A Major Qualifying Project
submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree in Bachelor of Science
in
Computer Science
By

Joshua Audibert
Elijah Gonzalez
Ryan Orlando
Nicholas Wong

_____

Professor Emmanuel Agu, Co-advisor
Professor Mark Claypool, Co-advisor

Table of Contents

1 Introduction

Physical inactivity is one of the leading causes of death in the United States, which increases the risk of many ailments including diabetes, cardiovascular disease, metabolic syndrome and some cancers (Mokdad, 2000). Obesity is the leading preventable cause of health problems afflicting children and young adults. Over 78 million U.S. adults and about 12.5 million U.S. children and adolescents were considered obese, according to a National Health and Nutrition Examination Survey administered in 2009-2010 (Holland, 2016). The core of the problem begins with overweight adolescents, who have a 70% chance of becoming overweight adults. Only one in three children are physically active each day due to the growing online lifestyle trending in adolescents and the lack of enjoyable physical activity alternatives besides organized sports (Agu & Claypool, 2016). A viable exercise alternative to engage adolescents is pervasive games, which extend the gaming experience into the physical realm. Exergames, such as Pokemon Go and Just Dance Now, are one particular form of pervasive game that shows great promise in appealing to adolescents.

Exergames are a type of pervasive game that incorporate gameplay elements into exercise to increase physical activity in an enjoyable way (Agu & Claypool, 2016). Playing exergames promotes good health, increases aerobic fitness, and improves metabolic and physiological variables in adolescents (Wang & Perry, 2006). They have also been proven to serve as a sufficient alternative to regular exercise (Kretschmann, 2010). However, 95% of all new game players stop playing within 3 months, and 85% of new players stop after just one day, a trend that encompasses all genres of games including exergames (Agu & Claypool, 2016). If exergames are to be effective, a system must be implemented that connects users to new exergames that they will enjoy when they lose interest in their current game.

Currently, there are multiple problems with the process of connecting adolescents to exergames they would enjoy (Agu & Claypool, 2016). These problems are:

1. It may take many attempts to find an appealing exergame.
2. Players need to actively seek new exergames when they get bored.
3. Game recommendations for are primarily influenced by popularity, making it difficult to find tailored to individual interest.
4. Measures of user enjoyment are not specific to exergames.
5. Feedback on user enjoyment of exergames is currently limited mostly to sparse user reviews on websites such as Amazon.com.

The proposed Cyber-Physical Recommender System (CyPRESS) aims to solve these problems. Figure 1.1 illustrates an overview of the CyPRESS system at a high level.

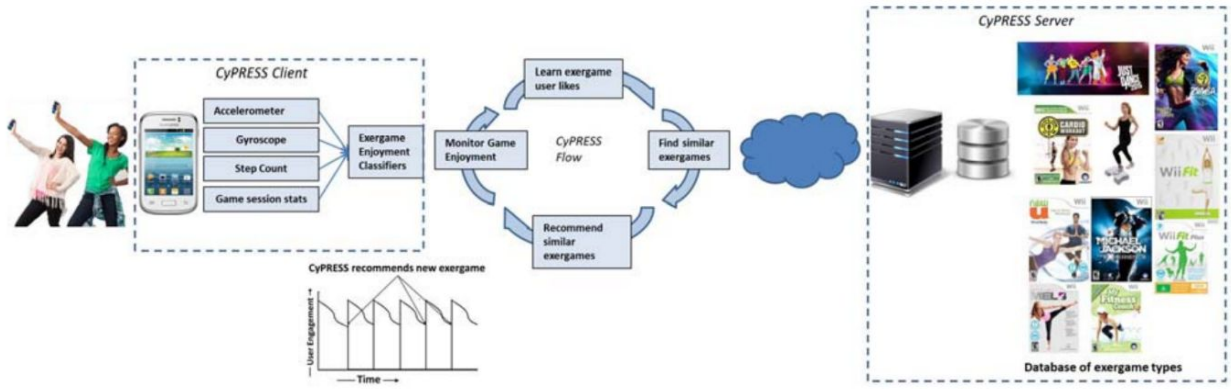

Figure 1.1: CyPRESS Overview (Agu, Claypool 2016)

CyPRESS will be a recommendation system which tries to connect players to exercise games that they may enjoy. By continuously monitoring player's motion data, step count and game play statistics from their phone, CyPRESS can determine whether or not an individual is enjoying a game. CyPRESS will then recommend new exergames whenever the player is bored. The research and development of CyPRESS involves the following five steps:

1. Selection of exergames for experiments

2. Smartphone instrumentation and generation of sensor data gathering app
3. Adaption of user game experience questionnaires
4. Synthesizing implicit interest indicators
5. Use predicted Enjoyment-scores (E-scores) to recommend new games.

On the user side is the CyPRESS smartphone client, which captures data from phone sensors and runs it through machine learning classifiers that have been trained on previous data to determine whether or not they have enjoyed the current experience. The system uses this information to determine which kinds of games the user enjoys, and when the user loses interest in the current game, CyPRESS will recommend similar games to the user based on a large database of games stored on CyPRESS servers.

The scope of this Major Qualifying Project (MQP) was to determine whether it is possible to train machine learning classifiers to measure excitement in exergames from accelerometer and gyroscope data from the gamer's smartphone. Our work roughly covered the first, second and fourth CyPRESS research agenda criteria above. The third step of adapting user questionnaire was addressed by a separate IQP group. Their group created a novel Exergame Enjoyment Questionnaire (EEQ) that was administered after a gamer played an exergame. The EEQ measured how much they enjoyed the game. We utilized EEQ for measuring exergame player enjoyment of games evaluated during our experiments.

The goal of this MQP was to determine whether or not it is possible to determine an individual's enjoyment from an exercise game based on motion data. The approach to meeting this goal was accomplished in the follow steps:

- Run experiments using the Pokémon Go and Just Dance Now exergames while subjects play them, in order to collect data
- Pre-process the raw data and calculate features from it

- Use machine learning on the processed data and extracted features to find trends in the data
- Determine whether or not it is possible to detect enjoyment from motion data by analyzing the output of the machine learning algorithms
- Synthesize classifiers and evaluate how accurately they classify reported player enjoyment scores from their smartphone sensor data gathered as they played

Through the use of machine learning classifiers we were able to create a classification model that was reasonably successful at predicting user enjoyment. Our final model was able to correctly predict enjoyment 75% of the time.

For the rest of our report we will explain how we ran our experiments and created our model, the results of these experiments, and the conclusions we drew from these experiments.

2 Background

Before attempting to achieve our goals regarding CyPRESS, it was important to review background information related to our work, and that we attempted to understand the different facets of a recommendation system such as CyPRESS. To accomplish this we researched the various genres of exercise games. We also looked into different methods of quantifying game experiences, different types of phone sensors, and learned the basics of machine learning.

2.1 Mobile Exergames

Before further discussing the details of our project, it is important to have a thorough understanding of what exergames are, as well as the different genres that are available. Exergames are electronic games that involve exercise, and thus offer a new way to work out (Koivisto, Sari & Kristian, 2011). Many think that videogames are the cause of childhood obesity, as they can lead to a sedentary life, but exergames can subvert this by engaging video game players in physical activity (Koivisto et al., 2011). Since the creation of the Wii console, exercise games have been branching out from their origins. Exergames can be found on consoles as well as on smartphones, and they have a variety of subgenres: dancing, location-based, running, and working out. As the genre keeps expanding there are a variety of ways to move about and enjoy exercise.

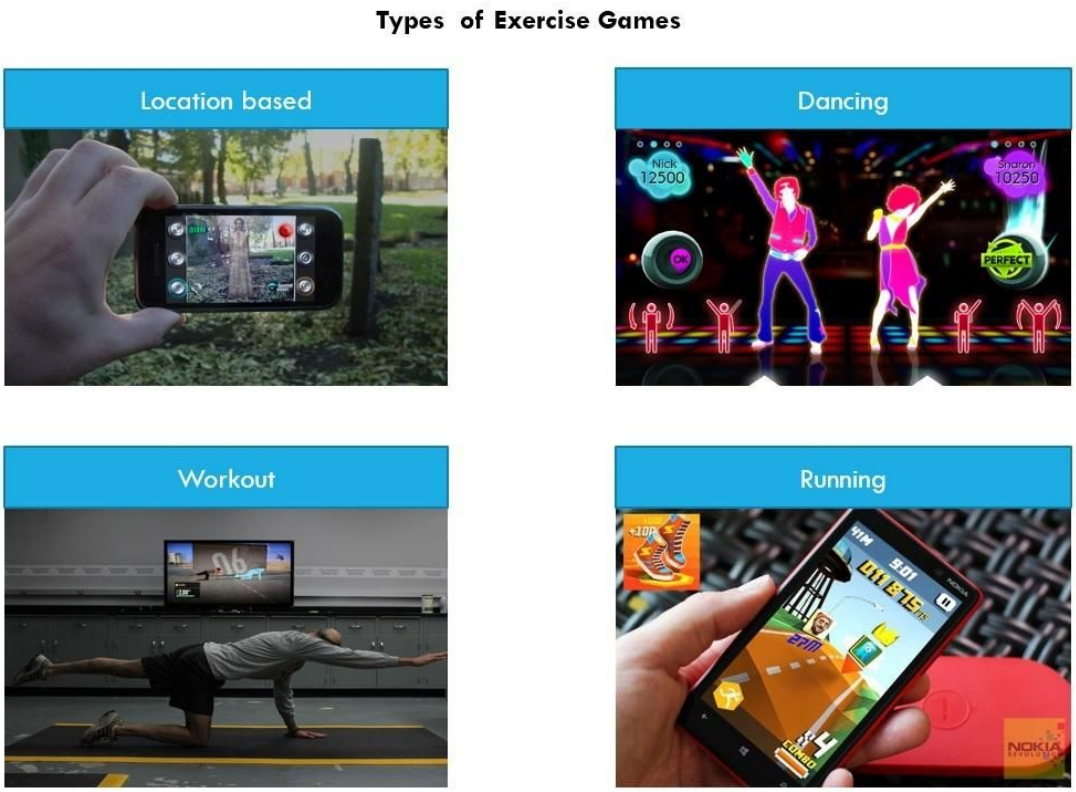

Figure 2.1: Four examples of types of exergames

Figure 2.1 shows examples of the various exergames. Location based games require players to traverse the real world, dancing games require players to match onscreen prompts as shown, workout games require matching various exercises, and running games require the player to follow the game's instructions while running (Koivisto, Sari & Kristian, 2011).

2.1.1 Dancing Games

The dancing subgenre of exergames attempts to make dancing a game activity by scoring player's performance (Smith, 2005). Dancing games use motion sensing technology as a controller (Prather & Korolenko, 2008). Some use pads on the ground to sense when the user step in certain locations, which is most notably seen in the game Dance Dance Revolution (Smith, 2005). ParaParaParadise is a dance game that uses octagonal motion sensing rings

above the user to detect motion (Crampton, 2007). However, it is now more common to use an accelerometer, like in Just Dance, Just Dance Now, Country Dance, and Real Dance (Charbonneau, 2009). These games use either the accelerometer in the WiiMote or in the user's smart phone. Dancing has a great deal of potential for getting people physically active because dancing has been shown to stimulate pleasure senses in the brain, as well as develop deeper emotional understanding (Krakauer, 2008).

2.1.2 Location-Based Games

The defining characteristic of location-based exergames is that they require players to seek out real-world locations to achieve in-game progress (Avouris & Yiannoutsu, 2012). Players can interact with events, landmarks, or items that are mapped to these real world locations by visiting these places. Location-based games tend to use augmented reality (AR) to meld the real world with a virtual one (Javornik, 2016). This can be done in many ways, such as projecting images from the game onto the feed from a phone's camera. Location-based games utilize a phone's GPS technologies to essentially turn the real world into a game board (De Souza, 2006).

Although many of these games are not explicitly designed to encourage exercise, they are designed to make it impossible to play them without engaging in physical activity, as the player must move around in the real world to make progress. In this sense, location-based games persuade players to exercise in a much more subtle fashion than traditional exergames. The recent phenomenon of Pokémon Go is an excellent example of a location-based game that became popular and united large groups of people (Javornik, 2016). Location-based games can be a successful subgenre of exergames because they can potentially appeal to people that are not explicitly interested in using videogames to become more fit, while still improving their

physical well-being.

Although Pokémon Go is a very popular location-based game, there are many other examples of the genre. Pokémon Go developer, Niantic, released a game called Ingress in 2012. This game pits players on different teams against each other to control real world locations (Myk, 2014). Geocaching is another location-based game that has been played since 2000, and it gives players GPS coordinates for real world items for them to discover (Dems, 2011).

2.1.3 Running Games

Running fitness games attempt to gamify jogs and sprints through the use of in game incentives and narrative. Running games use a combination of accelerometer, gyroscope, and occasionally GPS information to measure the player's progress in real time (Cooper, 2015). Some running games focus on narrative and immersion to convince gamers to go for a run. Zombies, Run! is a good example of this, as it is a story-based running game that uses in-game quests to increase the player's pace (Moses, 2012). It motivates people to run by offering new chapters in the story, using an engaging survivor story as an incentive for motivation. It also uses gamification by offering in game resources for exploring new areas.

Other running games focus more on the mechanics of running, and use timing based elements to control pace. Shape up Battle Run incorporates rhythm game elements to regulate when the player's feet makes contact the ground to control his or her running pace (Rubino, 2014). Battle Run uses a scoring system to measure the success of a run based on the player's speed and running form.

Overall, the purpose of running games is to make the process of running more engaging for users. Whether this is done through narrative or through using scores to measure progress,

the end goal is to get users to run more.

2.1.4 Workout Games

Similar to running games, workout games gamify the process of weight and body training by offering in game rewards for workout progress. These games provide players with preset workouts that they must perform to move forward in the game. One example of this is Superhero Workout, which casts the player as a superhero that must save the world by completing exercises. The game offers workouts in the form of missions, and these missions are tailored to different heroes, which are selected by the player based on workout priorities. Additionally, the game tracks calorie information, and actively gives workout feedback based on the player's position, as captured by the phone's camera. Superhero Workout is a good example of many of the techniques that workout games use, as it is story driven, offers immediate feedback, and scores workout progress (Williams, 2014).

2.2 Quantifying Game Experiences

To be able to measure the success of our enjoyment predictions, we need an objective measure of player enjoyment. This is necessary to have to compare an explicits enjoyment score with our predictions to determine model accuracy. Thus, we require a method for quantifying player's game experiences. There are a few tools currently available for quantifying game experiences through explicit indicators. Some of these indicators are questionnaires that are administered after participating in a game session (GEQ and IEQ) (Brockmyer, 2009) (Jennett, 2008), some are scales that are administered separate from any game sessions (PACES) (Kendzierski & Kenneth, 1991), and some are models that predict player enjoyment in

video games (GameFlow) (Sweetser & Wyeth, 2005).

