#If NPSID_6 have NAs, omit these rows that contain NA values
#RatingP6H1NPSdata3 <- RatingP6H1NPSdata2[!is.na(RatingP6H1NPSdata2$PeakCount_6),]
RatingP6H1NPSdata3 <- RatingP6H1NPSdata2[!is.na(RatingP6H1NPSdata2$NPSID_6),]
#str(RatingP6H1NPSdata3)


#Descriptives
describe(RatingP6H1NPSdata3$NPSID_6)
```

```
## RatingP6H1NPSdata3$NPSID_6
##        n  missing distinct     Info     Mean      Gmd
##       47        0        3    0.382     1.17   0.3016
##
## Value          1     2     3
## Frequency     40     6     1
## Proportion 0.851 0.128 0.021
```

```
plot.default(RatingP6H1NPSdata3$NPSID_6, main = "Rating Scores For P6",
        xlab = "Index", ylab = "Rating Scores")
```

```
hist.default(RatingP6H1NPSdata3$NPSID_6, main = "Histogram of Rating Scores For P6",
        xlab = "Rating", ylab = "Frequency")
```

```
d <- density(RatingP6H1NPSdata3$NPSID_6) # returns the density data
```

```
plot(d) #plots the results
```

```
#Create training and test data
str(RatingP6H1NPSdata3)
## 'data.frame':	47 obs. of  57 variables:
##  $ Subject           : Factor w/ 48 levels "s01","s02","s03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ InternetPerDay     : int  4 13 8 11 5 17 12 9 10 8 ...
##  $ OnlineShopPmont     : int  2 10 5 2 5 4 1 6 8 6 ...
##  $ PhotosVideosA      : int  7 2 7 7 6 7 7 5 4 7 ...
##  $ DescriptionsA      : int  7 6 5 7 6 6 4 6 5 5 ...
##  $ RatingsCommentsA    : int  7 6 6 7 7 7 7 5 7 6 ...
##  $ CostA              : int  7 7 7 7 7 6 5 4 4 5 ...
##  $ SportsPerWeek      : int  5 7 3 5 4 13 3 2 6 10 ...
##  $ SportsPerDay       : int  180 90 60 45 60 124 180 14 74 75 ...
##  $ CoachDrivenS       : int  80 70 0 0 50 90 59 20 71 NA ...
##  $ EBraSize           : int  34 36 34 36 36 32 32 34 38 32 ...
##  $ SportsBraSize      : int  3 3 4 3 4 1 3 3 5 3 ...
##  $ SportsBraUse       : int  5 5 5 5 5 5 4 5 5 4 ...
##  $ SportsbrasN        : int  5 5 3 5 5 5 5 3 5 3 ...
##  $ MoneyWishes        : int  60 30 30 25 50 35 30 60 50 50 ...
##  $ MoneyPractice      : int  35 30 20 15 25 55 10 30 40 15 ...
##  $ NPSID_6            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ProductVideos_6     : int  7 6 7 5 6 5 7 7 6 6 ...
##  $ RatingsComments_6   : int  7 6 7 7 7 6 7 7 7 7 ...
##  $ Descriptions_6      : int  7 6 1 4 6 6 7 6 6 5 ...
##  $ Cost_6             : int  7 2 1 6 6 1 4 3 6 2 ...
##  $ Confusion_6        : int  NA 9 0 3 0 4 7 0 0 8 ...
##  $ Distraction_6      : int  NA 7 0 4 0 3 6 2 0 8 ...
##  $ Enjoyment_6        : int  NA 2 0 7 7 4 4 4 2 1 ...
```

```
## $ Interest_6          : int NA 0 0 6 7 5 3 4 2 3 ...
## $ Tiredness_6         : int NA 2 3 4 0 6 8 5 0 2 ...
## $ Frustration_6       : int NA 2 0 7 0 1 10 0 0 2 ...
## $ PupilLeft_6         : num 2.71 2.36 3.15 2.4 2.38 ...
## $ PupilRight_6        : num 2.63 2.3 3.29 2.61 2.57 ...
## $ minFixationStart_6  : int 600 108 66 111 155 0 668 NA 82 56 ...
## $ sumFixationDuration_6: int 3830744 4660588 9122963 11993323 10023343 1057
8276 8886778 NA 8000907 2239013 ...
## $ Classification_6    : num 0.729 0.702 0.76 0.741 0.8 ...
## $ HighEngagement_6    : num 0.567 0.464 0.578 0.538 0.655 ...
## $ LowEngagement_6     : num 0.204 0.464 0.4 0.416 0.307 ...
## $ Distraction_6.1     : num 1.58e-01 7.18e-02 2.17e-07 4.61e-02 3.74e-02 ...
## $ Drowsy_6            : num 7.10e-02 5.84e-05 2.21e-02 0.00 3.53e-04 ...
## $ WorkloadFBDS_6      : num 0.635 0.501 0.54 0.381 0.587 ...
## $ WorkloadBDS_6       : num 0.532 0.439 0.492 0.356 0.477 ...
## $ WorkloadAverage_6   : num 0.583 0.47 0.516 0.368 0.532 ...
## $ BrowFurrow_6        : num 5.51e-07 5.87e-06 4.22e-05 1.64e-04 2.85e-05 ...
## $ BrowRaise_6         : num 7.69 10.684 5.899 0.908 9.662 ...
## $ LipCornerDepressor_6 : num 2.51284 0.00489 0.40518 1.02629 2.74333 ...
## $ Smile_6             : num 5.90e-04 2.60e-04 4.02e-01 1.10e-07 1.90e-02 ...
## $ Valence_6           : num -0.5799 0 0.0756 -0.939 -1.037 ...
## $ Attention_6         : num 85.8 92.6 82.3 84.2 82.6 ...
## $ Anger_6             : num 0.00516 0.01734 0.00228 0.0032 0.00306 ...
## $ Sadness_6           : num 0.01217 0.00304 0.02016 0.02758 0.01719 ...
## $ Disgust_6           : num 0.64 0.393 0.573 0.444 0.486 ...
## $ Joy_6               : num 0.0022 0.00138 0.35814 0.00181 0.0017 ...
## $ Surprise_6          : num 0.671 0.916 0.479 0.245 1.032 ...
## $ Fear_6              : num 0.28796 11.02816 0.00361 0.00388 0.00655 ...
## $ Contempt_6          : num 2.41 0.199 0.233 15.463 0.189 ...
## $ PeakCount_6         : int 3 0 9 10 2 7 0 10 0 8 ...
```

