Management Science & Technology
ATHENS UNIVERSITY OF ECONOMICS & BUSINESS

Emotions as implicit ratings through music stimuli: Synchronization of preferences and physiological responses

Kolias Evangelos
Student of the MSc in Management Science and Technology
Athens University of Economics and Business

ACKNOWLEDGEMENT

I wish to sincerely thank Dr. Lekakos Georgios, Professor and Director of the MSc in Management Science and Technology of Athens University of Economics and Business, for the opportunity he gave me to work in a thesis regarding emotion response to music stimuli, as well as his helpful comments and appreciable assistance towards the whole study. Additionally, I wish to thank Marianna Skiada, PhD Candidate of Management Science and Technology Department of Athens University of Economics and Business, as with the her cooperation, I was able to complete the present study. Her assistance during the various experiments and research this study required, was of primary importance.

TABLE OF CONTENTS

Figures

Appendices

ABSTRACT

The purpose of this study is to examine through a listening experiment and relevant theoretical approaches, whether emotions, evoked by music stimuli, can be considered as implicit rating data. Five subjects participated on the experiment in which, ten music excerpts of different valence were played to each individual separately. They had to rate the music they listened to, so as to provide us with their preferences towards each excerpt. Furthermore, the subjects while listening to music, had also their physiological measurements of galvanic skin response (GSR), skin temperature (ST) and beats per minute (BPM) recorded, through specific sensors implemented on a wristband. These particular measures were chosen because of the fact that, as stated in other studies, they have a strong relationship with the valence and arousal of an emotion and consequently can be used to define it. According to this research, GSR and ST, indeed present a strong correlation with the valence and arousal of subjects' emotions, as well as with their recorded preferences. Precisely, subjects' GSR and ST values were quite high during the listening of music they did not like, as opposed to music they liked. Additionally, the calculated correlation between the mentioned measures and ratings, for each individual separately, was negative and strong. Finally, various machine learning algorithms were applied to further investigate the mentioned correlation and make predictions about subjects' ratings. Although our experiment lacks the appropriate quantity, as the number of participants is low, it is sufficient to provide us with qualitative indications regarding the relationship between individuals' recorded measures and preferences in form of ratings.

Keywords: Emotions, music stimuli, GSR, ST, implicit ratings, recommender systems

INTRODUCTION

Recommender Systems

Over the years, through technology's rapid advance, new forms of social communication and business proceedings have been emerged and dominated every day transactions. The vast use of web services, as well as the constantly increasing availability of gathered information, have rendered e-commerce as one of the most profitable online services. Nowadays, markets need to provide specific products and services to specific customers with specific needs. By acquiring individuals' preferences and opinions, producers can successfully produce more customized products, for more customers. This personalization of products or services, depending on customers' preferences, has been made possible through the recommender systems (RS) (Burke et al., 2011). The most popular recommender algorithms are the Content-Based, which recommends new items based on previous items the user has bought or evaluated, the Collaborative Filtering, which recommends items based on other users' preferences, whose previous evaluations match with those of the target user and the Hybrids, which are a combination of the other two, improving that way the prediction performance.

Explicit vs Implicit Ratings

The preferences, or evaluations of an item, are been attained from the users in the form of ratings, made on a predetermined discrete scale, although, sometimes they are made also in the form of a review. These, are known as explicit ratings and their main characteristic is that the user has to examine the item and personally assign it a rating value. This procedure is costly for the evaluator, as he needs to spent time and focus and as a result he does not always give a rating. As a consequence, the problem of rating data sparsity is emerged, by which, the available ratings are insufficient for the algorithm to recommend similar items or users with the same preferences. Sometimes, even if a user gives ratings on a regular basis, the product quantity of an e-commerce site is so vast, that sparsity problems still reduce system's prediction efficiency. Another sparsity problem frequently occurs, is cold start. According to this, when a new user or item has been added into the database, it is difficult to find similar users because there is not enough information gathered yet. Of course, to tackle both data sparsity and cold start problem, several other approaches and methods have been developed, providing recommendation solutions even when sufficient ratings are unavailable. However, insufficient evaluation issues still occur to some extent (Sachan and Richhariya, 2013;

Huang and Yin, 2010; Papagelis et al., 2005). According to Nichols (1998), implicit ratings, namely, ratings gathered from the observation of user behaviour or by using other technologies, may be another solution to the sparsity problem. In this case, the evaluation cost of the users is eliminated, as it would be unnecessary to rate the items themselves.