2.2.1 GEQ

The Game Experience Questionnaire or GEQ is a multi-module scale, originally developed to reliably measure deep engagement in video game-playing (See Appendix A for the full questionnaire) (Brockmyer, 2009). It consists of the core questionnaire module, the social presence module, and the post-game module, each of which is meant to be administered, in the given order, immediately after a game session ends. Additionally, a concise in-game version of the GEQ was developed, meant to be administered multiple times during the game session. The core questionnaire and social presence modules probe the players' feelings and thoughts while playing the game, while the post-game module assesses how players felt after they had stopped playing. Each module consists of multiple items rated from 0-4, with 0 referring to not at all and 4 referring to extremely. A grading rubric is given at the end with instructions on which questions refer to what components. To compute the score for each component, the average value of the pertaining items is taken.

The core questionnaire module assesses game experience as scores on seven components: immersion, flow, competence, positive affect, negative affect, tension, and challenge (Brockmyer, 2009). Immersion and flow measure how engrossed the player is in a game. Competence, challenge, and tension refer to the feelings of stress or lack thereof that a game evokes. Negative and positive effects refer to if the player felt that the game was a positive or negative experience. For a robust measure, at least five items are needed per component; since there are a total of 33 items, some items refer to more than one component. As translation of questionnaire items sometimes results in suboptimal scoring patterns, a spare item has been added to all components.

The social presence module investigates psychological and behavioural involvement of the player with other social entities, be they virtual (i.e. in-game characters), mediated (e.g. others playing online), or co-located (Brockmyer, 2009). The module contains 17 items broken down into three graded components: psychological involvement and empathy, psychological involvement and negative feelings, and behavioural involvement. It should only be administered when at least one of the three types of co-players are involved in the game.

The post-game module assesses how players felt after they had stopped playing (Brockmyer, 2009). The module helps assess naturalistic gaming (i.e, when gamers have voluntarily decided to play) as well as experimental research. The module contains 17 items broken down into four graded components: positive experience, negative experience, tiredness, and returning to reality.

In terms of measuring deep engagement in video-game playing on a general level, the GEQ is an effective method for understanding player enjoyment.

2.2.2 IEQ

The IEQ or Immersive Experience Questionnaire is a scale used to subjectively measure immersion in games (See Appendix B for the full IEQ) (Jennett, 2008). It was developed and used as part of Charlene Jennett's paper "Measuring and Defining the experience of Immersion in games," whose purpose was to explore immersion further by investigating whether immersion could be defined quantitatively through three experiments. In each of the experiments, a specific version of the IEQ was administered to the participants and correlated with how they felt at the end of the game.

The first two experiments used the same version, where participants answered 33 items in total based on how far they would agree with the statements indicated (Jennett, 2008). The

first 32 items had a Likert scale with five options ranging from “Strongly Disagree” to “Strongly Agree. The last item asked the participant “How immersed did you feel?”, in which the player could answer on a scale ranging from 1 to 10.

The third experiment used an altered version of the IEQ where participants answered 31 items on a scale of 1 to 5 (Jennett, 2008). However, the range of the answers differed depending on the question asked. The general pattern was that 1 represented “Not at all” or “Very little” while 5 represented “A lot” or “Very much so”.

The IEQ is a useful tool for defining immersion in games, but the scope of the questionnaire is limited to subjectively measuring immersion. Since the GEQ has a section for immersion in addition to other game criteria, it may supersede the IEQ. As such, we will not use the IEQ as one of our external indicators.

2.3 Sensors

One of the predominant ways explicit data is gathered on smarthphones is through the sensors. Most smartphones have sensors to measure motion orientation and various other environmental conditions (Costello, 2016). The two most commonly used sensors are accelerometer and gyroscope. Accelerometers measure the acceleration of an object in different directions, and gyroscopes help determine an object’s orientation.

2.3.1 Accelerometers

An accelerometer is a device used to measure an object’s acceleration, which is a measurement of an object’s change in velocity. Acceleration is defined in the laws of motion equation, Force = Mass X Acceleration, meaning that accelerometers use relationships between force and mass to determine an object’s acceleration (Woodford, 2016). Accelerometers have

found a vast array of applications, from turning off a falling hard drive, to deploying airbags in a crashing car.(Goodrich, 2013). Accelerometers are able to help with these tasks because they are able to measure the acceleration of an object on the x, y, and z axis. Accelerometers usually detect acceleration in two different ways; piezoelectric effect, or capacitance sensor ("A Beginner's Guide to Accelerometer," n.d.). "The piezoelectric effect… uses microscopic crystal structures that become stressed due to accelerative forces. These crystals create a voltage from the stress, and the accelerometer interprets the voltage to determine velocity and orientation" (Goodrich, 2013). Capacitors work in a similar way, when the mass is moved in the device one of the metal plates of the capacitor shifts and moves closer to the other. When the plates are close they transfer an electric current which is sent to be interpreted (Woodford, 2016). In our experiments, we will be using a smartphone's accelerometer data to try to deduce a player's excitement. Presumably, there is some relationship between player enjoyment, and the acceleration with which they are moving their phone. Our machine learning program should be able to find that relationship.

2.3.2 Gyroscope

The basic function of a gyroscope is to use the Earth's gravity to help determine the orientation of an object. Gyroscopes are made up of a rotor, mounted onto a spinning axis in the center of a large wheel. The axis turns while the rotor remains still to measure the gravitational pull (Goodrich, 2013). Since gyroscopes are able to measure the rate of rotation around an axis, they are particularly useful for determining the if an aircraft is rolling too much, or the orientation of a phone. Phones in particular use vibration gyro sensors, which are smaller and less accurate than traditional gyroscopes ("Gyro Sensors - How They Work and What's Ahead," n.d.).

Just as with accelerometers, we will use data gathered from the gyroscope to attempt to

classify game enjoyment. Gyroscopes will be a useful sensor for our data gathering because they will give us information about the rotational acceleration of phones, which is an important facet of phone movement.

2.3.3 Step Detector and Step Counter

Step detectors allow phones to detect when a user has taken a step. Step counters work in concert with step detectors to track the number of steps a user has taken, usually measured over some unit of time. Using the Android API for step detection and counting, the phone's sensors are able to detect when the user is walking, running, or walking up stairs ("Sensor Types," n.d.). However, these sensors should not be triggered by biking, driving or moving in another vehicle ("Sensor Types," n.d.). The step detection and counters are frequently used as metrics for exergames. Step counters offer a simple means to measure the amount of steps a user has taken while playing a game, both indicating the amount of time a user has played the game, as well as how active he or she was while playing it. From step counts, it can be deduced how dedicated a player is to a game. In our experiments, use step counters to measure player's activity while playing Pokémon Go.

2.3.4 AndroSensor

AndroSensor is an Android application that reports information about the state of a device's sensors (Asim, 2013). While the app is limited to the sensors that exist on an android the phone, AndroSensor can monitor the following sensors: It uses an android phone's accelerometer and gyroscope to create additional sensor data. The relevant sensors and features that AndroSensor captures are as shown in Table 2.3.1.

Sensor/Feature	Unit of Measure	Description

Accelerometer	meters per second squared	Measures acceleration. Captures this on the x, y, and z axis.
Linear Acceleration	meters per second squared	Measures forward acceleration. Captures this on the x, y, and z axis.
Gyroscope	radians per second	Measures rotational speed. Captures this on the x, y, and z axis.
Location	Longitude and Latitude	The geographic location.
Orientation	degree	Captures how the phone is oriented relative to the ground.

Table 2.3.1: Androsensor sensors/features

AndroSensor has been used several research studies to gather mobile device data. For instance, a phone running AndroSensor was placed on the back of a tactile picture book page in order to evaluate the user experiences (Kim, 2014). The app has also been used to estimate the roughness of roads based on phone movement in the vehicle (Douangphachanh, 2013).

Figure 3: Screenshot of Androsensor (androfreeware.com)

The app allows the data recorded to be saved as a CSV file attached in an email, or saved directly on a phone. Among other settings, the time interval between data-gathering (sensor sampling rate) can be adjusted to as long as 15 minutes and as short as 0.005 seconds (5 milliseconds), although some Android phones are not able to support this.

The Android sensors that are relevant to our study of exergame enjoyment are the accelerometer and the gyroscope. Data from these sensors is gathered continuously while the subject plays an exergame, and analyzed to infer exergame enjoyment levels.

2.4 Machine Learning

Samuel defined machine learning as the “Field of study that gives computers the ability to learn without being explicitly programmed” (Munoz, 2007). As such, machine learning has a great deal of relevance in the field of Artificial Intelligence, allowing programmers to construct models that can observe data and make predictions about that data. With the advent of increasingly large data sets, recommendation systems are being utilized by companies to recommend products to customers based on their preferences. Additionally, there are many softwares that are programmed with machine learning algorithms such as MATLAB and Weka.

2.4.1 Basics of Machine Learning

There are two major types of machine learning, supervised learning and unsupervised learning (Alpaydin, 2014). Supervised learning has input and output variables, and trains an algorithm to be able to map a predictive function that given certain input will be able to give expected output. In other words, this function, otherwise known as a model, will be able to predict the outcome of a situation given input. The algorithm trains the model by looking at

previous data and building a model based on this data. In unsupervised learning, the prediction algorithms are instead given unlabeled data. The algorithms must establish unifying characteristics, or underlying features of the data. It is the notion of being given “answers”, as in real data points, and being trained based on these real answers that defines supervised machine learning as compared to unsupervised learning.

Another two sub-categories of machine learning problems are classification problems, and regression problems (Brownlee, 2016). Classification problems deal with discrete categories for data. For example, data in a classification problem could state whether a person is happy or sad. In regression problems the data is continuous, meaning it has a value with an infinite range, such as money.

In unsupervised learning the algorithm receives only input data, and given this input data it must create a model that indicates the structure of data. This is primarily done through clustering similar data together or through generating associations between data points.

Machine learning is often used to analyze and make predictions from “big data”, which is a term to describe the huge quantities of data that are unprocessable with traditional data processing methods. Through improvements in processing power, it has become possible for machine learning algorithms to use huge data sets to train models and make predictions in various fields such as healthcare and finance (Levine, n.d). As this field further develops, it will become increasingly possible for companies and the government to be able to predict future trends through the use of machine learning. Similarly, as methods for machine learning improve it will become increasingly possible to train intelligent agents that could make autonomous decisions.

In our project, are using machine learning to train classifiers and predict whether or not users are enjoying exercise games. We use our accelerometer, gyroscope and step count data captured from sensors to build a prediction model. We then analyze the results of these

predictions to determine if they accurate or not.

2.4.2 Weka

Weka is a suite of machine learning software that enables data analysis and predictive modeling. The software allows users to import formatted data files and then easily perform many data mining tasks on this data. Users can find interesting trends in their data through the use of the software's preprocessing, clustering, classification, regression, visualization and feature selection functionality (Eibe & Witten, 2005). Weka is useful because it is lightweight, and does not require users to implement prediction algorithms. It has widespread use for teaching and research purposes. In our project we use Weka to train a machine learning classifier from our experiment data.

2.4.3 MATLAB

MATLAB is an abbreviation for it's full name Matrix Laboratory. MATLAB as a high performance language for technical computation has millions of users worldwide. It is a common tool for math computations, modeling, data analysis, and machine learning. MATLAB is a high level matrix array language with control flow statements and a vast collection of mathematical libraries. MATLAB also has a great deal of modules for statistical modeling, machine learning, signal processing, and other potentially useful features. In our project, MATLAB will be used to help us process our raw smartphone sensor data, and especially extracting features, such as Standard Deviation or Min-Max from segments of our data.

2.4.4 Recommender Systems

As described in the introduction, the overall goal of our project is to help lay down the

foundation for a recommender system. Recommender systems are meant to filter information and predict the preference a user would give an item (Pouly, 2014). These systems have been created in a wide variety of domains, such as shopping (Amazon), movies (Netflix), music (Pandora), and many others. There are three methods used to create these individual preference predictions: content-based, collaborative, and hybrid.

Content-based recommender systems use a user's item-to-item data to make predictions (Pouly, 2014). In these systems, the items a user has preferred in the past is used to recommend similar items, but other users are not considered. To accomplish this, a user preference profile is created, either from explicit ratings and preferences made by the user or from implicit preferences derived from the user's similar items. All available items are given a "weighted similarity metric" measuring how similar items are to each other. Certain systems also have a "relevance feedback loop" where users provide feedback on the recommendations (Pandora's "like" and "dislike" buttons) to improve future recommendations. These systems solve the problem of only popular items being recommended because each user has an individual profile, and new items can be recommended from the beginning. However, it is difficult to recommend items to new users with sparse preference profiles, and recommendations made using this method generally do not provide new insight into preferences.

Collaborative recommender systems use user-to-user and user-to-item data to make predictions (Pouly, 2014). These systems look at groups of users that have liked the same items in the past. Based on how many users have liked similar items and how they rated them, recommendations are made. An advantage of these systems is recommendations can be made outside of categories the user knows, which increases the knowledge of the user's preferences. Disadvantages include needing a large base of people to participate to make good recommendations, which could be difficult for mobile exergames.

Hybrid recommender systems use a combination of content-based and collaborative methods (Pouly, 2014). One way these systems combine recommendation methods is to weight the score generated for each item by each method. Another way is to cascade the results of different methods so that one method refines the results of the next. The combination of these methods can combine advantages and mitigate disadvantages of individual methods.

For all of these recommender systems, there must be enough data to make good predictions of user preferences. Knowing what features of the data correlate to user preferences is important for any recommender system. This is the area of recommender systems that our project focuses on.

2.5 Related Systems

To be able to better design our own exergame enjoyment prediction system, it is important to look at similar systems that infer related user states such as mood from the sensors in a smart phone. From other systems we can determine which strategies have been proven to work, as well as which do not. We can also determine the limits of the current state of the art, and attempt to pursue this.

2.5.1 EmotionSense

EmotionSense is a system for sensing emotions and for monitoring human interaction on mobile devices. Through the use of smartphone sensors, researchers can collect data on group dynamics, collecting information such as how activities, group interactions, and the time of day impact the emotions of individuals. Mobile platforms were targeted for this software, because mobile devices offer an unobtrusive means for monitoring human behavior. It is important for this technology to be unobtrusive because individuals tend to behave differently when they are

aware that they are being monitored. The EmotionSense framework uses sensors, such as gyroscope and accelerometer, that are relatively standard across many different phones, allowing for information to be collected from a great deal of different individuals (Rachuri, Musolesi, Mascolo, Rentfrow, Longworth, & Aucinas, 2010).