```
##  $ Peak.Min_6          : num  2.05 0 6.59 6.76 1.36 4.77 0 6.8 0 5.76 ...
##  $ AveAmplitude_6      : num  0.093 NA 0.048 0.0813 0.0552 ...
##  $ MaxAmplitude_6      : num  0.1185 NA 0.2916 0.2955 0.0552 ...
##  $ GSR_CAL_6           : num  2.41 1.78 5.09 3.07 1.17 ...
```

```
train <- sample (1:nrow(RatingP6H1NPSdata3), size=0.8*nrow(RatingP6H1NPSdata3)) # training row indices
RatingP6H1NPS_train <- RatingP6H1NPSdata3[train, ] # training data
RatingP6H1NPS_test <- RatingP6H1NPSdata3[-train, ] # test data


hist.default(RatingP6H1NPS_train$NPSID_6, main = "Histogram of Rating Scores in Train Data For P6",
        xlab = "Rating", ylab = "Frequency")
```

```
hist.default(RatingP6H1NPS_test$NPSID_6, main = "Histogram of Rating Scores in Test Data For P6",
        xlab = "Rating", ylab = "Frequency")
```

```
#Classification Tree
formula=NPSID_6 ~ InternetPerDay+OnlineShopPmont+PhotosVideosA+DescriptionsA+RatingsCommentsA+CostA+SportsPerWeek+SportsPerDay+CoachDrivenS+EBraSize+SportsBraSize+SportsBraUse+SportsbrasN+MoneyWishes+MoneyPractice+ProductVideos_6+RatingsComments_6+Descriptions_6+Cost_6+Confusion_6+Distraction_6+Enjoyment_6+Interest_6+Tiredness_6+Frustration_6 + PupilLeft_6 + PupilRight_6 + minFixationStart_6 + sumFixationDuration_6 + Classification_6 + HighEngagement_6 + LowEngagement_6 + Distraction_6.1 + Drowsy_6 + WorkloadFBDS_6 + WorkloadBDS_6 + WorkloadAverage_6 + BrowFurrow_6 + BrowRaise_6 + LipCornerDepressor_6 + Smile_6 + Valence_6 + Attention_6 + Anger_6 + Sadness_6 + Disgust_6 + Joy_6 + Surprise_6 + Fear_6 + Contempt_6 + PeakCount_6 + Peak.Min_6 + AveAmplitude_6 + MaxAmplitude_6 + GSR_CAL_6


RatingP6H1NPSdtree=rpart(formula,data=RatingP6H1NPS_train,method="class",control=rpart.control(minsplit=5,cp=0.001)) # build the model
```

```
#plot(dtree)
plot(RatingP6H1NPSdtree, uniform=TRUE,
   main="Classification Tree For P6 Rating")
text(RatingP6H1NPSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
```

```
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RatingP6H1NPSdtree)
##
## Classification tree:
## rpart(formula = formula, data = RatingP6H1NPS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Descriptions_6 Interest_6
##
## Root node error: 6/37 = 0.16216
##
## n= 37
##
##         CP nsplit rel error xerror    xstd
## 1 0.33333      0   1.00000 1.0000 0.37368
## 2 0.00100      2   0.33333 1.3333 0.41734
plotcp(RatingP6H1NPSdtree)
```

```
#summary(RatingP6H1NPSdtree)
RatingP6H1NPSdtree
## n= 37
##
```

```
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 37 6 1 (0.83783784 0.13513514 0.02702703)
##   2) Interest_6< 7.5 30 1 1 (0.96666667 0.03333333 0.00000000) *
##   3) Interest_6>=7.5 7 3 2 (0.28571429 0.57142857 0.14285714)
##     6) Descriptions_6>=6.5 3 1 1 (0.66666667 0.00000000 0.33333333) *
##     7) Descriptions_6< 6.5 4 0 2 (0.00000000 1.00000000 0.00000000) *
RatingP6H1NPS_train$NPSID_6
##  [1] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 3 2 2
## [36] 1 2
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RatingP6H1NPSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
##  [1] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2
## [36] 1 1
length(train_response_predicted)
## [1] 37
train_response_actual<- RatingP6H1NPS_train$NPSID_6 # actuals
train_response_actual
##  [1] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 3 2 2
## [36] 1 2
length(train_response_actual)
## [1] 37
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9459459
mean (train_response_actual != train_response_predicted) # Misclassification %
```

```
## [1] 0.05405405
```

```
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Rating Scores For P6 3 Grouped",
    xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RatingP6H1NPS_train$NPSID_6, train_response_predicted)
print(confusion.matrix)
```

```
##    train_response_predicted
##      1  2
##   1 31  0
##   2  1  4
##   3  1  0
```

```
#create a Actual vs Predicted Table from Decision Tree
(RatingP6H1NPSdelta<-as.data.frame(cbind(train_response_actual,train_response_predicted)))
```

```
##    train_response_actual train_response_predicted
## 1                      1                        1
## 2                      1                        1
## 3                      1                        1
## 4                      1                        1
## 5                      1                        1
## 6                      1                        1
## 7                      1                        1
## 8                      2                        2
## 9                      1                        1
```

```
## 10 1 1
## 11 1 1
## 12 1 1
## 13 1 1
## 14 1 1
## 15 1 1
## 16 1 1
## 17 1 1
## 18 1 1
## 19 1 1
## 20 1 1
## 21 1 1
## 22 1 1
## 23 1 1
## 24 1 1
## 25 1 1
## 26 2 2
## 27 1 1
## 28 1 1
## 29 1 1
## 30 1 1
## 31 1 1
## 32 1 1
## 33 3 1
## 34 2 2
## 35 2 2
## 36 1 1
## 37 2 1
```

```
names(RatingP6H1NPSdelta)<-c("RatingP6H1NPSactual","RatingP6H1NPSpredicted")
head(RatingP6H1NPSdelta)
```

```
##   RatingP6H1NPSactual RatingP6H1NPSpredicted
## 1                  1                      1
## 2                  1                      1
## 3                  1                      1
## 4                  1                      1
## 5                  1                      1
## 6                  1                      1
```

```
RatingP6H1NPSdelta$Match<-ifelse(RatingP6H1NPSdelta$RatingP6H1NPSactual==RatingP6H1NPSdelta$RatingP6H1NPSpredicted,1,0)
head(RatingP6H1NPSdelta)
```

```
##   RatingP6H1NPSactual RatingP6H1NPSpredicted Match
## 1                  1                      1     1
## 2                  1                      1     1
## 3                  1                      1     1
## 4                  1                      1     1
## 5                  1                      1     1
## 6                  1                      1     1
```

```
tail(RatingP6H1NPSdelta)
```

```
##    RatingP6H1NPSactual RatingP6H1NPSpredicted Match
## 32                   1                      1     1
## 33                   3                      1     0
## 34                   2                      2     1
## 35                   2                      2     1
## 36                   1                      1     1
## 37                   2                      1     0
```

```
write.csv(RatingP6H1NPSdelta,file="RatingP6H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P6/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
```

```
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9459459
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.05405405
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RatingP6H1NPSdtree, RatingP6H1NPS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
##  [1] 1 1 1 1 2 1 2 1 1 1
length(test_response_predicted)
## [1] 10
test_response_actual<- RatingP6H1NPS_test$NPSID_6 # actuals
test_response_actual
##  [1] 1 1 1 1 1 1 1 1 1 2
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.7
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.3
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Rating S
cores For P6 3 Grouped",
    xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
```

```
par(pty='s') #produces a square plot


#Obtain a confusion matrix
```

```
confusion.matrix <- table(RatingP6H1NPS_test$NPSID_6, test_response_predicted)
print(confusion.matrix)
```

```
##    test_response_predicted
##     1 2
##   1 7 2
##   2 1 0
```

```
#create a Actual vs Predicted Table from Decision Tree
(RatingP6H1NPSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted)))
```

```
##    test_response_actual test_response_predicted
## 1                     1                       1
## 2                     1                       1
## 3                     1                       1
## 4                     1                       1
## 5                     1                       2
## 6                     1                       1
## 7                     1                       2
## 8                     1                       1
## 9                     1                       1
## 10                    2                       1
```

```
names(RatingP6H1NPSdelta)<-c("RatingP6H1NPSactual","RatingP6H1NPSpredicted")
head(RatingP6H1NPSdelta)
```

```
##   RatingP6H1NPSactual RatingP6H1NPSpredicted
## 1                   1                      1
## 2                   1                      1
## 3                   1                      1
## 4                   1                      1
## 5                   1                      2
## 6                   1                      1
```

```
RatingP6H1NPSdelta$Match<-ifelse(RatingP6H1NPSdelta$RatingP6H1NPSactual==RatingP6H1NPSdelta$RatingP6H1NPSpredicted,1,0)
head(RatingP6H1NPSdelta)
```

```
##   RatingP6H1NPSactual RatingP6H1NPSpredicted Match
## 1                   1                      1     1
## 2                   1                      1     1
## 3                   1                      1     1
## 4                   1                      1     1
## 5                   1                      2     0
## 6                   1                      1     1
```

```
tail(RatingP6H1NPSdelta)
```

```
##    RatingP6H1NPSactual RatingP6H1NPSpredicted Match
## 5                    1                      2     0
## 6                    1                      1     1
## 7                    1                      2     0
## 8                    1                      1     1
## 9                    1                      1     1
## 10                   2                      1     0
```

```
write.csv(RatingP6H1NPSdelta,file="RatingP6H1NPSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/P6/H1NPS"
```

```
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
```

```
## [1] 0.7
```

```
mean (test_response_actual != test_response_predicted) # Misclassification %
```

```
## [1] 0.3
```

APPENDIX K

PREDICTION OF USER'S FIRST CHOICE IN LOW SUPPORT

RankingAB_LS.R
irfan
Sun Mar 18 23:34:11 2018

```
#UoB Ranking Analysis
#RankingAB_LS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_LS")

# Upload the data
RankingAB_LSdata <- read.csv("UoB.v14.csv", header=TRUE)

#Display header names
```

```r
#names(RankingAB_LSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RankingAB_LSdata2 <- RankingAB_LSdata[c(1,12,13,18:21,25,26,30:34,36,37,48,51:6
0,62,65:74,76,157:187,189:218)]
names(RankingAB_LSdata2)
```

```
##  [1] "Subject"             "InternetPerDay"
##  [3] "OnlineShopPmont"     "PhotosVideosA"
##  [5] "DescriptionsA"       "RatingsCommentsA"
##  [7] "CostA"               "SportsPerWeek"
##  [9] "SportsPerDay"        "CoachDrivenS"
## [11] "EBraSize"            "SportsBraSize"
## [13] "SportsBraUse"        "SportsbrasN"
## [15] "MoneyWishes"         "MoneyPractice"
## [17] "Rating_1"            "ProductVideos_1"
## [19] "RatingsComments_1"   "Descriptions_1"
## [21] "Cost_1"              "Confusion_1"
## [23] "Distraction_1"       "Enjoyment_1"
## [25] "Interest_1"          "Tiredness_1"
## [27] "Frustration_1"       "Rating_2"
## [29] "ProductVideos_2"     "RatingsComments_2"
## [31] "Descriptions_2"      "Cost_2"
## [33] "Confusion_2"         "Distraction_2"
## [35] "Enjoyment_2"         "Interest_2"
## [37] "Tiredness_2"         "Frustration_2"
## [39] "LS.FC"               "Product_1"
## [41] "PupilLeft_1"         "PupilRight_1"
## [43] "minFixationStart_1"  "sumFixationDuration_1"
## [45] "Classification_1"    "HighEngagement_1"
## [47] "LowEngagement_1"     "Distraction_1.1"
```

```
## [49] "Drowsy_1"             "WorkloadFBDS_1"
## [51] "WorkloadBDS_1"        "WorkloadAverage_1"
## [53] "BrowFurrow_1"         "BrowRaise_1"
## [55] "LipCornerDepressor_1" "Smile_1"
## [57] "Valence_1"            "Attention_1"
## [59] "Anger_1"              "Sadness_1"
## [61] "Disgust_1"            "Joy_1"
## [63] "Surprise_1"           "Fear_1"
## [65] "Contempt_1"           "PeakCount_1"
## [67] "Peak.Min_1"           "AveAmplitude_1"
## [69] "MaxAmplitude_1"       "GSR_CAL_1"
## [71] "PupilLeft_2"          "PupilRight_2"
## [73] "minFixationStart_2"   "sumFixationDuration_2"
## [75] "Classification_2"     "HighEngagement_2"
## [77] "LowEngagement_2"      "Distraction_2.1"
## [79] "Drowsy_2"             "WorkloadFBDS_2"
## [81] "WorkloadBDS_2"        "WorkloadAverage_2"
## [83] "BrowFurrow_2"         "BrowRaise_2"
## [85] "LipCornerDepressor_2" "Smile_2"
## [87] "Valence_2"            "Attention_2"
## [89] "Anger_2"              "Sadness_2"
## [91] "Disgust_2"            "Joy_2"
## [93] "Surprise_2"           "Fear_2"
## [95] "Contempt_2"           "PeakCount_2"
## [97] "Peak.Min_2"           "AveAmplitude_2"
## [99] "MaxAmplitude_2"       "GSR_CAL_2"
```

```
#If LS.FC have NAs, omit these rows that contain NA values
RankingAB_LSdata3 <- RankingAB_LSdata2[!is.na(RankingAB_LSdata2$PupilLeft_1),
]
#RankingAB_LSdata3 <- RankingAB_LSdata2[!is.na(RankingAB_LSdata2$LS.FC),]
```