The goal of EmotionSense is to enable social psychologists to easily be able to run experiments on test subjects who have installed the software on their phones. To ensure robustness, the framework can be programmed using a declarative language, allowing non-programmers to be able to change the software to suit their needs. EmotionSense is able to recognize different speakers, as well as monitor their emotions by running machine learning classifiers locally, meaning the application handles the process of assigning emotions to user behavior without the help of external hardware or software (Rachuri et al., 2010).

The EmotionSense system is made up of sensor monitors, the programmable framework, and two declarative databases. The monitors logs events to the Knowledge Base repository, which hold all information from the sensors. The other database, the Action Base, is made up of all of the sensing actions that EmotionSense must perform. These actions can be programmed by researchers using the software, and actions are also generated based on sensor data (Rachuri et al., 2010).

The software determines the user's emotional states in large part through analyzing his or her's speech patterns. EmotionSense generates reports on the user's emotions based on a Gaussian Mixture Model (GMM) classifier, which was trained using the Emotional Prosody Speech and Transcripts library. The Emotional Prosody Speech and Transcripts library is the standard library for emotion and speech processing, and it was generated through recording professional actors reading dates and numbers while emulating speech from fourteen different emotional categories. This means that the classifier which determines the user's emotional state was made using vast quantities of voice recordings. In addition to sensing speech,

EmotionSense also tracks accelerometer and gyroscope data, mostly to detect when sensor actions should be activated (Rachuri et al., 2010).

Broad emotion	Narrow emotions
Happy	Elation, Interest, Happy
Sad	Sadness
Fear	Panic
Anger	Disgust, Dominant, Hot anger
Neutral	Neutral normal, Neutral conversation, Neutral distant, Neutral tete, Boredom, Passive

Table 2.5.1: EmotionSense emotion classifications

Table 2.5.1 shows the broad emotions that EmotionSense analyzes, and the more specific specific emotions that are encompassed by the broad emotions (Rachuri et al., 2010).

In a 10-week study with 18 participants, EmotionSense was able to estimate emotional states which generally correlated with the true emotional states of individuals. Overall, the framework offers an interesting tool for psychologists and social scientists to monitor behavior through the use of machine learning.

2.5.3 Other Dance Enjoyment-Related Detection Research

There are two other systems that have tried to using the movements from dancing to enjoyment. The first system is called the Musical Synchrotron. The Musical Synchotron used the accelerometer in the Wiimote, and matched the data to the tempo of the songs playing to measure the participants enjoyment (Demey, 2008). The second system was used to measure the enjoyment of music in a club in order to learn and as time passed create a better music to create a positive environment (Kunh, 2011). The system used wearable accelerometers combined with the data on dancing by the participants and measured four states; dancing, walking, foot tapping, and standing (Kunh, 2011). They match the tempo of dancing and foot tapping against the tempo of the song to decide how much the participants are enjoying the

song. Because we do not have access to the song being played we do not compare the tempo of accelerometer data against the the temp of the song to figure out enjoyment.

3 Methodology

It was our objective to determine if it is possible to create a prediction model that could successfully predict user enjoyment of exergames. To do this we performed three consecutive steps:

1. Pilot study to establish a procedure for generating a prediction model
2. Experiments with Just Dance Now and Pokémon Go to gather sensor data
3. Build a Prediction model using data from our Just Dance Now experiments.

3.1 Flowchart

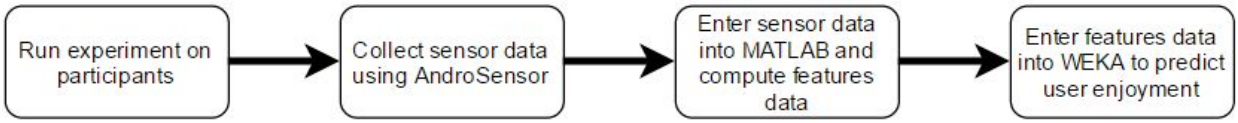

Figure 3.1.1: A flowchart for the sequence of our experiments

The flowchart in Figure 3.1.1 outlines the basic process of our experiments. . The first step in our pipeline was to have participants play either Just Dance Now or Pokémon Go. We used the AndroSensor application running on our test phone to collect sensor data as users played the game. In addition to gathering data from the phone's sensors, we had the users fill out a survey to determine how they felt after playing the game. This survey was used to calculated an enjoyment score. It was important to gather player opinions on how much fun the game was, because when performing our supervised machine learning it was necessary to label how all of the sensor data correlated to enjoyment.

Next, we took the phone sensor data from AndroSensor and entered it into MATLAB to extract mathematical features from the raw sensor data (Asim, 2015). We took these features

and entered them into WEKA, where we used several machine learning classifier functions to create different prediction models. We gave WEKA the exact E-scores of the participants, which we calculated through adding up the point totals in a survey we gave them after the experiment. These prediction models were then able to predict how much a participant enjoyed the game by guessing an E-score from their sensor data.

3.2 Pilot Study

To establish the start-to-finish procedure of generating a prediction model for exergame enjoyment, we conducted a pilot study. In the pilot study, we each participated in two Just Dance Now sessions using AndroSensor to gather sensor data. In one session we acted “excited”, meaning we exaggerated our movements. In the other session we acted “bored”, meaning we tried to minimize our movements. The reason we acted bored and excited in the pilot study was to see if the model could work in an extreme case where we attempted to bias the data to make it easier to predict. If we could not predict the enjoyment in this contrived case, then we would have to evaluate our methods. After we gathered the data, we entered it into MATLAB and broke up the sensor data into one-second segments. For each segment, we computed features such as skewness, kurtosis, min-max difference, standard deviation, and root mean squared. We then entered the data for each session into Weka and created our first enjoyment score (E-score) prediction model. With the only variable being the participant, we checked to see if the model could tell for each segment of a game session whether the participant was enjoying the game or was bored.

3.3 Just Dance Now Experiments

We decided upon using Just Dance Now as our primary exergame for collecting sensor data. Just Dance Now was selected because its three- to five-minute songs were an ideal length for experiments. It was also an appropriate choice because there have been successful studies related to detecting enjoyment from people's dancing patterns, showing that the concept has promise (Kunh, 2011).

We gathered fifty two individuals to participate in a controlled Just Dance Now experiment. Participant ages ranged from eighteen to twenty-three and included an equal number of males and females. Experiments were performed one participant at a time.

The experiment procedure was as follows:

1. We described the experiment procedure to participants.
2. We administered a pre-experiment survey, where participants provided demographic information such as age, gender, and weekly amount of exercise.
3. We activated the Androsensor application on the phone that the participant used. We set the application to capture user motion data every 10 milliseconds.
4. We had the participants play the song "Taste the Feeling". The participant stood seven to ten feet from the screen, and all proctors provided some privacy by not watching the participant play.
5. Once the song was completed, the participant answered a survey to gauge how much they enjoyed the game using strongly disagree to strongly agree questions. (See Appendix B)
6. The participant began their second Just Dance game session, in which they were instructed to pick of choice and dance.

7. The participant filled out the same survey mentioned above to gauge how much they enjoyed the second song.
8. The experiment ended with a simple post-survey which explicitly asked whether the participant enjoyed the Just Dance Now sessions.

3.4 Pokémon Go Experiments

For our secondary exercise game we selected Pokémon Go. Pokémon Go was selected because it is a radically different type of exergame than Just Dance Now, and its ubiquity made it immediately familiar to many people. For our Pokémon Go experiments we had five participants, four of which were male, and one which was female. The procedure for our Pokémon Go experiments was as follows:

1. We described the experiment procedure to the participant.
2. We administered a pre-experiment survey, where each participant provided demographic information such as age, gender, and weekly amount of exercise.
3. We asked each participant if they had played Pokémon Go, and if they had not, we informed them of the basics of the game.
4. We told each participant to walk around the WPI campus and attempt to go to 10 PokéStops (locations within the real world that offer in-game benefits). The experiment ended when the participant went to ten PokéStops or when twenty minutes expired. Whichever came first. Each participant was also informed to catch Pokémon as their own leisure. We activated the Androsensor app to capture the motion data when the participants began.
5. We had each participant fill out a post-experiment survey, where they were asked several questions to implicitly determine their enjoyment of the game. We then explicitly

asked the player if they enjoyed the game.

3.5 MATLAB and Weka

After gathering data from our Just Dance Now experiments, we had to transform the raw sensor data into features which could correlate to user enjoyment and be used to generate a prediction model. We adapted existing MATLAB code that had been developed from other WPI projects. The accelerometer data was developed from Muxi Qi's master thesis (Qi, 2016). The gyroscope features were developed from Christina Aiello's thesis (Aiello, 2016). The full list of the features we calculated can be found in the Analysis Chapter. After this, we used the previously created code to determine the correlation for each feature to E-scores by determining the p-value and correlation coefficient. We then removed the features that had p-values greater than 0.05. We took the correlated features and entered them into Weka to create our prediction model.

We varied many factors to shape the creation of our model. The first factor was which classifier we used to create our model. We ran four different classifiers on the data, Random Forest, SMO, J48, and NaiveBayes. Each of these classifiers created different prediction models. Additionally, we used ten-fold cross-validation to ensure that all of our data was used for both testing and training our model.

Another factor that we varied was bucket size. When predicting the E-scores, we chose to have our prediction model attempt to predict if the enjoyment was within a certain range of E-scores, as opposed to predicting specific scores. We called these various E-score prediction ranges different buckets. For instance if we had two buckets for player enjoyment, "having fun", and "not having fun", then "not having fun" would represent any E-score from 0 to 40, and "having fun" would represent an E-score from 40 to the maximum score of 80.

The final element that we varied when creating our prediction models was using only the first song that users played in the Just Dance Now experiment, or using both songs the user had played. Since the first song was the same for all participants, we wanted to see if limiting the sensor data that the model was being built off of would make a noticeable difference in the success rate of E-score prediction.

We tested combinations of these variables with each other to make numerous prediction models. We saved these prediction models, as well as the Weka output that helps determine the effectiveness of each model, such as the stratified cross-validation summary, and the confusion matrix. We then used the Weka output to determine which prediction models had the highest percentage of correct classification of player enjoyment.

Once we had determined a few of our of best prediction models, we evaluated how well our prediction models performed predicting the player enjoyment of Pokémon Go. This was important to test because we wanted to see if our prediction model, which had been created with entirely Just Dance Now data, could easily be generalized to an exercise game of a different genre.

4 Results and Analysis

As previously discussed, the primary objective for our project was to create a classification model that can be used to predict user enjoyment for exercise games from their smartphone sensor data. In this chapter we analyze the effectiveness of various combinations of different features, machine learning classifiers, and other variables. Through this analysis we justify the combination of these variables to use for our final classification model for exergame enjoyment. We also discuss demographics information, as well as some of the results of our surveys.

4.1 Demographics

This section presents the demographics of the participants in the Just Dance Now Experiment.

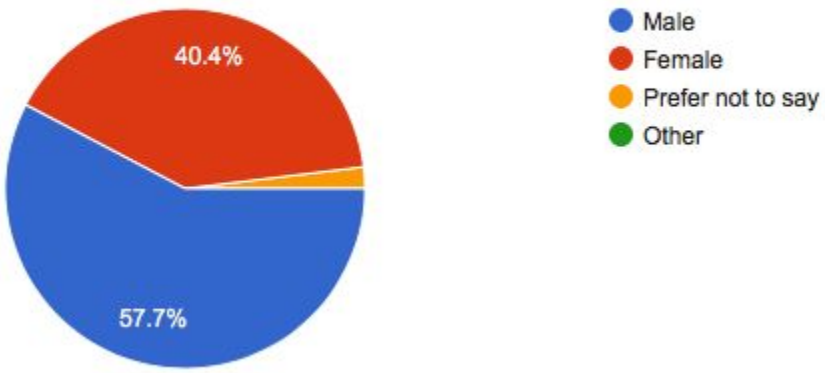

Figure 4.1.1: Pie Chart of Gender of Participants

In the Just Dance Now experiment we had a majority of male participants. As shown in Figure 4.1.1, 21 of participants were female, 30 of participants were male, and 1 prefered not to

answer.

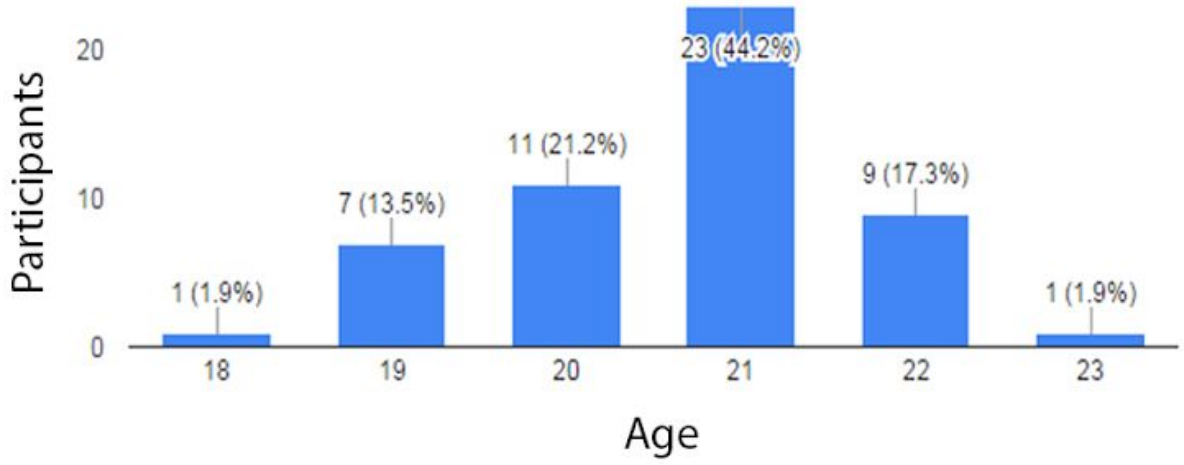

Figure 4.1.2: Age of study participants

Figure 4.1.2 shows the ages of the participants in the Just Dance Now Experiment. All of the participants were college students, so the age range was inherently limited. The participants were all between the ages of 18 and 23, with 21 being the most common age.