```r
#str(RankingAB_LSdata3)


#Descriptives
describe(RankingAB_LSdata3$LS.FC)
## RankingAB_LSdata3$LS.FC
##        n  missing distinct     Info      Sum     Mean      Gmd
##       43        0        2    0.731       18   0.4186   0.4983
plot.default(RankingAB_LSdata3$LS.FC, main = "Ranking Scores For P1",
            xlab = "Index", ylab = "Ranking Scores")
hist.default(RankingAB_LSdata3$LS.FC, main = "Histogram of Ranking Scores For P1",
            xlab = "Ranking", ylab = "Frequency")
d <- density(RankingAB_LSdata3$LS.FC) # returns the density data
plot(d) # plots the results
#Create training and test data
#str(RankingAB_LSdata3)


train <- sample (1:nrow(RankingAB_LSdata3), size=0.8*nrow(RankingAB_LSdata3)) # training row indices
RankingAB_LS_train <- RankingAB_LSdata3[train, ] # training data
RankingAB_LS_test <- RankingAB_LSdata3[-train, ] # test data


hist.default(RankingAB_LS_train$LS.FC, main = "Histogram of Ranking Scores in Train Data For P1",
            xlab = "Ranking", ylab = "Frequency")
hist.default(RankingAB_LS_test$LS.FC, main = "Histogram of Ranking Scores in Test Data For P1",
            xlab = "Ranking", ylab = "Frequency")
#Classification Tree
formula=LS.FC ~ PupilLeft_1+Sadness_1+sumFixationDuration_1+Classification_2+Enjoyment_2+WorkloadBDS_2
```

```
RankingAB_LSdtree=rpart(formula,data=RankingAB_LS_train,method="class",control=rpart.control(minsplit=5,cp=0.001)) # build the model


#plot(dtree)
plot(RankingAB_LSdtree, uniform=TRUE,
   main="Classification Tree For P1 Ranking")
text(RankingAB_LSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number of observations at each node
# xpd = TRUE keeps the labels from exteding outside the plot
printcp(RankingAB_LSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RankingAB_LS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Classification_2 Sadness_1        WorkloadBDS_2
##
## Root node error: 14/34 = 0.41176
##
## n= 34
##
##         CP nsplit rel error  xerror    xstd
## 1 0.500000      0   1.00000 1.00000 0.20498
## 2 0.107143      1   0.50000 0.78571 0.19485
## 3 0.071429      3   0.28571 0.92857 0.20240
## 4 0.001000      4   0.21429 0.85714 0.19904
```

```
plotcp(RankingAB_LSdtree)
#summary(RankingAB_LSdtree)
RankingAB_LSdtree
```

```
## n= 34
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 34 14 0 (0.5882353 0.4117647)
##    2) Sadness_1>=0.008247115 21  4 0 (0.8095238 0.1904762)
##      4) Classification_2< 0.757926 11  0 0 (1.0000000 0.0000000) *
##      5) Classification_2>=0.757926 10  4 0 (0.6000000 0.4000000)
##       10) WorkloadBDS_2< 0.5376033 7  1 0 (0.8571429 0.1428571) *
##       11) WorkloadBDS_2>=0.5376033 3  0 1 (0.0000000 1.0000000) *
##    3) Sadness_1< 0.008247115 13  3 1 (0.2307692 0.7692308)
##      6) WorkloadBDS_2< 0.5060932 3  1 0 (0.6666667 0.3333333) *
##      7) WorkloadBDS_2>=0.5060932 10  1 1 (0.1000000 0.9000000) *
```

```
RankingAB_LS_train$LS.FC
```

```
##  [1] 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1
```

```
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RankingAB_LSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
```

```
##  [1] 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1
```

```
length(train_response_predicted)
```

```
## [1] 34
```

```
train_response_actual<- RankingAB_LS_train$LS.FC # actuals
train_response_actual
```

```
##  [1] 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1
```

```
length(train_response_actual)
```

```
## [1] 34
```

```
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9117647
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.08823529
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Ranki
ng Scores For P1 3 Grouped",
    xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RankingAB_LS_train$LS.FC, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      0  1
##   0 19  1
##   1  2 12
#create a Actual vs Predicted Table from Decision Tree
#(RankingAB_LSdelta<-as.data.frame(cbind(train_response_actual,train_response_pred
icted)))
#names(RankingAB_LSdelta)<-c("RankingAB_LSactual","RankingAB_LSpredicted")
#head(RankingAB_LSdelta)
#RankingAB_LSdelta$Match<-ifelse(RankingAB_LSdelta$RankingAB_LSactual==Rank
ingAB_LSdelta$RankingAB_LSpredicted,1,0)
#head(RankingAB_LSdelta)
#tail(RankingAB_LSdelta)
#write.csv(RankingAB_LSdelta,file="RankingAB_LSdelta.csv",row.names=TRUE)
#getwd()


#Summary of Accuracy
```

```
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.9117647
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.08823529
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RankingAB_LSdtree, RankingAB_LS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
## [1] 0 0 0 0 1 0 0 1 1
length(test_response_predicted)
## [1] 9
test_response_actual<- RankingAB_LS_test$LS.FC # actuals
test_response_actual
## [1] 0 1 1 1 0 0 1 0 0
length(test_response_actual)
## [1] 9
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.2222222
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.7777778
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Ranking
Scores For Support Levels",
    xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
par(pty='s') #produces a square plot


#Obtain a confusion matrix
```

```
confusion.matrix <- table(RankingAB_LS_test$LS.FC, test_response_predicted)
print(confusion.matrix)
##    test_response_predicted
##     0 1
##   0 2 3
##   1 4 0
#create a Actual vs Predicted Table from Decision Tree
(RankingAB_LSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted
)))
##   test_response_actual test_response_predicted
## 1                    0                       0
## 2                    1                       0
## 3                    1                       0
## 4                    1                       0
## 5                    0                       1
## 6                    0                       0
## 7                    1                       0
## 8                    0                       1
## 9                    0                       1
names(RankingAB_LSdelta)<-c("RankingAB_LSactual","RankingAB_LSpredicted")
#head(RankingAB_LSdelta)
RankingAB_LSdelta$Match<-ifelse(RankingAB_LSdelta$RankingAB_LSactual==Rank
ingAB_LSdelta$RankingAB_LSpredicted,1,0)
#head(RankingAB_LSdelta)
#tail(RankingAB_LSdelta)
write.csv(RankingAB_LSdelta,file="RankingAB_LSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_LS"
#Summary of Accuracy
#Decision Tree Model
```

```
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.2222222
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.7777778
```

APPENDIX L

PREDICTION OF USER'S FIRST CHOICE IN MEDIUM SUPPORT

RankingAB_MS.R
irfan
Mon Mar 19 00:20:45 2018

```
#UoB Ranking Analysis
#RankingAB_MS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_MS")

# Upload the data
RankingAB_MSdata <- read.csv("UoB.v14.csv", header=TRUE)

#Display header names
```

```r
#names(RankingAB_MSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RankingAB_MSdata2 <- RankingAB_MSdata[c(1,12,13,18:21,25,26,30:34,36,37,80,83:
92,94,97:106,108,220:249,251:280)]
names(RankingAB_MSdata2)
```

```
##  [1] "Subject"              "InternetPerDay"
##  [3] "OnlineShopPmont"      "PhotosVideosA"
##  [5] "DescriptionsA"        "RatingsCommentsA"
##  [7] "CostA"                "SportsPerWeek"
##  [9] "SportsPerDay"         "CoachDrivenS"
## [11] "EBraSize"             "SportsBraSize"
## [13] "SportsBraUse"         "SportsbrasN"
## [15] "MoneyWishes"          "MoneyPractice"
## [17] "Rating_3"             "ProductVideos_3"
## [19] "RatingsComments_3"    "Descriptions_3"
## [21] "Cost_3"               "Confusion_3"
## [23] "Distraction_3"        "Enjoyment_3"
## [25] "Interest_3"           "Tiredness_3"
## [27] "Frustration_3"        "Rating_4"
## [29] "ProductVideos_4"      "RatingsComments_4"
## [31] "Descriptions_4"       "Cost_4"
## [33] "Confusion_4"          "Distraction_4"
## [35] "Enjoyment_4"          "Interest_4"
## [37] "Tiredness_4"          "Frustration_4"
## [39] "MS.FC"                "PupilLeft_3"
## [41] "PupilRight_3"         "minFixationStart_3"
## [43] "sumFixationDuration_3" "Classification_3"
## [45] "HighEngagement_3"     "LowEngagement_3"
## [47] "Distraction_3.1"      "Drowsy_3"
```

```
## [49] "WorkloadFBDS_3"        "WorkloadBDS_3"
## [51] "WorkloadAverage_3"     "BrowFurrow_3"
## [53] "BrowRaise_3"           "LipCornerDepressor_3"
## [55] "Smile_3"               "Valence_3"
## [57] "Attention_3"           "Anger_3"
## [59] "Sadness_3"             "Disgust_3"
## [61] "Joy_3"                 "Surprise_3"
## [63] "Fear_3"                "Contempt_3"
## [65] "PeakCount_3"           "Peak.Min_3"
## [67] "AveAmplitude_3"        "MaxAmplitude_3"
## [69] "GSR_CAL_3"             "PupilLeft_4"
## [71] "PupilRight_4"          "minFixationStart_4"
## [73] "sumFixationDuration_4" "Classification_4"
## [75] "HighEngagement_4"      "LowEngagement_4"
## [77] "Distraction_4.1"       "Drowsy_4"
## [79] "WorkloadFBDS_4"        "WorkloadBDS_4"
## [81] "WorkloadAverage_4"     "BrowFurrow_4"
## [83] "BrowRaise_4"           "LipCornerDepressor_4"
## [85] "Smile_4"               "Valence_4"
## [87] "Attention_4"           "Anger_4"
## [89] "Sadness_4"             "Disgust_4"
## [91] "Joy_4"                 "Surprise_4"
## [93] "Fear_4"                "Contempt_4"
## [95] "PeakCount_4"           "Peak.Min_4"
## [97] "AveAmplitude_4"        "MaxAmplitude_4"
## [99] "GSR_CAL_4"
```

```
#If MS.FC have NAs, omit these rows that contain NA values
#RankingAB_MSdata3 <- RankingAB_MSdata2[!is.na(RankingAB_MSdata2$PupilLeft_1),]
RankingAB_MSdata3 <- RankingAB_MSdata2[!is.na(RankingAB_MSdata2$MS.FC),]
```

```
#str(RankingAB_MSdata3)


#Descriptives
describe(RankingAB_MSdata3$MS.FC)
## RankingAB_MSdata3$MS.FC
##        n  missing distinct     Info      Sum     Mean      Gmd
##       48        0        2    0.667       32   0.6667   0.4539
plot.default(RankingAB_MSdata3$MS.FC, main = "Ranking Scores For P1",
             xlab = "Index", ylab = "Ranking Scores")
hist.default(RankingAB_MSdata3$MS.FC, main = "Histogram of Ranking Scores For P1
",
             xlab = "Ranking", ylab = "Frequency")
d <- density(RankingAB_MSdata3$MS.FC) # returns the density data
plot(d) # plots the results
#Create training and test data
#str(RankingAB_MSdata3)


train <- sample (1:nrow(RankingAB_MSdata3), size=0.8*nrow(RankingAB_MSdata3))
# training row indices
RankingAB_MS_train <- RankingAB_MSdata3[train, ] # training data
RankingAB_MS_test <- RankingAB_MSdata3[-train, ] # test data


hist.default(RankingAB_MS_train$MS.FC, main = "Histogram of Ranking Scores in Tra
in Data For P1",
             xlab = "Ranking", ylab = "Frequency")
hist.default(RankingAB_MS_test$MS.FC, main = "Histogram of Ranking Scores in Test
Data For P1",
             xlab = "Ranking", ylab = "Frequency")
#Classification Tree
formula=MS.FC ~ GSR_CAL_3+Interest_3+LipCornerDepressor_3+MoneyPractice+Sp
ortsPerDay+Valence_3+Drowsy_4+Enjoyment_4+Surprise_4
```

```
RankingAB_MSdtree=rpart(formula,data=RankingAB_MS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model


#plot(dtree)
plot(RankingAB_MSdtree, uniform=TRUE,
    main="Classification Tree For P1 Ranking")
text(RankingAB_MSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number o
f observations at each node
# xpd = TRUE keeps the labeMS from exteding outside the plot
printcp(RankingAB_MSdtree)
```

```
##
## Classification tree:
## rpart(formula = formula, data = RankingAB_MS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Enjoyment_4 Interest_3
##
## Root node error: 13/38 = 0.34211
##
## n= 38
##
##        CP nsplit rel error  xerror    xstd
## 1 0.34615      0   1.00000 1.00000 0.22496
## 2 0.00100      2   0.30769 0.38462 0.16029
```