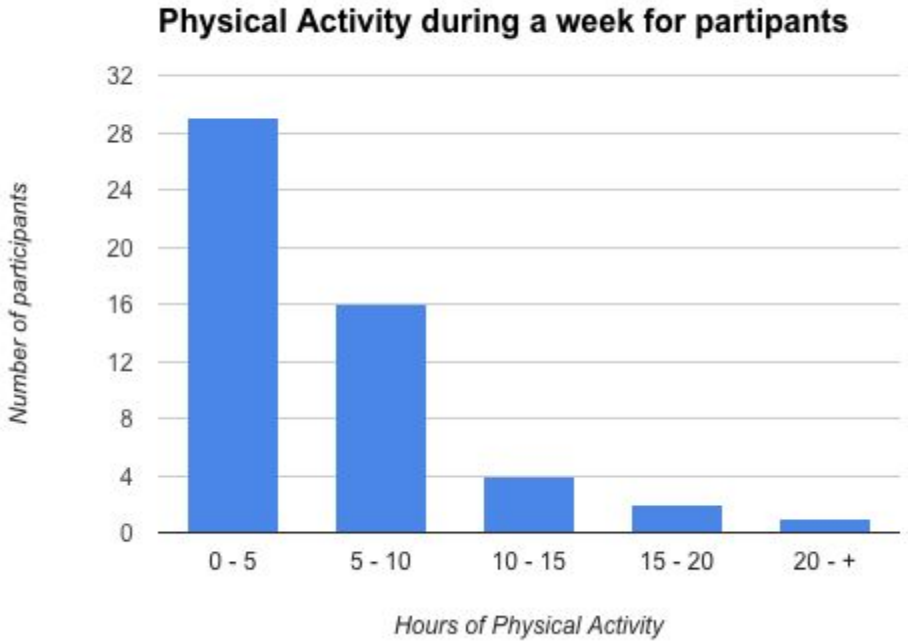

Figure 4.1.3: Average physical activity per week of participants

Figure 4.1.3 shows the participants' average time per week they spend engaged in a

physical activity. Most of the participants exercised between zero to five hours.

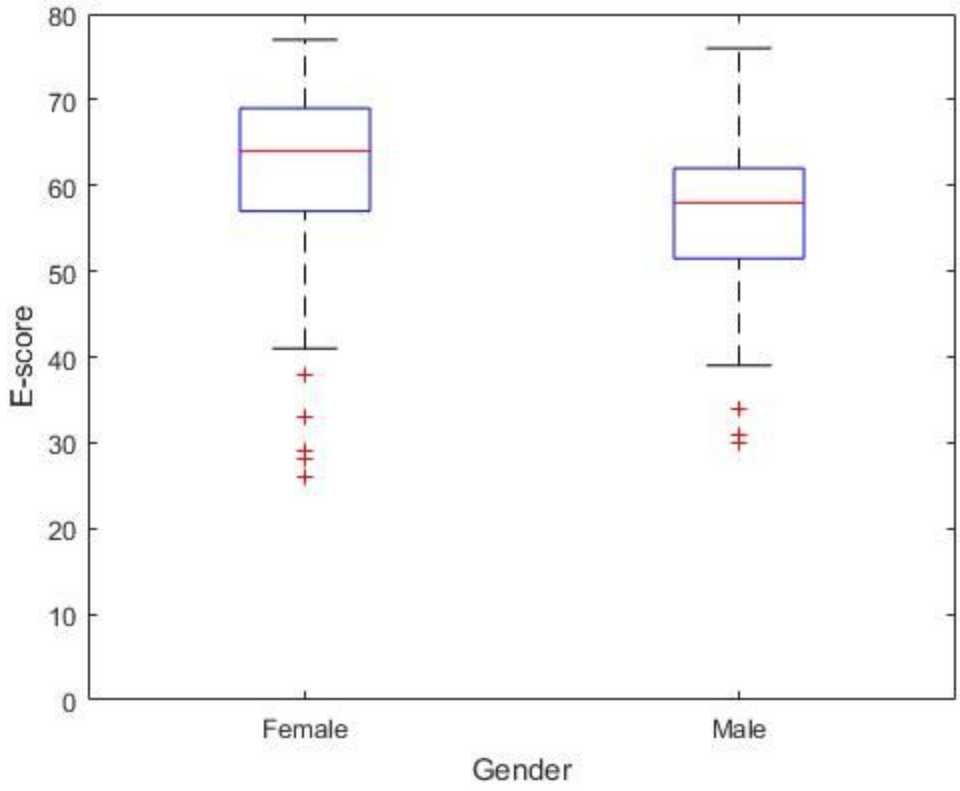

Figure 4.1.4: E-score vs gender box-and-whisker plot for Just Dance Now Experiment

Figure 4.1.4 compares the E-scores to the genders of the participants for Just Dance Now. It shows that there was no notable difference between males and females. The p-value is .0746 which is greater than the threshold of 0.05, means it is not statistically significant, and the correlation coefficient is -0.1773 which means it is not correlated.

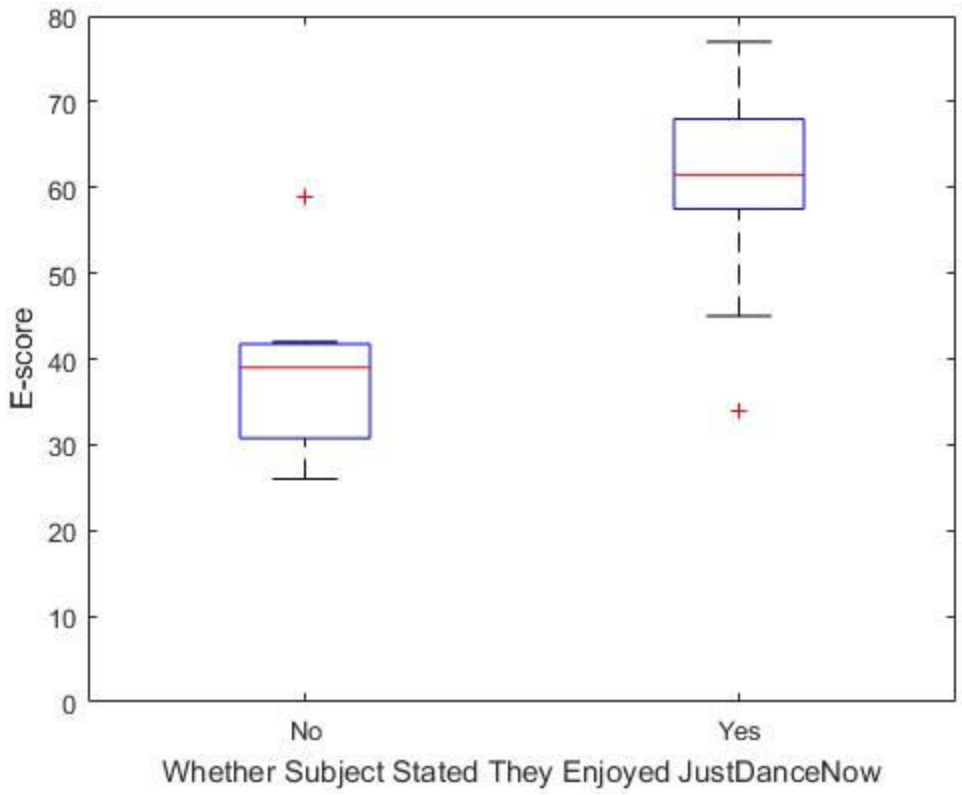

Figure 4.1.5: Song 2 E-score vs enjoyment

Figure 4.1.5 compared the E-score calculated using a question on the survey explicitly asking them whether they enjoyed the game or not. We use this question to validate the survey that was used to calculate participants E-scores. As shown by the E-scores from the second song, which was chosen by the participants, there is a stark contrast between the two distributions. This suggests that the E-score calculation was a good gauge on whether someone enjoyed the game or not. The p-value is 0.00000 which means that it is statistically significant, the correlation coefficient is 0.6770 which means it's highly correlated.

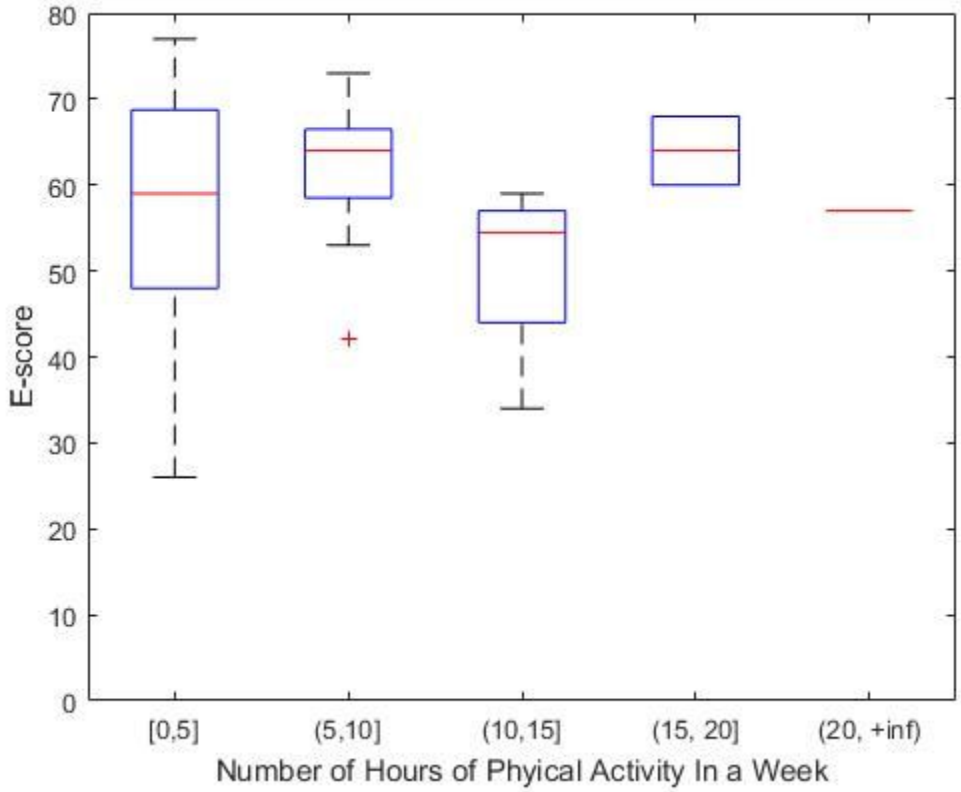

Figure 4.1.6: Hours of Physical Activity per week vs E-score

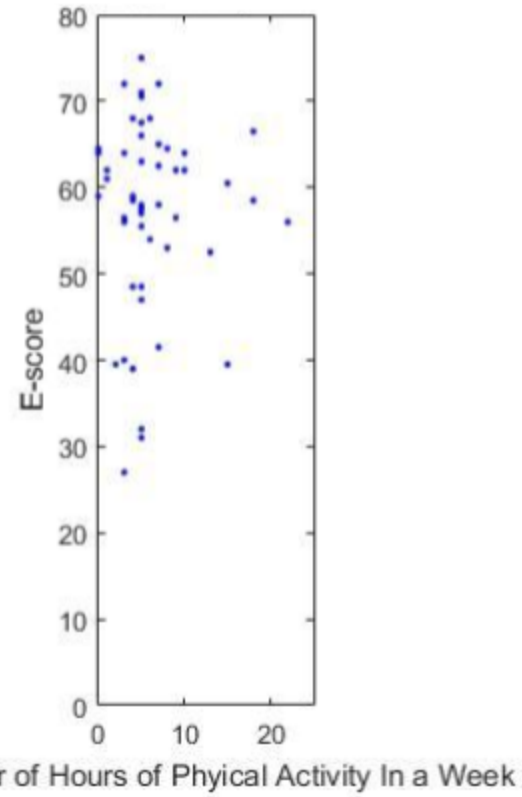

Figure 4.1.7: Hours of Physical Activity per week vs E-score

Figure 4.1.6 compares the participants E-score with their average number of hours of physical exercise during a week broken into 5-hour buckets. There were few participants in the buckets with more hours of exercise per week, but the majority of participants exercised between zero and ten hours a week. There does not seem to be any relation between the amount that the participant exercised, and the amount they they enjoyed the game. There was no reasonable trend line to fit in Figure 4.1.7.

4.2 E-score Distribution

Before one can understand the predictions of E-scores, it's helpful to know the distribution of E-scores from the experiment participants.

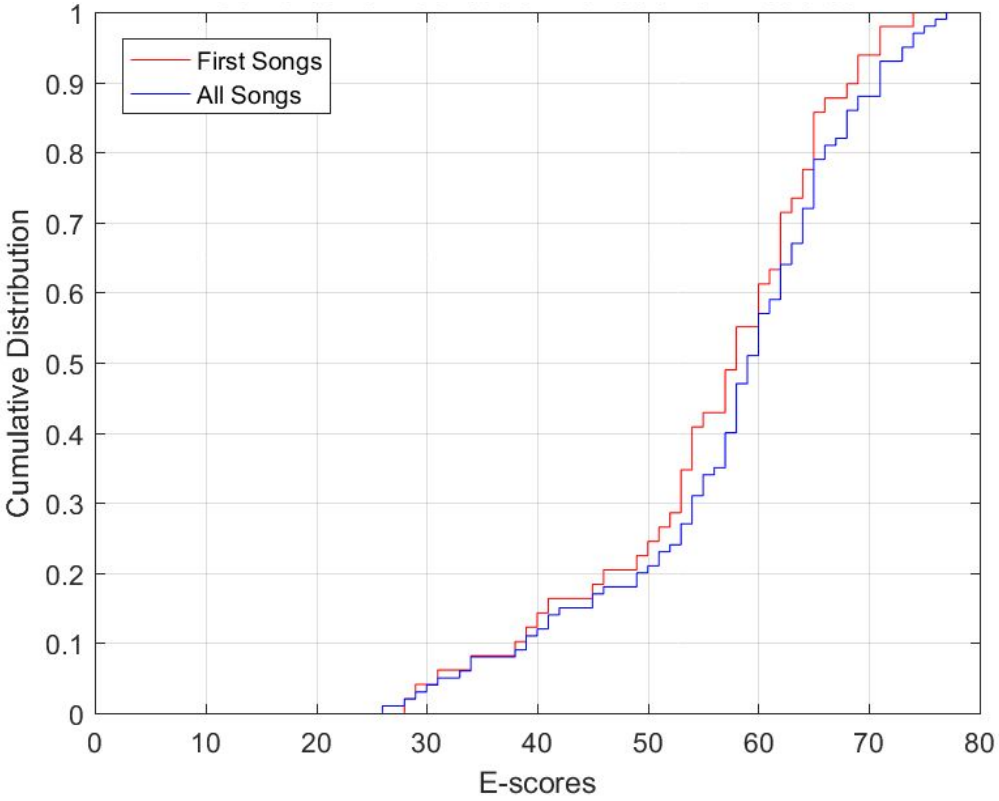

Figure 4.2.1: Cumulative Distribution of E-scores

The range of possible E-scores was 0 to 80, while the range of actual E-scores was 26

to 77. The cumulative distribution graphs indicate the median E-score overall was 59 for both song data, while the median E-score for the first song was 58. The average for both song data was approximately 57.3 while the average for the first song was approximately 55.9. There is little difference between these distributions, although the E-scores for the both songs is consistently slightly greater than for the first songs, indicating that the second songs (the ones participants chose) were enjoyed more.