```
plotcp(RankingAB_MSdtree)
#summary(RankingAB_MSdtree)
RankingAB_MSdtree
```

```
## n= 38
```

```
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 38 13 1 (0.34210526 0.65789474)
##   2) Interest_3< 5.5 18  7 0 (0.61111111 0.38888889)
##     4) Enjoyment_4>=4 11  1 0 (0.90909091 0.09090909) *
##     5) Enjoyment_4< 4 7  1 1 (0.14285714 0.85714286) *
##   3) Interest_3>=5.5 20  2 1 (0.10000000 0.90000000) *
RankingAB_MS_train$MS.FC
##  [1] 1 0 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0
## [36] 1 0 1
#Predict on fitted data and calculate misclassification percentage
#Model Response in Train Data
train_Out<-predict(RankingAB_MSdtree)
train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted
train_response_predicted
##  [1] 1 0 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0
## [36] 1 0 1
length(train_response_predicted)
## [1] 38
train_response_actual<- RankingAB_MS_train$MS.FC # actuaMS
train_response_actual
##  [1] 1 0 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0
## [36] 1 0 1
length(train_response_actual)
## [1] 38
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.8947368
```

```
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.1052632
plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Ranki
ng Scores For P1 3 Grouped",
     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")
lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it
par(pty='s') #produces a square plot


#Obtain a confusion matrix
confusion.matrix <- table(RankingAB_MS_train$MS.FC, train_response_predicted)
print(confusion.matrix)
##    train_response_predicted
##      0  1
##   0 10  3
##   1  1 24
#create a Actual vs Predicted Table from Decision Tree
#(RankingAB_MSdelta<-as.data.frame(cbind(train_response_actual,train_response_pre
dicted)))
#names(RankingAB_MSdelta)<-c("RankingAB_MSactual","RankingAB_MSpredicted")
#head(RankingAB_MSdelta)
#RankingAB_MSdelta$Match<-ifelse(RankingAB_MSdelta$RankingAB_MSactual==Ra
nkingAB_MSdelta$RankingAB_MSpredicted,1,0)
#head(RankingAB_MSdelta)
#tail(RankingAB_MSdelta)
#write.csv(RankingAB_MSdelta,file="RankingAB_MSdelta.csv",row.names=TRUE)
#getwd()


#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
```

```
## [1] 0.8947368
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.1052632
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RankingAB_MSdtree, RankingAB_MS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method
= c("first"))] )# predicted
test_response_predicted
##  [1] 1 1 1 1 1 1 1 0 1 1
length(test_response_predicted)
## [1] 10
test_response_actual<- RankingAB_MS_test$MS.FC # actuaMS
test_response_actual
##  [1] 0 1 1 1 1 0 0 1 1 1
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Ranking
Scores For Support LeveMS",
    xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
par(pty='s') #produces a square plot

#Obtain a confusion matrix
confusion.matrix <- table(RankingAB_MS_test$MS.FC, test_response_predicted)
print(confusion.matrix)
```

```
##    test_response_predicted
##     0 1
##   0 0 3
##   1 1 6
```

```r
#create a Actual vs Predicted Table from Decision Tree
(RankingAB_MSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted)))
```

```
##    test_response_actual test_response_predicted
## 1                     0                       1
## 2                     1                       1
## 3                     1                       1
## 4                     1                       1
## 5                     1                       1
## 6                     0                       1
## 7                     0                       1
## 8                     1                       0
## 9                     1                       1
## 10                    1                       1
```

```r
names(RankingAB_MSdelta)<-c("RankingAB_MSactual","RankingAB_MSpredicted")
#head(RankingAB_MSdelta)
RankingAB_MSdelta$Match<-ifelse(RankingAB_MSdelta$RankingAB_MSactual==RankingAB_MSdelta$RankingAB_MSpredicted,1,0)
#head(RankingAB_MSdelta)
#tail(RankingAB_MSdelta)
write.csv(RankingAB_MSdelta,file="RankingAB_MSdelta.csv",row.names=TRUE)
getwd()
```

```
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_MS"
```

```r
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
```

```
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
```

APPENDIX M

PREDICTION OF USER'S FIRST CHOICE IN HIGH SUPPORT

RankingAB_HS.R
irfan
Mon Mar 19 00:16:04 2018

```
#UoB Ranking Analysis
#RankingAB_HS-ALL
# Prediction error rate in training data = Root node error * rel error * 100%
# Prediction error rate in cross-validation = Root node error * xerror * 100%

# load the package
library(rpart)
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# remove existing variables
rm(list = ls(all = TRUE))

# Set workplace
setwd("/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_HS")

# Upload the data
RankingAB_HSdata <- read.csv("UoB.v14.csv", header=TRUE)

#Display header names
```

```r
#names(RankingAB_HSdata)