4.3 Feature Selection

Features used to create the prediction model that do not correlate to the E-score may confuse the classifier and therefore reduce the number of correct predictions.

We calculated 31 features from the mobile sensor data during our experiments. As previously stated, we derived these features from previous study WPI projects (Qi, 2016) (Aiello, 2016). Of the 31 features, 27 were calculated based on the accelerometer (x, y, and z) and 4 were calculated based on the gyroscope (x, y, and z). Depending on the song data used to create the prediction model, features correlated differently to the E-score.

Feature Names	Features Coef	Features Coef(abs)	P-value	Predictable (p<0.05)
radioSpectralPeak_DCT	0.3095	0.3095	0.0017	1
radioSpectralPeak	0.2957	0.2957	0.0028	1
radioSpectralPeak_FFT	0.2622	0.2622	0.0084	1
spectralCentroid	-0.1694	0.1694	0.0921	0
averageStepTime	-0.1420	0.1420	0.1589	0
peakFreq	0.1394	0.1394	0.1667	0
coef of var of stepTime	0.1271	0.1271	0.2076	0
numSteps	0.1187	0.1187	0.2395	0
averageCadence	0.0981	0.0981	0.3316	0
averageStepLength	0.0981	0.0981	0.3316	0
std	0.0980	0.0980	0.3319	0
rms	0.0980	0.0980	0.3319	0
YZEllipse	0.0919	0.0919	0.3631	0
kurtosis	-0.0884	0.0884	0.3820	0
harmonic ratio	0.0868	0.0868	0.3906	0
energy in _5 to 3	0.0826	0.0826	0.4141	0
wavelet entropy	0.0802	0.0802	0.4275	0
wavelet band	-0.0800	0.0800	0.4289	0
averagePower	0.0776	0.0776	0.4427	0
windowed energy in _5 to 3	0.0769	0.0769	0.4468	0
entropy rate	0.0752	0.0752	0.4572	0
gaitVelocity	0.0738	0.0738	0.4656	0
cross correlation	-0.0736	0.0736	0.4670	0
snr	0.0725	0.0725	0.4732	0
thd	0.0699	0.0699	0.4898	0
XZEllipse	0.0561	0.0561	0.5795	0
calculatedVolume	-0.0503	0.0503	0.6194	0
skewness	-0.0494	0.0494	0.6255	0
minMaxDiff	0.0398	0.0398	0.6944	0
bandwidth	-0.0357	0.0357	0.7246	0
XYEllipse	0.0184	0.0184	0.8556	0

Table 4.3.1: Feature correlation to E-score for both songs

For instance, Table 4.3.1 shows the correlations of the features to the E-scores for all the song data collected during the Just Dance Now experiments. At the top are the features that have a p-value below 0.05, while the rest are sorted by decreasing absolute correlation coefficient. As shown in Table 4.3.1, the three correlated features are all based on maximum spectral density (power over frequency). The reason these features are most correlated (albeit weakly correlated) to E-score could be because they represent the amount of energy used in the phone's movements. A person expending more energy while playing could be enjoying it

more.

It was a possibility that it may be easier to determine the truly correlated features using a single Just Dance Now song, so we used only the first songs for each participant (which were all the same song). Determining the feature correlations using the first songs, only the discrete cosine transform (DCT) of the maximum spectral density feature correlated to E-score even weakly. These feature correlations are shown in Table 4.3.2. Peak frequency and the other maximum spectral density features were the next most correlated features to E-score.

We split up the song data into the songs that had E-scores less than or equal to 40 and greater than 40, because 40 was the center of the E-score range and represented neutral enjoyment. In order to account for the fact that almost 90% of our participants said they enjoyed the game, we sampled an even distribution of twelve songs with E-scores above 40 as we had twelve songs with E-scores below 40. This way we would have an equal number (12) of songs in both bins. Using this data, we produced the feature correlations shown in Table 4.3.3. The two correlated features were different from those in Table 4.3.1, although both describe energy used. These are moderately correlated to E-score, while the spectral density features were weakly correlated. The reason for this may be that we used fewer songs to correlate features to E-scores, thus reducing the accuracy of these results.

Feature Names	Features Coef	Features Coef(abs)	P-value	Predictable (p<0.05)
radioSpectralPeak_DCT	0.3014	0.3014	0.0354	1
spectralCentroid	-0.2629	0.2629	0.0680	0
peakFreq	0.2616	0.2616	0.0694	0
radioSpectralPeak	0.2357	0.2357	0.1031	0
radioSpectralPeak_FFT	0.2230	0.2230	0.1235	0
YZEllipse	0.2196	0.2196	0.1296	0
thd	0.2000	0.2000	0.1682	0
averageStepTime	-0.1989	0.1989	0.1707	0
XZEllipse	0.1813	0.1813	0.2124	0
numSteps	0.1697	0.1697	0.2437	0
averageStepLength	0.1690	0.1690	0.2458	0
averageCadence	0.1690	0.1690	0.2458	0
skewness	-0.1480	0.1480	0.3103	0
gaitVelocity	0.1479	0.1479	0.3105	0
kurtosis	-0.1468	0.1468	0.3143	0
std	0.1442	0.1442	0.3229	0
rms	0.1442	0.1442	0.3229	0
averagePower	0.1224	0.1224	0.4023	0
XYEllipse	0.0996	0.0996	0.4958	0
coef of var of stepTime	0.0960	0.0960	0.5115	0
energy in _5 to 3	0.0802	0.0802	0.5840	0
bandwidth	0.0718	0.0718	0.6240	0
harmonic ratio	-0.0456	0.0456	0.7558	0
entropy rate	0.0365	0.0365	0.8036	0
windowed energy in _5 to 3	0.0301	0.0301	0.8372	0
snr	-0.0291	0.0291	0.8427	0
cross correlation	0.0239	0.0239	0.8703	0
minMaxDiff	-0.0224	0.0224	0.8788	0
calculatedVolume	-0.0096	0.0096	0.9476	0
wavelet entropy	0.0024	0.0024	0.9867	0
wavelet band	-0.0022	0.0022	0.9883	0

Table 4.3.2: Feature correlation for first songs

Feature Names	Features Coef	Features Coef(abs)	P-value	Predictable (p<0.05)
energy in _5 to 3	0.4864	0.4864	0.0160	1
windowed energy in _5 to 3	0.4708	0.4708	0.0202	1
radioSpectralPeak_DCT	0.4041	0.4041	0.0502	0
wavelet entropy	0.3623	0.3623	0.0819	0
wavelet band	-0.3619	0.3619	0.0822	0
radioSpectralPeak_FFT	0.3498	0.3498	0.0938	0
cross correlation	0.3012	0.3012	0.1527	0
numSteps	0.2835	0.2835	0.1795	0
radioSpectralPeak	0.2760	0.2760	0.1917	0
averageStepTime	-0.2725	0.2725	0.1977	0
bandwidth	-0.2200	0.2200	0.3017	0
averageCadence	0.2103	0.2103	0.3241	0
averageStepLength	0.2103	0.2103	0.3241	0
entropy rate	0.1939	0.1939	0.3639	0
std	0.1815	0.1815	0.3960	0
rms	0.1815	0.1815	0.3960	0
gaitVelocity	0.1701	0.1701	0.4269	0
spectralCentroid	0.1649	0.1649	0.4414	0
coef of var of stepTime	-0.1509	0.1509	0.4815	0
averagePower	0.1441	0.1441	0.5017	0
peakFreq	0.1437	0.1437	0.5030	0
XYEllipse	0.1367	0.1367	0.5242	0
snr	0.1064	0.1064	0.6206	0
YZEllipse	0.0893	0.0893	0.6783	0
kurtosis	-0.0881	0.0881	0.6822	0
minMaxDiff	0.0563	0.0563	0.7940	0
skewness	-0.0439	0.0439	0.8387	0
thd	0.0296	0.0296	0.8908	0
harmonic ratio	-0.0191	0.0191	0.9293	0
calculatedVolume	-0.0129	0.0129	0.9522	0
XZEllipse	-0.0126	0.0126	0.9536	0

Table 4.3.3: Feature correlation for even split around 40 (12 each side)

4.4 Data Processing and Classifiers

Key variables can be changed when making prediction models through Weka, including altering the dataset through data processing and running various machine learning algorithms. This section describes the different variables adjusted while creating the best prediction model.

4.4.1 Data Processing

Weka takes in a dataset as input and runs a machine-learning algorithm on that dataset in order to generate a prediction model. We choose the contents of the dataset to generate different prediction models through data processing, which is the selection of data to retrieve, transform, or classify information. Table 4.4.1 shows the different methods used to vary the dataset we inputted into Weka.

Data Processing Method	**Possible Values**	**Description**
Number of Bins	2, 3, 6, 10	The number of bins (buckets) for categorized the E-scores into.
Bin range	Bins split at the median of our observed E-scores (i.e. 0-59, 60-80) , bins split at the median possible E-score (i.e. 0-40, 41-80), bins with same number of possible E-scores (i.e. 0-26, 27-53, 54-80)	The range or size of each categorized E-score bin.
Sample of song sessions	Both songs (100), first songs (49), sampling of both songs (24)	The song sessions included in the dataset.
Features	All features, statistically-significant features, Weka-chosen feature selection, combinations of other statistically significant features	The features used to create the prediction model

Table 4.4.1: Data Processing methods

Number of bins refers to the number of bins we categorized E-scores into. We tested sizes of 2, 3, 6 and 10, with 2 bins loosely translated into like/dislike and 3 bins loosely translated into like/neutral/dislike.

For bin range, we tried different sizes including bins split at the average of our observed E-scores (i.e. 2 bins with E-scores between 0 and 59 placed in one bin and of E-scores between 60 and 80 placed in another), split at the halfway point of possible E-scores (i.e. 2 bins with E-scores between 0 and 40 placed in one bin and E-scores between 41 and 80 placed in another), and bins with the same number of possible E-scores (i.e. E-scores between 0 and 10, 10 and 20, 20 and 30, etc.). Different bins sizes altered the accuracy of the prediction model, so we tried many variations to improve the accuracy.

Sample of song sessions is a mix of data from the different song types and number of songs chosen from that type. Both songs includes sessions from both song data. First songs means data from the same song across all participants. Sampling of both songs is used for evenly-sized 2 bins where we select the same number of songs in the smaller bin as the larger bin by sampling different areas of the larger bin.

Features refers to the which features we create the prediction model off of, depending on whether we cull any misleading features or include all features. All features includes all 31 features. Statistically significant only includes the features with a p-value less than or equal to 0.05. Weka-chosen feature selection uses Weka's built-in feature selection to select different features. Mixes of other correlated features includes features that had a p-value less than or equal to 0.05 for other datasets, specifically split depending on number of songs chosen.

4.4.2 Classifiers

Weka has many built-in machine-learning algorithms used to create prediction models. Table 4.4.2 briefly describes different classifiers ran on our datasets.

Classifier	Description
Random Forest	Class for constructing a forest of random decision trees
J48	Class for generating a pruned or unpruned C4.5 decision tree
SMO	Class implements a sequential minimal optimization algorithm for training a support vector classifier
Naive Bayes	Class for a Naive Bayes classifier using estimator classes

Table 4.4.2: Popular classifiers and their descriptions

We chose these classifiers because they represent a wide variety of algorithms that cover the basics of most machine-learning algorithms. Random Forest and J48 use different methods of analyzing decision trees (also called classification trees), SMO (Sequential Minimal Optimization) is a faster implementation of the popular SVM (Support Vector Machine) algorithm, and Naive Bayes is a simple probabilistic classifier that applies Bayes' theorem with strong independence assumptions between the features.

4.4.3 Stratified Cross-validation Summary

After running a classifier on any dataset, Weka displays a number of sections as output. Table 4.4.3 describes the output of one of these sections: a stratified cross-validation which summarizes useful statistics calculated from the analysis.

Output	Description

Correctly Classified Instances	Percentage of test instances correctly classified (also called accuracy or sample accuracy)
Incorrectly Classified Instances	Percentage of test instances incorrectly classified
Kappa statistic	Agreement of prediction with the true class; 1.0 signifies complete agreement, 0.0 is equivalent to random chance.
Mean absolute error	Average magnitude of the errors in a set of forecasts, without considering their direction.
Root mean squared error	Quadratic scoring rule for measuring average magnitude of the errors (squared before averaging)
Relative absolute error	Ratio that describes the mean absolute error divided by the classifier's error
Root relative squared error	Ratio that describes the root mean squared error divided by the classifier's error
Total Number of Instances	Number of instances in the dataset

Table 4.4.3: Stratified cross-validation summary description

Out of these eight outputs, we present the correctly classified instances and the kappa statistic fields, as these are the most important outputs used to determine which combination of data processing and machine-learning algorithm generates the most most accurate prediction model.

4.4.4 Detailed Accuracy by Class

In addition to the stratified cross-validation summary, the detailed accuracy by class section yields detailed information for each class in the dataset. Table 4.4.4 describes the different output shown in the detailed accuracy by class section.

Output	**Description**
TP Rate	Rate of true positives (instances correctly classified as a given class)

FP Rate	Rate of false positives (instances falsely classified as a given class)
Precision	Proportion of instances that truly belong to a class divided by the total instances classified as that class
Recall	Proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate)
F-Measure	A combined measure for precision and recall calculated as 2*Precision*Recall / (Precision+Recall)
MCC	Measure of the quality of binary classification
ROC Area	Probability that a positive will be ranked higher than a negative. Optimal classifiers have ROC area values approaching 1, while 0.5 is comparable to random guessing
PRC Area	Probability used for determining whether the dataset has class imbalance problems
Class	Labels for the different classes (bins) the data is split into

Table 4.4.4: Detailed accuracy by class

These fields give more insight to the success of each class and allows us to compare classes between one another. The most important field for us is the weighted average of the receiver operating characteristic area for all classes, which we used to help determine the most optimal prediction model.

4.4.5 Confusion Matrix

Weka also displays a confusion matrix as part of the output. A confusion matrix, also called an error matrix or a special kind of contingency table, show the performance of a classifier. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class. The number of correctly classified instances is the sum of the diagonals in the matrix; all others are incorrectly classified.

In Table 4.4.5, of the 34 actual E-scores in the “a” bucket, the system predicted that 17 were in the “a” bucket, 12 were in the “b” bucket, and “5” were in the c bucket. Similarly, the system predicted that 10 were “a”, 9 were “b”, and “14” were c for the 33 actual E-scores in the “b” bucket. Lastly, the system predicted for the 33 actual E-scores in the “c” bucket that 10 were “a”, 13 were “b”, and 10 were “c”. In other words, the system predicted 36 correct instances for Table 4.4.5.

		Predicted class		
		a	**b**	**c**
Actual class	**a = 55**	17	12	5
	b = 63	10	9	14
	c = 80	10	13	10

Table 4.4.5: Sample confusion matrix for 3 buckets

A confusion matrix is useful for visualizing the performance of a classification model. We used the confusion matrix to ensure that the prediction model was attempting to predict between the different bins rather than leaving out some bins, or worse, placing all E-scores into the same bin, as well as look for patterns in any misclassification

4.5 Just Dance Now Final Model

After generating numerous prediction models from our Just Dance Now data, we determined the most effective prediction models by splitting the data into 90% training data and 10% test data through a process called k-fold cross-validation with a k-value of 10. We determined the best prediction models by comparing three metrics: the percent of instances that were correctly classified, the kappa statistic, and the weighted average of the ROC area. Table

4.5.1 highlights how these prediction models compared to each other, with the best prediction model for that dataset generated from the algorithms highlighted and bolded.

Both songs, 2 bins (0-59, 60-80), even distribution (50 songs per bin), 3 features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)				
	Random Forest	**J48**	**SMO**	Naïve Bayes
Correctly Classified Instances	52.0%	**65.0%**	**65.0%**	62.0%
Kappa Statistic	0.040	**0.293**	**0.292**	0.232
ROC Area Weighted Average	0.546	**0.613**	**0.644**	0.599
First Songs, 2 bins, even distribution (24-25 songs per bins), all features				
	Random Forest	J48	SMO	**Naïve Bayes**
Correctly Classified Instances	59.2%	53.1%	65.3%	**69.4%**
Kappa Statistic	0.183	0.059	0.303	**0.389**
ROC Area Weighted Average	0.648	0.497	0.651	**0.677**
Both songs, 2 Bins (0-40, 41-80), even distribution (12 songs per bin), 3 features (energy in 0.5 to 3, windowed energy in 0.5 to 3, and radioSpectralPeak_DCT). *Note that we added radioSpectralPeak_DCT because it had a P-value of 0.0502, which is close enough to having a P-value below 0.05.				
	Random Forest	J48	SMO	**Naïve Bayes**
Correctly Classified Instances	66.7%	70.8%	66.7%	**75.0%**
Kappa Statistic	0.333	0.417	0.333	**0.500**

ROC Area Weighted Average	0.715	0.590	0.667	**0.590**

Table 4.5.1: Comparison of our best prediction models

From these results, it seems that J48, SMO, and Naive Bayes classifiers generate the best prediction models under different conditions. Random Forest serves as a consistent baseline to compare against, and the overall best prediction model with 75% accuracy was generated from running Naive Bayes on the dataset consisting of both songs, 2 bins split between 0 to 40 and 41 to 80 of equal size (12 songs per bin) and the energy in 0.5 to 3, windowed energy in 0.5 to 3, and radioSpectralPeak_DCT features. Note that we kept radioSpectralPeak_DCT in the dataset because it had a P-value of 0.0502, which is close enough to having a P-value below 0.05. This dataset performed better than the one without radioSpectralPeak_DCT, which justifies our addition of radioSpectralPeak_DCT into the mix. Also note that the prediction model had a relatively strong kappa statistic of 0.5 and a decent ROC Area weighted average of 0.59. Lastly, Table 4.5.2 shows the confusion matrix for this prediction model. It indicates that the prediction model is very good at predicting for the “dislike” bin and above average at predicting for the “like” bin.

		Predicted class	
		a	**b**
Actual class	**a = 40**	11	1
	b = 80	5	7

Table 4.5.2: Confusion matrix for overall best prediction model

4.6 Pokémon Go Tests

In the Pokémon Go experiment we had five participants. One was female and four were male. The weight of participants ranged from 100 lbs to 200lbs. Participants were limited to college students and the age range was 18-21 with a standard deviation of 1.41 and a mean of 19. The participant's heights ranged from 5'0" to 6'4", with a standard deviation of 5.932', and a mean of 69.8'. None of them spent any time playing any exercise game during the week. A summary of this information is presented in Table 4.6.1.

Gender Split	Weight range	Age Range	Age std dev.	Age mean	Height Range	Height std dev.	Height Mean
4 male, 1 female	100 lbs - 200 lbs	18-21	1.41	19	5'0" - 6'4"	5.932'	69.8'

Table 4.6.1: Summary of demographics from Pokemon Go experiment

To test the accuracy of our prediction model on other exergame genres, we applied our overall best prediction model to our Pokemon Go data and obtained a 75% accuracy with a kappa statistic of 0 and an ROC area weighted average of 0.333. Table 4.6.2 shows the corresponding confusion matrix, which tells us that all predictions were made for the "like" bin.

		Predicted class	
		a	**b**
Actual class	**a = 40**	0	1
	b = 80	0	3

Table 4.6.2: Confusion matrix for overall best prediction model used with Pokemon Go data

With an accuracy of 75%, we would normally conclude that the prediction model had the same accuracy regardless of whether the data came from Pokemon Go or Just Dance Now. However, the high accuracy of the results is skewed because we only obtained Pokemon Go data from four participants. The sub-optimal kappa statistic value of 0 that is equivalent to random guessing, the weak ROC area weighted average value of 0.333, and the confusion matrix showing that all predictions were made for the “like” bin leads us to conclude that we do not have sufficient Pokemon Go data to properly test our prediction model on.

5 Conclusions

Exergames offer one potential solution to the societal problem of physical inactivity. Although exergames can get users active who would normally not consider physical activity, games a problem with player retention. To solve this problem the CyPRESS (Cyber Physical Recommender System) is being developed.

The purpose of our project was to use machine learning classifiers to predict user enjoyment of exercise games from smartphone sensor data.

We aimed to develop an approach towards predicting user enjoyment by collecting sensor data, extracting features from the sensor data, and then creating a prediction model from these features in Weka. This was meant to work as a first step towards eventually creating a recommender system.

Through our experiments and our subsequent creation of prediction models, we were able to draw conclusions about the validity of our methods for calculating E-scores, and determine the most predictive features, the best classifier types and criteria for the best prediction model.

5.1 E-score Calculator Correlates to User Enjoyment

Our method for calculating E-scores effectively captures user enjoyment. As shown in Figure 4.1.5, when we compared the calculated E-scores to the question in our survey that explicitly asked if they enjoyed the game, there was correlation between their E-score and their answer to this question. The graph shows this, as in the box and whisker plots for the yes and no answers, there is no overlap between the player's E-scores for these questions.

5.2 Feature Selection using Correlation was Effective

It was more effective to build a model using a smaller number of features that correlate with a p-value < 0.05, rather than a larger number of features with weaker correlation or p-value > 0.05. Our results show this because when we built models with our full list of 31 features, compared to when we built our models with only the features with p-value < 0.05, the models that were built with the small list of features with a better p-value performed better on average. This can be seen in Appendix E.

From our analysis, we could also conclude that since the spectral density peak (radioSpectralPeak in the p-value tables) and the energy in 0.5Hz to 3Hz features had the greatest correlation with p-value < 0.05 to E-score, differences in the amount of energy a person expended was the easiest way to tell the difference in enjoyment levels. This means that people who were expending more energy, or people that were moving more, were enjoying the game more than people who were moving less.

5.3 Final Model

Our most effective prediction model was created using the Naive Bayes classifiers, with two prediction bins of 0 to 40 and 41 to 80. There was an equal number of sessions in each bin via sampling and only included the statistically significant features of energy in 0.5hz to 3hz, windowed energy in 0.5hz to 3hz, and radioSpectralPeak_DCT. To populate our training set for the 41-80 bin, we took samples of 12 sessions that had E-scores in the 41-80 bin in order to

match the number of sessions from the first bin.

The prediction model was more accurate when we had an equal number of sessions in each bin. Having bins with an equal number of sessions was more effective because it prevented the case where the model would predict the bin with the most sessions, causing prediction errors. Since the average E-score of our participants was above the cutoff E-score cutoff of 40 for enjoyment, the model was less accurate because it would almost always guess that people enjoyed the game purely because a majority of the people did. This is reflected in the confusion matrix and the lower Kappa statistic and ROC values from the models with the equal sized bins seen in Table 4.5.1.

Another factor that affected prediction accuracy the most was the range of each bin. The prediction model was more accurate when we used bins of equal ranges around the neutral value compared to bins of unequal ranges. For example, having 2 bins ranging from 0 to 40 and 41 to 80 performed better than having bins ranging from 0 to 59 and 60 to 80. To be precise, the best classifier from the 2 bins with equal ranges had 75% correct classification, while the bins with unequal ranges maxed out at 65% correct classification. This is likely due to the fact that the neutral value for enjoying the game was 40, which makes for a better split than having the split be at 59.

Examining our results, we saw that when the prediction model was created with data from the song that all participants played (the first song), it performed better than the prediction model that used all of the data from both songs[1], as can be seen in Figure 4.5.1. the classifier that was built with only first song data had a correct classification percentage of 69%, while the best both songs classifier had 65% correct classification. This is because when the model was created with a specific song, it would be better at predicting player enjoyment with that specific song. However, since building a prediction model on just one song does not lead to a

[1] Note that we are talking about the models where sampling was not involved

generalizable model, we chose to use the prediction model that was built with data from all of the songs that users selected.

5.4 Overall

In general, it is feasible to create a model that can predict user enjoyment of Just Dance Now from their smartphone sensor data reasonably well. Our final model was able to correctly classify enjoyment 75% of the time for 2 bins (enjoy, not enjoy). However, our model is trained using very sparse data on subjects who did not enjoy the game. We believe that if we had more motion data from people who did not enjoy Just Dance Now, it is possible that the prediction model would have been able to perform even better.

5.5 Future Work

Although we found our project to be reasonably successful, there is a great deal of future work that could further improve the prediction model and offer useful context for the model's success.

- *More data on subjects that did not enjoy the exergame (Low e-Score):* As was previously discussed, our classifier would have likely performed better if we had more mobile sensor data of participants that did not enjoy the game. This means that when doing experiments regarding exergames, it is important to have a roughly equal percentage of users that enjoyed and did not enjoy the game. Because it is easier to build a successful prediction model when there is an equal number of people enjoying and not enjoying a game, we recommend that more games users are likely to enjoy less than Just Dance Now be used in future exergame experiments.

- *More Pokemon Go data:* When we tested our overall best prediction model on Pokemon Go data, we were unable to come to a decisive conclusion due to a lack of experimental data. Gathering more Pokemon Go data would allow for a proper test our prediction model and come to a conclusion on the feasibility of using our prediction model for Just Dance Now for other genres of exergames.
- *Comparison of our enjoyment classifier with human raters:* It would also be useful to measure the success rate of our prediction model in predicting player enjoyment against the success rate of another person predicting player enjoyment. This could be done by running more Just Dance Now dancing experiments and having a human rater present. The rater would observe an individual's dancing and guess if they were enjoying the game or not (guess their e-Score from preset ranges). The human guesser could base their guess on the player's facial expressions, dancing patterns and more. The human rater's success rate would then be compared to the prediction model's success rate to determine how useful the prediction model is.
- *Creating a model from different genres of exergames:* Considering that our project mainly prioritized building a prediction model with Just Dance Now, future experiments should focus more on building a model that can be generalized across different games, and games of different genres. This may require gathering different mobile sensor per genre, such as the touch sensor. Figure 5.5.1 and Figure 5.5.2 depict the difference in accelerometer readings from Just Dance Now and from Pokemon Go respectively. While some features of the games can be seen, the touch sensor could be much more indicative of user enjoyment when playing Pokemon Go than accelerometer.
- *Experiment with other feature selection algorithms:* Lastly, we recommend that future works continue to try out different combinations of features and create new features with new sensor data. One way to do so is to utilize Weka's Attribute Selection Filter feature, which

automatically chooses features to use for machine learning classification.