# Create a new dataset with only the variables we want to use in our Decision Tree
RankingAB_HSdata2 <- RankingAB_HSdata[c(1,12,13,18:21,25,26,30:34,36,37,112,115:124,126,129:138,140,282:311,313:342)]
names(RankingAB_HSdata2)
```

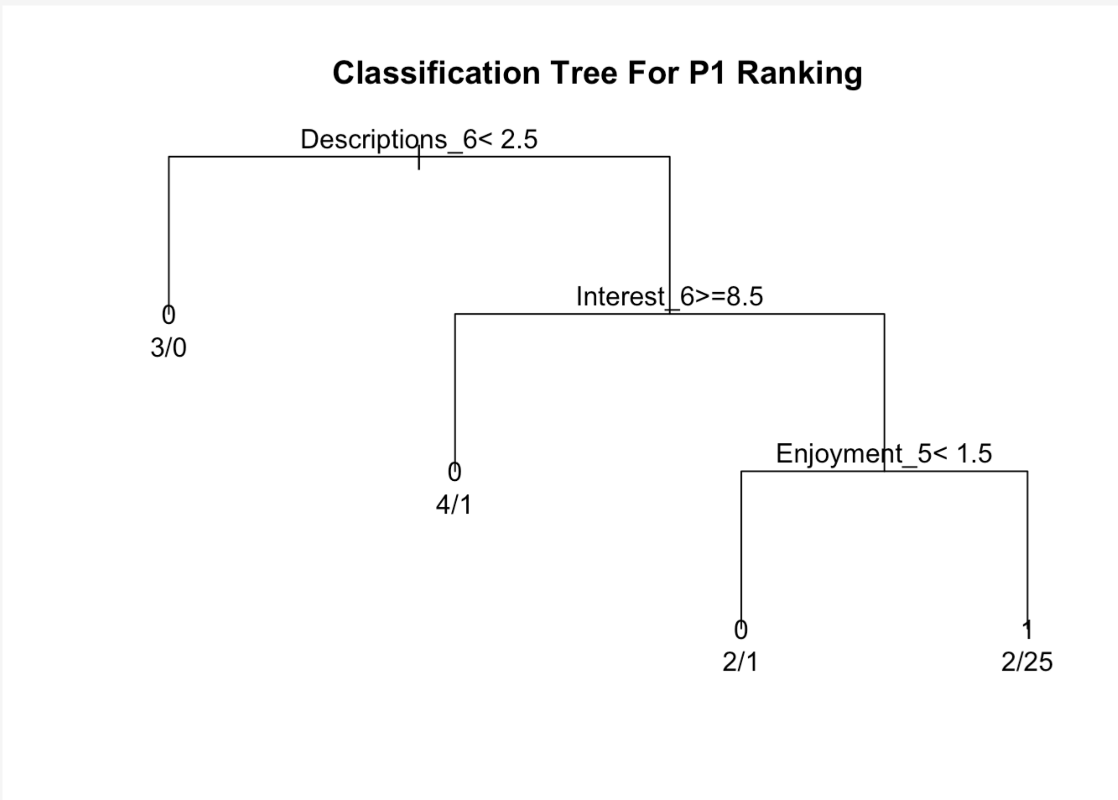
```
##  [1] "Subject"             "InternetPerDay"
##  [3] "OnlineShopPmont"     "PhotosVideosA"
##  [5] "DescriptionsA"       "RatingsCommentsA"
##  [7] "CostA"               "SportsPerWeek"
##  [9] "SportsPerDay"        "CoachDrivenS"
## [11] "EBraSize"            "SportsBraSize"
## [13] "SportsBraUse"        "SportsbrasN"
## [15] "MoneyWishes"         "MoneyPractice"
## [17] "Rating_5"            "ProductVideos_5"
## [19] "RatingsComments_5"   "Descriptions_5"
## [21] "Cost_5"              "Confusion_5"
## [23] "Distraction_5"       "Enjoyment_5"
## [25] "Interest_5"          "Tiredness_5"
## [27] "Frustration_5"       "Rating_6"
## [29] "ProductVideos_6"     "RatingsComments_6"
## [31] "Descriptions_6"      "Cost_6"
## [33] "Confusion_6"         "Distraction_6"
## [35] "Enjoyment_6"         "Interest_6"
## [37] "Tiredness_6"         "Frustration_6"
## [39] "HS.FC"               "PupilLeft_5"
## [41] "PupilRight_5"        "minFixationStart_5"
## [43] "sumFixationDuration_5" "Classification_5"
## [45] "HighEngagement_5"    "LowEngagement_5"
## [47] "Distraction_5.1"     "Drowsy_5"
```

```
## [49] "WorkloadFBDS_5"       "WorkloadBDS_5"
## [51] "WorkloadAverage_5"    "BrowFurrow_5"
## [53] "BrowRaise_5"          "LipCornerDepressor_5"
## [55] "Smile_5"              "Valence_5"
## [57] "Attention_5"          "Anger_5"
## [59] "Sadness_5"            "Disgust_5"
## [61] "Joy_5"                "Surprise_5"
## [63] "Fear_5"               "Contempt_5"
## [65] "PeakCount_5"          "Peak.Min_5"
## [67] "AveAmplitude_5"       "MaxAmplitude_5"
## [69] "GSR_CAL_5"            "PupilLeft_6"
## [71] "PupilRight_6"         "minFixationStart_6"
## [73] "sumFixationDuration_6" "Classification_6"
## [75] "HighEngagement_6"     "LowEngagement_6"
## [77] "Distraction_6.1"      "Drowsy_6"
## [79] "WorkloadFBDS_6"       "WorkloadBDS_6"
## [81] "WorkloadAverage_6"    "BrowFurrow_6"
## [83] "BrowRaise_6"          "LipCornerDepressor_6"
## [85] "Smile_6"              "Valence_6"
## [87] "Attention_6"          "Anger_6"
## [89] "Sadness_6"            "Disgust_6"
## [91] "Joy_6"                "Surprise_6"
## [93] "Fear_6"               "Contempt_6"
## [95] "PeakCount_6"          "Peak.Min_6"
## [97] "AveAmplitude_6"       "MaxAmplitude_6"
## [99] "GSR_CAL_6"
```

```
#If HS.FC have NAs, omit these rows that contain NA values
#RankingAB_HSdata3 <- RankingAB_HSdata2[!is.na(RankingAB_HSdata2$PupilLeft_1
),]
RankingAB_HSdata3 <- RankingAB_HSdata2[!is.na(RankingAB_HSdata2$HS.FC),]
```

```
#str(RankingAB_HSdata3)


#Descriptives
describe(RankingAB_HSdata3$HS.FC)
## RankingAB_HSdata3$HS.FC
##        n  missing distinct     Info      Sum     Mean      Gmd
##       48        0        2    0.667       32   0.6667   0.4539
plot.default(RankingAB_HSdata3$HS.FC, main = "Ranking Scores For P1",
             xlab = "Index", ylab = "Ranking Scores")
hist.default(RankingAB_HSdata3$HS.FC, main = "Histogram of Ranking Scores For P1"
,
             xlab = "Ranking", ylab = "Frequency")
d <- density(RankingAB_HSdata3$HS.FC) # returns the density data
plot(d) # plots the results
#Create training and test data
#str(RankingAB_HSdata3)


train <- sample (1:nrow(RankingAB_HSdata3), size=0.8*nrow(RankingAB_HSdata3)) #
training row indices
RankingAB_HS_train <- RankingAB_HSdata3[train, ] # training data
RankingAB_HS_test <- RankingAB_HSdata3[-train, ] # test data


hist.default(RankingAB_HS_train$HS.FC, main = "Histogram of Ranking Scores in Trai
n Data For P1",
             xlab = "Ranking", ylab = "Frequency")
hist.default(RankingAB_HS_test$HS.FC, main = "Histogram of Ranking Scores in Test
Data For P1",
             xlab = "Ranking", ylab = "Frequency")
#Classification Tree
formula=HS.FC ~ CoachDrivenS+Enjoyment_5+OnlineShopPmont+PhotosVideosA+Pu
pilLeft_5+SportsPerWeek+Descriptions_6+Interest_6
```

```
RankingAB_HSdtree=rpart(formula,data=RankingAB_HS_train,method="class",control
=rpart.control(minsplit=5,cp=0.001)) # build the model


#plot(dtree)
plot(RankingAB_HSdtree, uniform=TRUE,
   main="Classification Tree For P1 Ranking")
text(RankingAB_HSdtree, use.n = TRUE, xpd = TRUE) # use.n = TRUE adds number of
observations at each node
```




```
# xpd = TRUE keeps the labeMS from exteding outside the plot
printcp(RankingAB_HSdtree)
##
## Classification tree:
## rpart(formula = formula, data = RankingAB_HS_train, method = "class",
##     control = rpart.control(minsplit = 5, cp = 0.001))
##
```

```
## Variables actually used in tree construction:
## [1] Descriptions_6 Enjoyment_5    Interest_6
##
## Root node error: 11/38 = 0.28947
##
## n= 38
##
##         CP nsplit rel error  xerror    xstd
## 1 0.272727      0   1.00000 1.00000 0.25415
## 2 0.090909      2   0.45455 0.81818 0.23825
## 3 0.001000      3   0.36364 0.90909 0.24677
```

```
plotcp(RankingAB_HSdtree)
#summary(RankingAB_HSdtree)
RankingAB_HSdtree
```

```
## n= 38
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 38 11 1 (0.28947368 0.71052632)
##    2) Descriptions_6< 2.5 3  0 0 (1.00000000 0.00000000) *
##    3) Descriptions_6>=2.5 35  8 1 (0.22857143 0.77142857)
##      6) Interest_6>=8.5 5  1 0 (0.80000000 0.20000000) *
##      7) Interest_6< 8.5 30  4 1 (0.13333333 0.86666667)
##       14) Enjoyment_5< 1.5 3  1 0 (0.66666667 0.33333333) *
##       15) Enjoyment_5>=1.5 27  2 1 (0.07407407 0.92592593) *
```

```
RankingAB_HS_train$HS.FC
```

```
##  [1] 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 1 0 1
## [36] 1 1 0
```

```
#Predict on fitted data and calculate misclassification percentage
```

```
#Model Response in Train Data

train_Out<-predict(RankingAB_HSdtree)

train_response_predicted<- as.numeric(colnames(train_Out)[max.col(train_Out, ties.meth
od = c("first"))] )# predicted

train_response_predicted

##  [1] 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1

## [36] 1 1 0

length(train_response_predicted)

## [1] 38

train_response_actual<- RankingAB_HS_train$HS.FC # actuaMS

train_response_actual

##  [1] 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 1 0 1

## [36] 1 1 0

length(train_response_actual)

## [1] 38

mean (train_response_actual == train_response_predicted) # Accuracy %

## [1] 0.8947368

mean (train_response_actual != train_response_predicted) # Misclassification %

## [1] 0.1052632

plot(train_response_actual, train_response_predicted, main = "Actual vs Predicted Ranki
ng Scores For P1 3 Grouped",

     xlab = "Actual Scores in Train Data", ylab = "Predicted Scores in Test Data")

lines(lowess(train_response_actual, train_response_predicted), col = "blue") # Add loess f
it

par(pty='s') #produces a square plot


#Obtain a confusion matrix

confusion.matrix <- table(RankingAB_HS_train$HS.FC, train_response_predicted)

print(confusion.matrix)

##    train_response_predicted

##      0  1
```

```
##   0  9  2
##   1  2 25
#create a Actual vs Predicted Table from Decision Tree
#(RankingAB_HSdelta<-as.data.frame(cbind(train_response_actual,train_response_predicted)))
#names(RankingAB_HSdelta)<-c("RankingAB_HSactual","RankingAB_HSpredicted")
#head(RankingAB_HSdelta)
#RankingAB_HSdelta$Match<-ifelse(RankingAB_HSdelta$RankingAB_HSactual==RankingAB_HSdelta$RankingAB_HSpredicted,1,0)
#head(RankingAB_HSdelta)
#tail(RankingAB_HSdelta)
#write.csv(RankingAB_HSdelta,file="RankingAB_HSdelta.csv",row.names=TRUE)
#getwd()


#Summary of Accuracy
#Decision Tree Model
mean (train_response_actual == train_response_predicted) # Accuracy %
## [1] 0.8947368
mean (train_response_actual != train_response_predicted) # Misclassification %
## [1] 0.1052632
#Predict the Test data and calculate misclassification percentage
#Model Response in Test Data
test_Out<-predict(RankingAB_HSdtree, RankingAB_HS_test)
test_response_predicted<- as.numeric(colnames(test_Out)[max.col(test_Out, ties.method = c("first"))] )# predicted
test_response_predicted
##  [1] 1 0 1 1 1 0 1 1 1 0
length(test_response_predicted)
## [1] 10
test_response_actual<- RankingAB_HS_test$HS.FC # actuaMS
test_response_actual
```

```
##  [1] 1 1 1 1 1 0 0 0 0 0
length(test_response_actual)
## [1] 10
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
plot(test_response_actual, test_response_predicted, main = "Actual vs Predicted Ranking Scores For Support LeveMS",
     xlab = "Actual Scores in Test Data", ylab = "Predicted Scores in Test")
lines(lowess(test_response_actual, test_response_predicted), col = "blue") # Add loess fit
par(pty='s') #produces a square plot

#Obtain a confusion matrix
confusion.matrix <- table(RankingAB_HS_test$HS.FC, test_response_predicted)
print(confusion.matrix)
##    test_response_predicted
##     0 1
##   0 2 3
##   1 1 4
#create a Actual vs Predicted Table from Decision Tree
(RankingAB_HSdelta<-as.data.frame(cbind(test_response_actual,test_response_predicted)))
##    test_response_actual test_response_predicted
## 1                    1                       1
## 2                    1                       0
## 3                    1                       1
## 4                    1                       1
## 5                    1                       1
## 6                    0                       0
```

```
## 7                0                1
## 8                0                1
## 9                0                1
## 10               0                0
names(RankingAB_HSdelta)<-c("RankingAB_HSactual","RankingAB_HSpredicted")
#head(RankingAB_HSdelta)
RankingAB_HSdelta$Match<-ifelse(RankingAB_HSdelta$RankingAB_HSactual==RankingAB_HSdelta$RankingAB_HSpredicted,1,0)
#head(RankingAB_HSdelta)
#tail(RankingAB_HSdelta)
write.csv(RankingAB_HSdelta,file="RankingAB_HSdelta.csv",row.names=TRUE)
getwd()
## [1] "/Volumes/GoogleDrive/My Drive/Dissertation/Analysis/O3/Ranking/AB_HS"
#Summary of Accuracy
#Decision Tree Model
mean (test_response_actual == test_response_predicted) # Accuracy %
## [1] 0.6
mean (test_response_actual != test_response_predicted) # Misclassification %
## [1] 0.4
```

BIOGRAPHICAL SKETCH

Irfan Kula is a Ph.D. candidate at Human Systems Engineering Program of Ira A. Fulton Schools of Engineering at Arizona State University. He received his M.Ed in Educational Technology from Arizona State University in 2014, and he has 10 years instructional design and usability experience. Irfan is a certified researcher on biometric human behavior research. His research interests include user experience, user-centered design, affective computing, multiple affect recognition, customer decision making processes. In particular, he specializes in the impact of multi-sensor affect recognition on user experience evaluation systems.