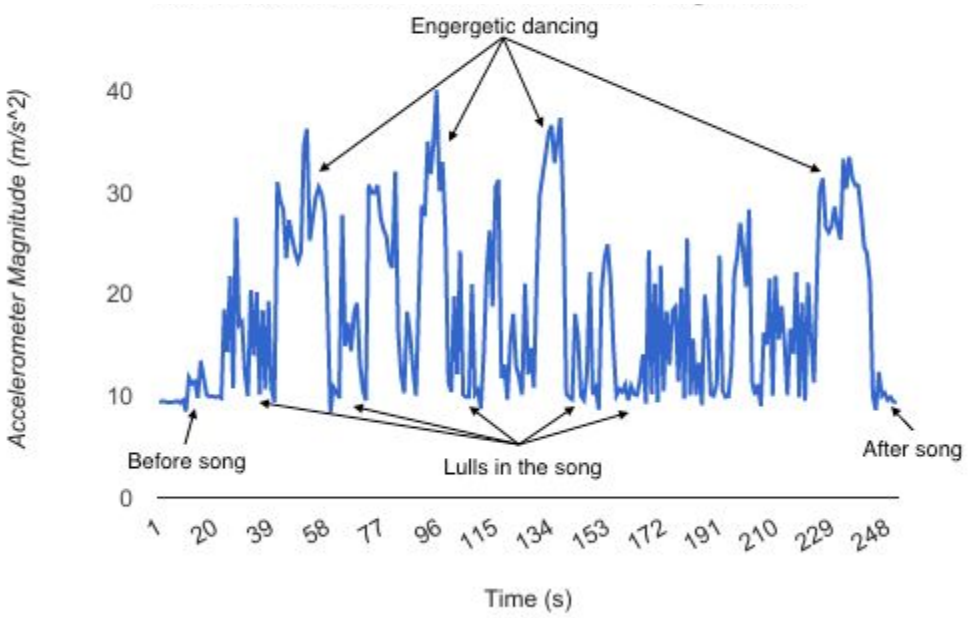

Figure 5.5.1: Just Dance Now Accelerometer Magnitude over time

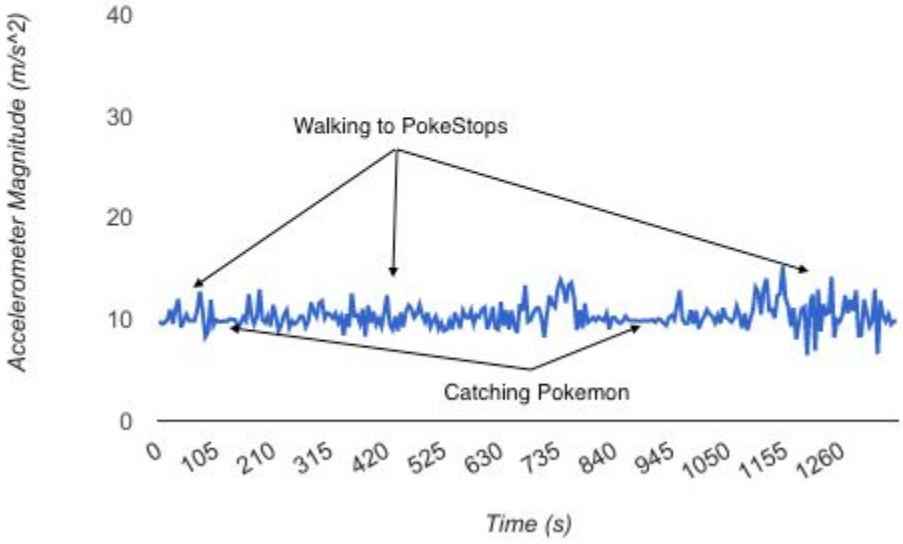

Figure 5.5.2: Pokemon Go Accelerometer Magnitude over time

References

1. Alpaydin, E. (2010). *Introduction to Machine Learning.* Cambridge, MA: MIT.
2. Agu, E., & Mark, C. *CyPRESS: A Cyber-Physical Recommender System to Discover SmartPhone Exergame Enjoyment.* Worcester, MA: WPI.
3. Aiello, Christina. (April, 2016). *Investigating Gyroscope Sway Features, Normalization, and Personalization in Detecting Intoxication in Smartphone Users.* Worcester, MA: WPI.
4. Arnold, Z., & LaRose, D. (2015). *Smartphone Gait Inference*. Worcester, MA: WPI.
5. Avouris, N., & Nikoleta Y. (2012). A Review of Mobile Location-based Games for Learning across Physical and Virtual Spaces*. Journal of Universal Computer Science 18(15)*, 2120-2142.
6. Bird, M., Clark, B., Millar J., Whetton, Sue., & Smith, S. (2015). Exposure to "Exergames" Increases Older Adults' Perception of the Usefulness of Technology for Improving Health and Physical Activity: A Pilot Study. *JMIR Serious Games, 3(2)*.
7. Brockmyer, J. (2009). The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, *45(4)*, 624-634.
8. Brownlee, J. (2016, March 16). Supervised and Unsupervised Machine Learning Algorithms - Machine Learning Mastery. Retrieved from http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
9. Cooper, J. (2015, January 15). Are Gamified Fitness Apps about to Change Your Life? Retrieved from http://www.telegraph.co.uk/men/active/11340652/Are-gamified-fitness-apps-about-to-cha

nge-your-life.html

10. Dems, Kristina. (2011, February 3). How Did Geocaching Start? Discover The Origins of Geocaching. Retrieved from http://www.brighthub.com/electronics/gps/articles/75678.aspx
11. Graves, L., Ridgers N., Williams, K., Stratton, G., Atkinson, G., & Cable, N. (2010). The Physiological Cost and Enjoyment of Wii Fit in Adolescents, Young Adults, and Older Adults. *Journal of Physical Activity and Health*, *7(3)*, 393-401.
12. Gregory, M. (2014, August 7). Ingress: A Game, Lifestyle and Social Network in One! Retrieved from http://www.wheninmanila.com/ingress-game-lifestyle-social-network/
13. Holland, C. (n.d) President's Council on Fitness, Sports & Nutrition. Facts & Statistics -. President's Council on Fitness, Sports & Nutrition. Retrieved from https://www.fitness.gov/resource-center/facts-and-statistics/
14. Javornik, A. (2016, October 4) The Mainstreaming of Augmented Reality: A Brief History. Retrieved from https://hbr.org/2016/10/the-mainstreaming-of-augmented-reality-a-brief-history
15. Kendzierski, D., & Kenneth J. (1991) Physical activity enjoyment scale: two validation studies." *Journal of Sport & Exercise Psychology, 13(2)*.
16. Kretschmann, R. (2010) Exergames and Health Promotion - Nintendo Wii Sports: Physiological Measures vs. Perceived Opinions, *VII Congress Int'l Assoc Colleges Phys*. *Ed.* Stuttgart, Germany :University of Stuttgart
17. Levine, P. (2015, January 22). Machine Learning + Big Data. Retrieved from http://a16z.com/2015/01/22/machine-learning-big-data/
18. Marr, B. (2016, February 19). A Short History of Machine Learning -- Every Manager Should Read. Retrieved from http://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning

-every-manager-should-read/#46f76c04323f

19. Mokdad, A., Marks, J., Stroup D, & Gerberding I. (2004). Actual causes of death in the United States, 2000. *Jama, 291(10)*, 1238-1245.
20. Moses, T. (2012, March 24) Zombies, Run! – Review. Retrieved from https://www.theguardian.com/technology/2012/mar/25/zombies-run-naomi-alderman-app
21. Motl, R. M., Dishman, R. K., Saunders, R., Dowda, M., Felton, G., & Pate, R. R. (2001) Measuring enjoyment of physical activity in adolescent girls. *American journal of preventive medicine, 21(2)*, 110-117.
22. Munoz, A. (n.d) Machine Learning and Optimization. *Courant Institute of Mathematical Sciences*. New York, NY: NYU.
23. Qi, Muxi. (April, 2016). *A Comprehensive Comparative Performance Evaluation of Signal Processing Features in Detecting Alcohol Consumption from Gait Data*. Worcester, MA: WPI.
24. Rubino, D. (2014, November 14). Ubisoft's Shape Up Battle Run Launches for Windows Phone to Make Sprinting Fun. Retrieved from http://www.windowscentral.com/ubisofts-shape-battle-run-launches-windows-phone
25. Rachuri, K. K., Mirco M., Cecilia M., Peter J. R., Chris L., & Andrius A. (2010) EmotionSense. *Proceedings of the 12th ACM International Conference on Ubiquitous Computing - Ubicomp '10.* New York, NY: ACM.
26. Raedeke, T. D. (2007) The Relationship Between Enjoyment and Affective Responses to Exercise. *Journal of Applied Sport Psychology, 19(1)*, 105-15.
27. Samuel, A. L. (1959) Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 3(3)*, 210-29.
28. Silva, A, & De Souza E. (2006) Hybrid Reality Games Reframed: Potential Uses in Educational Contexts. *Games and Culture 1(3)*, 231-51.

29. Simon, S. (March 18, 2013). Too Big to Ignore: The Business Case for Big Data. MA: Wiley.
30. Wang, X., & Arlette C. P. (2006) Metabolic and physiologic responses to video game play in 7-to 10-year-old boys. *Archives of Pediatrics & adolescent medicine, 160(4),* 411-415.
31. Williams, W. (2014) The Story-driven Superhero Workout App Makes Getting Fit Super-fun. Retrieved from https://betanews.com/2014/08/22/the-story-driven-superhero-workout-app-makes-getting-fit-super-fun/
32. Witten, I. H., & Eibe F. (2005). Data Mining: Practical Machine Learning Tools and Techniques." Amsterdam, Netherlands: Morgan Kaufman.
33. (2015). Androsensor. Retrieved from https://play.google.com/store/apps/details?id=com.fivasim.androsensor&hl=en

Appendices

Appendix A - Game Experience Questionnaire

This shows the Game Experience Questionnaire, or GEQ, which is one method for determining player enjoyment. We chose to use a different questionnaire to create baseline E-scores, which can be seen in Appendix B.

Game Experience Questionnaire – Core Module

Please indicate how you felt while playing the game for each of the items, on the following scale:

not at all	slightly		moderately	fairly	extremely
0	1	2	3	4	

1	I felt content
2	I felt skillful
3	I was interested in the game's story
4	I thought it was fun
5	I was fully occupied with the game
6	I felt happy
7	It gave me a bad mood
8	I thought about other things
9	I found it tiresome
10	I felt competent
11	I thought it was hard
12	It was aesthetically pleasing
13	I forgot everything around me
14	I felt good
15	I was good at it
16	I felt bored

17 I felt successful
18 I felt imaginative
19 I felt that I could explore things
20 I enjoyed it
21 I was fast at reaching the game's targets
22 I felt annoyed
23 I felt pressured
24 I felt irritable
25 I lost track of time
26 I felt challenged
27 I found it impressive
28 I was deeply concentrated in the game
29 I felt frustrated
30 It felt like a rich experience
31 I lost connection with the outside world
32 I felt time pressure
33 I had to put a lot of effort into it

In-game GEQ

Please indicate how you felt while playing the game for each of the items, on the following scale:

not at all	slightly	moderately	fairly	extremely
0	1	2	3	4

1 I was interested in the game's story GEQ Core – 3
2 I felt successful GEQ Core – 17
3 I felt bored GEQ Core – 16
4 I found it impressive GEQ Core – 27
5 I forgot everything around me GEQ Core – 13
6 I felt frustrated GEQ Core – 29
7 I found it tiresome GEQ Core – 9
8 I felt irritable GEQ Core – 24
9 I felt skillful GEQ Core – 2
10 I felt completely absorbed GEQ Core – 5
11 I felt content GEQ Core – 1
12 I felt challenged GEQ Core – 26
13 I had to put a lot of effort into it GEQ Core – 33
14 I felt good GEQ Core – 14

GEQ - Social Presence Module

Please indicate how you felt while playing the game for each of the items, on the following scale:

not at all	slightly		moderately	fairly	extremely
0	1	2	3	4	

1	I empathized with the other(s)
2	My actions depended on the other(s) actions
3	The other's actions were dependent on my actions
4	I felt connected to the other(s)
5	The other(s) paid close attention to me
6	I paid close attention to the other(s)
7	I felt jealous about the other(s)
8	I found it enjoyable to be with the other(s)
9	When I was happy, the other(s) was(were) happy
10	When the other(s) was(were) happy, I was happy
11	I influenced the mood of the other(s)
12	I was influenced by the other(s) moods
13	I admired the other(s)
14	What the other(s) did affected what I did
15	What I did affected what the other(s) did
16	I felt revengeful
17	I felt schadenfreude (malicious delight)

GEQ – post-game module

Please indicate how you felt after you finished playing the game for each of the items, on the following scale:

not at all	slightly		moderately	fairly	extremely
0	1	2	3	4	

1	I felt revived
2	I felt bad
3	I found it hard to get back to reality
4	I felt guilty
5	It felt like a victory
6	I found it a waste of time
7	I felt energised
8	I felt satisfied

9 I felt disoriented
10 I felt exhausted
11 I felt that I could have done more useful things
12 I felt powerful
13 I felt weary
14 I felt regret
15 I felt ashamed
16 I felt proud
17 I had a sense that I had returned from a journey

Scoring guidelines

Scoring guidelines GEQ Core Module
The Core GEQ Module consists of seven components; the items for each are listed below.

Component scores are computed as the average value of its items.
Competence: Items 2, 10, 15, 17, and 21.
Sensory and Imaginative Immersion: Items 3, 12, 18, 19, 27, and 30.
Flow: Items 5, 13, 25, 28, and 31.
Tension/Annoyance: Items 22, 24, and 29.
Challenge: Items 11, 23, 26, 32, and 33.
Negative affect: Items 7, 8, 9, and 16.
Positive affect: Items 1, 4, 6, 14, and 20.

Scoring guidelines GEQ In-Game version
The In-game Module consists of seven components, identical to the core Module. However, only two items are used for every component. The items for each are listed below.
Component scores are computed as the average value of its items.
Competence: Items 2 and 9.
Sensory and Imaginative Immersion: Items 1 and 4.
Flow: Items 5 and 10.
Tension: Items 6 and 8.
Challenge: Items 12 and 13.
Negative affect: Items 3 and 7.
Positive affect: Items 11 and 14.

Scoring guidelines GEQ Social Presence Module
The Social Presence Module consists of three components; the items for each are listed below.
Component scores are computed as the average value of its items.
Psychological Involvement – Empathy: Items 1, 4, 8, 9, 10, and 13.
Psychological Involvement – Negative Feelings: Items 7, 11, 12, 16, and 17.
Behavioural Involvement: Items 2, 3, 5, 6, 14, and 15.

Scoring guidelines GEQ Post-game Module
The post-game Module consists of four components; the items for each are listed below.
Component scores are computed as the average value of its items.
Positive Experience: Items 1, 5, 7, 8, 12, 16.
Negative experience: Items 2, 4, 6, 11, 14, 15.
Tiredness: Items 10, 13.
Returning to Reality: Items 3, 9, and 17.

Appendix B - Immersive Experience Questionnaire

This appendix shows the questionnaire that we administered to see if participants actually enjoyed playing the game. The answers that participants submitted were then converted in each individual's E-score that would be used to build our models.

Version 1

Your personal experience of the game

Please rate how far you would agree with the statements below just before you were interrupted.

SD=strongly disagree; D=disagree; N=neutral; A=agree; SA=strongly agree.

I felt that I really empathised/felt for with the game.									
SD	D	N	A	SA					
I did not feel any emotional attachment to the game.									

SD	D	N	A	SA					
I was interested in seeing how the game's events would progress.									
SD	D	N	A	SA					
It did not interest me to know what would happen next in the game.									
SD	D	N	A	SA					
I was in suspense about whether I would win or lose the game.									
SD	D	N	A	SA					
I was not concerned about whether I would win or lose the game.									
SD	D	N	A	SA					
I sometimes found myself to become so involved with the game that I wanted to speak to the game directly.									
SD	D	N	A	SA					
I did not find myself to become so caught up with the game that I wanted to speak to directly to the game.									
SD	D	N	A	SA					
I enjoyed the graphics and imagery of the game.									
SD	D	N	A	SA					
I did not like the graphics and imagery of the game.									
SD	D	N	A	SA					
I enjoyed playing the game.									
SD	D	N	A	SA					
Playing the game was not fun.									
SD	D	N	A	SA					
The controls were not easy to pick up.									

SD	D	N	A	SA					
There were not any particularly frustrating aspects of the controls to get the hang of.									
SD	D	N	A	SA					
I became unaware that I was even using any controls.									
SD	D	N	A	SA					
The controls were not invisible to me.									
SD	D	N	A	SA					
I felt myself to be directly travelling through the game according to my own volition.									
SD	D	N	A	SA					
I did not feel as if I was moving through the game according to my own will.									
SD	D	N	A	SA					
It was as if I could interact with the world of the game as if I was in the real world.									
SD	D	N	A	SA					
Interacting with the world of the game did not feel as real to me as it would be in the real world.									
SD	D	N	A	SA					
I was unaware of what was happening around me.									
SD	D	N	A	SA					
I was aware of surroundings.									
SD	D	N	A	SA					
I felt detached from the outside world.									
SD	D	N	A	SA					
I still felt attached to the real world.									
SD	D	N	A	SA					

At the time the game was my only concern.									
SD	D	N	A	SA					
Everyday thoughts and concerns were still very much on my mind.									
SD	D	N	A	SA					
I did not feel the urge at any point to stop playing and see what was going on around me.									
SD	D	N	A	SA					
I was interested to know what might be happening around me.									
SD	D	N	A	SA					
I did not feel like I was in the real world but the game world.									
SD	D	N	A	SA					
I still felt as if I was in the real world whilst playing.									
SD	D	N	A	SA					
To me it felt like only a very short amount of time had passed.									
SD	D	N	A	SA					
When playing the game time appeared to go by very slowly.									
SD	D	N	A	SA					
How immersed did you feel? (10=very immersed; 0=not at all immersed)									
1	2	3	4	5	6	7	8	9	10

Version 2

Your experience of the game

Please answer the following questions by circling the relevant number. In particular, remember that these questions are asking you about how you felt at the end of the game.

To what extent did the game hold your attention?						
Not at all	1	2	3	4	5	A lot
To what extent did you feel you were focused on the game?						
Not at all	1	2	3	4	5	A lot
How much effort did you put into playing the game?						
Very little	1	2	3	4	5	A lot
Did you feel that you were trying you best?						
Not at all	1	2	3	4	5	Very much so
To what extent did you lose track of time?						
Not at all	1	2	3	4	5	A lot
To what extent did you feel consciously aware of being in the real world whilst playing?						
Not at all	1	2	3	4	5	Very much so
To what extent did you forget about your everyday concerns?						
Not at all	1	2	3	4	5	A lot
To what extent were you aware of yourself in your surroundings?						
Not at all	1	2	3	4	5	Very aware
To what extent did you notice events taking place around you?						
Not at all	1	2	3	4	5	A lot
Did you feel the urge at any point to stop playing and see what was happening around you?						
Not at all	1	2	3	4	5	Very much so
To what extent did you feel that you were interacting with the game environment?						
Not at all	1	2	3	4	5	Very much so
To what extent did you feel as though you were separated from your real-world environment?						
Not at all	1	2	3	4	5	Very much so

To what extent did you feel that the game was something you were experiencing, rather than something you were just doing?						
Not at all	1	2	3	4	5	Very much so
To what extent was your sense of being in the game environment stronger than your sense of being in the real world?						
Not at all	1	2	3	4	5	Very much so
At any point did you find yourself become so involved that you were unaware you were even using controls?						
Not at all	1	2	3	4	5	Very much so
To what extent did you feel as though you were moving through the game according to you own will?						
Not at all	1	2	3	4	5	Very much so
To what extent did you find the game challenging?						
Not at all	1	2	3	4	5	Very difficult
Were there any times during the game in which you just wanted to give up?						
Not at all	1	2	3	4	5	A lot
To what extent did you feel motivated while playing?						
Not at all	1	2	3	4	5	A lot
To what extent did you find the game easy?						
Not at all	1	2	3	4	5	Very much so
To what extent did you feel like you were making progress towards the end of the game?						
Not at all	1	2	3	4	5	A lot
How well do you think you performed in the game?						
Very poor	1	2	3	4	5	Very well
To what extent did you feel emotionally attached to the game?						
Not at all	1	2	3	4	5	Very much so

To what extent were you interested in seeing how the game's events would progress?						
Not at all	1	2	3	4	5	A lot
How much did you want to “win” the game?						
Not at all	1	2	3	4	5	Very much so
Were you in suspense about whether or not you would win or lose the game?						
Not at all	1	2	3	4	5	Very much so
At any point did you find yourself become so involved that you wanted to speak to the game directly?						
Not at all	1	2	3	4	5	Very much so
To what extent did you enjoy the graphics and the imagery?						
Not at all	1	2	3	4	5	A lot
How much would you say you enjoyed playing the game?						
Not at all	1	2	3	4	5	A lot
When interrupted, were you disappointed that the game was over?						
Not at all	1	2	3	4	5	Very much so
Would you like to play the game again?						
Definitely not	1	2	3	4	5	Definitely yes

Appendix C - Just Dance Now Surveys

This appendix shows the full Just Dance Now questionnaire. This shows the demographic section as well as the section in Appendix B.

Pre Exergame Questionnaire

* Required

1. **Participant Number** *

2. **Gender** *

Mark only one oval.

- ◯ Male
- ◯ Female
- ◯ Prefer not to say
- ◯ Other:

3. **Height? (ex. 5 foot 9 inches or 5' 9")** *

4. **Weight** *

Mark only one oval.

- ◯ Less than 100 lbs
- ◯ 100-125 lbs
- ◯ 126-150 lbs
- ◯ 151-175 lbs
- ◯ 176-200 lbs
- ◯ 201-225 lbs
- ◯ 226-250
- ◯ More than 250 lbs
- ◯ Prefer not to say

5. **Age** *

6. **Do you know what an exergame is?** *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

7. **Have you heard of or played either of the games: Just Dance Now or Pokémon Go?** *
Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

8. **What, if any, mobile games have you played?** *

9. **How many hours a week would you say you engage in physical activity?** *

10. **How many hours a week would you say that you play exergames?** *

Just Dance Now Game Questionnaire (for Taste the Feeling)

11. **For each of the statements below, please choose how much you agree or disagree with the statement.** *

Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I felt excited about the physical activities in the game.					
The exercise in this game made me feel good.					
I felt like I lost track of time while playing.					
I felt that it was difficult to understand how the game works.					
I was focused on the game.					
I felt that the game would have been more enjoyable without physical activity.					
I felt that it was easy to familiarize myself with the game controls.					
I felt emotionally attached to the game.					
I consider playing the game "exercise".					
I felt that the physical activity was too intense for me.					
I did not feel a desire to make progress in the game.					
I felt a strong sense of being in the world of the game to the point that I was unaware of my surroundings.					
I would rather not be exercising, even though the exercise was "gamified".					
I felt that I benefitted from playing the game.					
I cared about winning in the game.					
I felt that this game provided an enjoyable challenge.					
I felt a sense of accomplishment from playing the game.					
I felt that the game was responsive.					
I did not feel like I wanted to keep playing.					
I would prefer that this physical activity was not "gamified".					
I felt in control of the game.					

Just Dance Now Game Questionnaire (participant's choice)

12. **Song Title** *

13. **For each of the statements below, please choose how much you agree or disagree with the statement.** *

Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I felt excited about the physical activities in the game.	◯	◯	◯	◯	◯
The exercise in this game made me feel good.	◯	◯	◯	◯	◯
I felt like I lost track of time while playing.	◯	◯	◯	◯	◯
I felt that it was difficult to understand how the game works.	◯	◯	◯	◯	◯
I was focused on the game.	◯	◯	◯	◯	◯
I felt that the game would have been more enjoyable without physical activity.	◯	◯	◯	◯	◯
I felt that it was easy to familiarize myself with the game controls.	◯	◯	◯	◯	◯
I felt emotionally attached to the game.	◯	◯	◯	◯	◯
I consider playing the game “exercise”.	◯	◯	◯	◯	◯
I felt that the physical activity was too intense for me.	◯	◯	◯	◯	◯
I did not feel a desire to make progress in the game.	◯	◯	◯	◯	◯
I felt a strong sense of being in the world of the game to the point that I was unaware of my surroundings.	◯	◯	◯	◯	◯
I would rather not be exercising, even though the exercise was “gamified”.	◯	◯	◯	◯	◯
I felt that I benefitted from playing the game.	◯	◯	◯	◯	◯
I cared about winning in the game.	◯	◯	◯	◯	◯
I felt that this game provided an enjoyable challenge.	◯	◯	◯	◯	◯
I felt a sense of accomplishment from playing the game.	◯	◯	◯	◯	◯
I felt that the game was responsive.	◯	◯	◯	◯	◯
I did not feel like I wanted to keep playing.	◯	◯	◯	◯	◯
I would prefer that this physical activity was not “gamified”.	◯	◯	◯	◯	◯
I felt in control of the game.	◯	◯	◯	◯	◯

Post Exergame Questionnaire

14. **Did you enjoy this game?** *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Other: .....

15. **What goals or achievements did you accomplish in the game?** *

.....

.....

.....

.....

Powered by
Google Forms

Appendix D - Weka Data

This appendix shows important Weka output data for all prediction models created, including correctly classified instances, kappa statistic, and ROC Area weighted average.

Weka Results

All Songs 2 Buckets All Features 59

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	57.0%	56.0%	62.0%	58.0%
Kappa Statistic	0.137	0.120	0.235	0.157
ROC Area Weighted Average	0.622	0.567	0.617	0.562

All Songs 2 Buckets 3 Features 59 (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes

Correctly Classified Instances	52.0%	65.0%	65.0%	62.0%
Kappa Statistic	0.040	0.293	0.292	0.232
ROC Area Weighted Average	0.546	0.613	0.644	0.599

All Songs 2 Buckets 13 Features 59 (Principal Components attribute selection, Ranker search)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	55.0%	51.0%	56.0%	52.0%
Kappa Statistic	0.098	0.011	0.115	0.043
ROC Area Weighted Average	0.548	0.522	0.557	0.486

All Songs 2 Buckets All Features 40

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	86.0%	85.0%	88.0%	53.0%
Kappa Statistic	-0.036	0.204	0.000	0.074
ROC Area Weighted Average	0.714	0.665	0.500	0.699

All Songs 2 Buckets 3 Features 40 (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	86.0%	88.0%	88.0%	59.0%
Kappa Statistic	-0.036	0.000	0.000	0.167
ROC Area	0.406	0.424	0.500	0.679

Weighted Average				

All Songs 3 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	35.0%	36.0%	36.0%	36.0%
Kappa Statistic	0.024	0.040	0.038	0.038
ROC Area Weighted Average	0.529	0.537	0.502	0.545

All Songs 3 Buckets 3 Features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified	42.0%	39.0%	42.0%	46.0%

Instances				
Kappa Statistic	0.129	0.080	0.124	0.186
ROC Area Weighted Average	0.576	0.501	0.562	0.591

All Songs 6 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	17.0%	24.0%	19.0%	16.0%
Kappa Statistic	0.001	0.087	0.024	-0.012
ROC Area Weighted Average	0.597	0.520	0.540	0.537

All Songs 6 Buckets 3 Features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random	J48	SMO	Naïve Bayes

	Forest			
Correctly Classified Instances	22.0%	15.0%	16.0%	25.0%
Kappa Statistic	0.062	-0.022	-0.015	0.098
ROC Area Weighted Average	0.514	0.511	0.496	0.526

All Songs 10 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	10.0%	20.0%	19.0%	19.0%
Kappa Statistic	-0.006	0.109	0.088	0.099
ROC Area Weighted Average	0.551	0.546	0.547	0.612

All Songs 10 Buckets 3 Features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	09.0%	13.0%	11.0%	17.0%
Kappa Statistic	-0.016	0.031	-0.010	0.072
ROC Area Weighted Average	0.431	0.481	0.476	0.523

First Songs 2 Buckets All Features 59

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	59.2%	53.1%	65.3%	69.4%
Kappa Statistic	0.183	0.059	0.30	0.389
ROC Area	0.648	0.497	0.651	0.677

Weighted Average				

First Songs 2 Buckets 3 Features 59 (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	59.2%	51.0%	57.1%	59.2%
Kappa Statistic	0.182	0.025	0.153	0.191
ROC Area Weighted Average	0.628	0.578	0.466	0.602

First Songs 2 Buckets 1 Feature 59 (radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified	51.1%	55.1%	65.3%	65.3%

Instances				
Kappa Statistic	0.145	0.109	0.316	0.310
ROC Area Weighted Average	0.611	0.510	0.660	0.677

First Songs 2 Buckets All Feature 40

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	83.7%	81.6%	85.7%	63.3%
Kappa Statistic	-0.037	0.203	0.000	0.060
ROC Area Weighted Average	0.570	0.500	0.500	0.553

First Songs 2 Buckets 3 Feature 40 (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random	J48	SMO	Naïve Bayes

	Forest			
Correctly Classified Instances	85.7%	85.7%	85.7%	59.2%
Kappa Statistic	0.170	0.000	0.000	0.091
ROC Area Weighted Average	0.384	0.333	0.500	0.616

First Songs 2 Buckets 1 Feature 40 (radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	75.5%	85.7%	85.7%	85.7%
Kappa Statistic	0.000	0.000	0.000	0.000
ROC Area Weighted Average	0.468	0.333	0.500	0.498

First Songs 3 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	30.6%	34.7%	26.5%	30.6%
Kappa Statistic	-0.048	0.008	-0.129	-0.050
ROC Area Weighted Average	0.459	0.500	0.448	0.507

First Songs 3 Buckets 3 Features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	42.9%	32.7%	32.7%	42.86%
Kappa Statistic	0.132	-0.031	-0.051	0.129
ROC Area	0.553	0.459	0.507	0.563

Weighted Average				

First Songs 3 Buckets 1 Features (radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	24.5%	34.7%	36.7%	51.0%
Kappa Statistic	-0.137	0.005	0.007	0.246
ROC Area Weighted Average	0.376	0.408	0.518	0.585

First Songs 6 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	30.6%	18.37%	20.4%	16.3%

Kappa Statistic	0.162	0.014	0.027	-0.008
ROC Area Weighted Average	0.565	0.529	0.558	0.517

First Songs 6 Buckets 3 Features (radioSpectralPeak, radioSpectralPeak_FFT, radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	24.5%	20.4%	18.4%	12.2%
Kappa Statistic	0.094	0.050	-0.020	-0.064
ROC Area Weighted Average	0.587	0.546	0.467	0.429

First Songs 6 Buckets 1 Feature (radioSpectralPeak_DCT)

	Random Forest	J48	SMO	Naïve Bayes

Correctly Classified Instances	12.2%	16.3%	18.4%	24.5%
Kappa Statistic	-0.056	-0.018	-0.019	0.079
ROC Area Weighted Average	0.465	0.503	0.432	0.478

All Songs Even Split 40 2 Buckets All Features

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	54.2%	50.0%	62.5%	62.5%
Kappa Statistic	0.083	0.000	0.250	0.250
ROC Area Weighted Average	0.517	0.483	0.625	0.624

All Songs Even Split 40 2 Buckets 3 Features (energy in _5 to 3, windowed energy in _5

to 3, and radioSpectralPeak_DCT). *Note that we added radioSpectralPeak_DCT because it had a P-value of 0.0502, which is close enough to having a P-value below 0.05

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	66.7%	70.8%	66.7%	75.0%
Kappa Statistic	0.333	0.417	0.333	0.500
ROC Area Weighted Average	0.715	0.590	0.667	0.590

All Songs Even Split 40 2 Buckets 2 Features (windowed energy in _5 to 3, energy in _5 to 3)

	Random Forest	J48	SMO	Naïve Bayes
Correctly Classified Instances	66.7%	70.8%	50.0%	70.8%

Kappa Statistic	0.333	0.417	0.000	0.417
ROC Area Weighted Average	0.708	0.590	0.500	0.556