Emotions as Implicit Data

Potentially, as technology means advance, generation of implicit data will be an easier task. IoT sensor gadgets, in combination with real-time information gathering through data streams, will provide recommender algorithms with enough data to increase their prediction efficiency. Even if these type of data contain less value than a self made rate evaluation, their quantity will be big enough to replace it sufficiently, as they will be generated in every user interaction with a system. What is remained to be answered is whether or not emotions can be detected from a technology gadget and play the part of these data streams providing us with enough information to predict an individual's rating. In other words, whether emotions can be considered as implicit data, capable of being transformed into ratings, with enough value to describe users' preferences towards an item. As that item, in this study we chose music because, as stated by several studies, it is a quite suitable mediator for evoking emotions, but we will discuss that more thoroughly in the following paragraphs.

Emotions

First, we should define what emotions are. To begin with, although many researchers had tried to give a proper answer to the subject, currently there is no scientific consensus on a definition. According to Scherer's (2009) component process model, emotions are processes of unintentionally linked mental and behavioral elements, while Reisenzein (2007) described them as mental states. In another study, Lindquist et al. (2012), defined a psychological constructionist model according to which, emotions are ''situated conceptualizations'' as they are evoked from the personalized meaning each individual give to both the environmental and body inputs. Specifically, individuals translate every internal or external stimulus, according to previous experiences, as well as the situation they are currently faced with (Lindquist et al., 2012). Whatever the case is, we can conclude that emotions are responsible for the various changes in our mood, behaviour and psychological state.

Measuring Emotions

Secondly, a proper way of measuring emotions should also be discussed, as it is of primary importance for the progression of our study. A recent review of studies regarding emotions induced through music, revealed that the two most used emotion models are the discrete emotion model and the two dimensional model (Eerola and Vuoskoski, 2011).

As far as the discrete emotion model (or basic emotion model) is concerned, all emotions can be summarized to the instinctive basic ones: happiness, sadness, fear, surprise, anger, and disgust (Kowalska and Wróbel, 2017). Also, it should be noted that some of the basic emotions, cannot be expressed while listening to music and for that reason discrete model very often needs to be modified accordingly, to better describe the emotions that are commonly represented by music. For example, disgust is often changed to tenderness or peacefulness in order to reflect listeners' emotions (Eerola and Vuoskoski, 2011). Finally, this particular model supports that each basic emotion arises from an independent neural system (Barrett and Wager, 2006).

On the other hand, the two dimensional emotion model suggests that, instead of an independent neural system, every emotion is induced from two discrete physiological and psychological states: the valence and the arousal. The first alternates between the attractiveness (a positive feeling) and the averseness (a negative feeling), while the second is related to the intensity with which you feel each aspect of valence. In other words, all emotions can be understood as a mixture of those two physiological dimensions (Eerola and Vuoskoski, 2011).

Discrete vs Two-Dimensional Model

Russel et al. (2005), proposes that all emotions can be described as varying degrees of valence (depicted as an unpleasant-pleasant scale) and arousal (depicted as a deactivation-activation scale). More precisely, when an individual defines the exact amount of valance and arousal he feels (for example in a scale from 0 to 10), a vector with these coordinates will be formed, describing a specific emotion. Such vector is presented in the figure bellow:

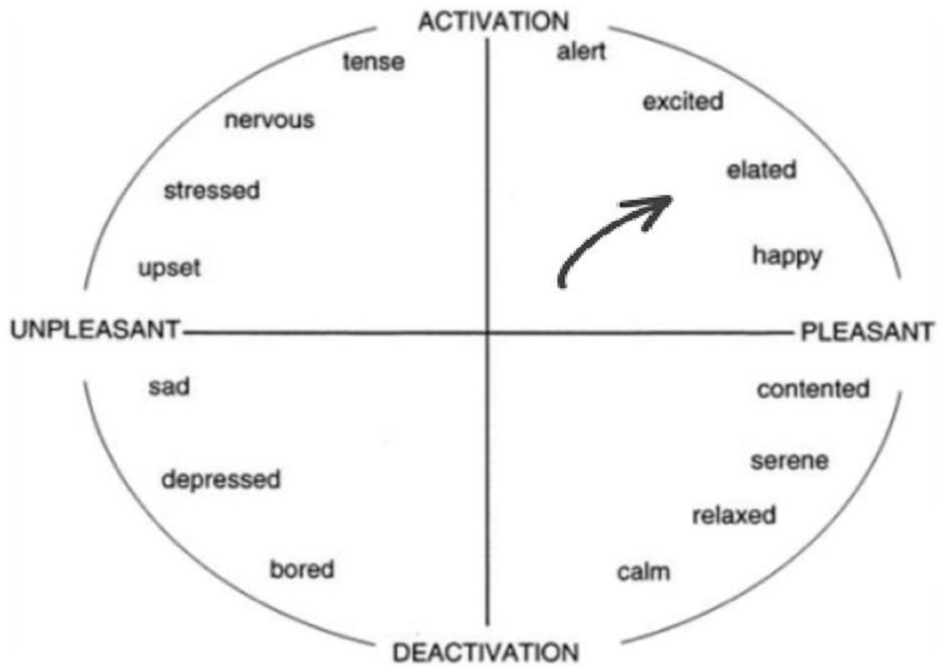


Figure 1: Russel's Dimensional Emotion Model

However, as shown in the above figure, a major drawback of this dimensional model is the lack of differentiation when emotion vectors have very close Cartesian coordinates. That situation is depicted better in the next figure.

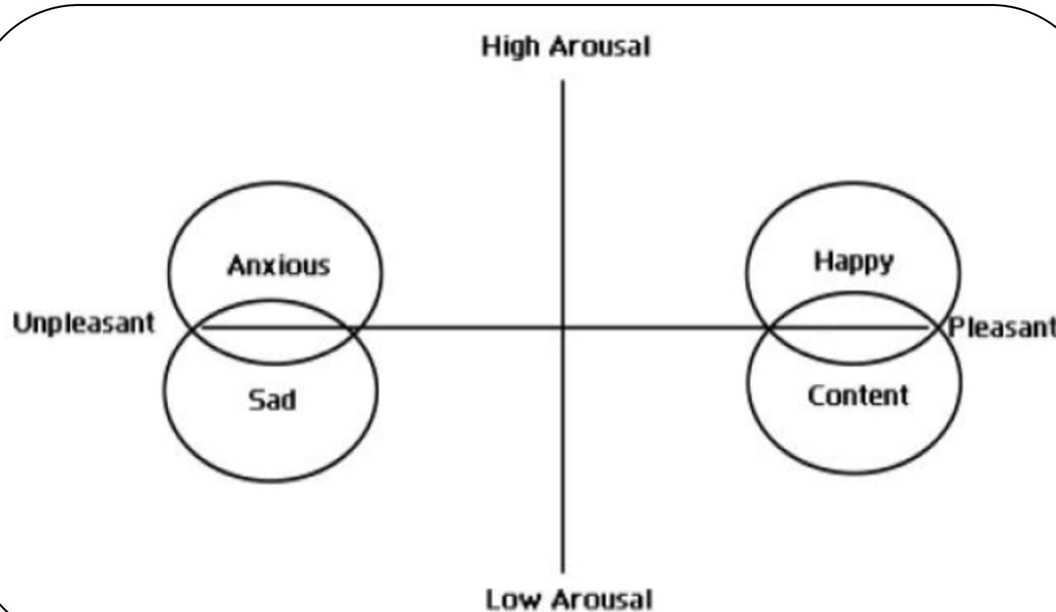


Figure 2: Emotions' Convergence

As we can observe, Anxious (or Upset) and Sad are very close neighbours in the Unpleasant – Pleasant scale. The same goes also for the emotional states of Happy and Content. Because of that, we cannot identify what exact emotion the subject feels when he give such valence and arousal rates. Such situations of emotional convergence occur very often, especially in the case of music listening. Specifically, when an individual listens to music, various emotions are evoked nearly simultaneously and that is something we should take under consideration when deciding which emotion model to choose for our study.

Eerola and Vuoskoski (2011), in a listening experiment concerning the comparison of discrete and dimensional emotion model concluded that, although both models, most of the time, provided approximately the same results regarding the correlation between perceived emotions and ratings, the discrete one, was less dependable when the music excerpts induced more than one perceived emotion to the subjects. In other words, it was difficult for the subjects to rate excerpts when the emotions induced by it were many or ambiguous (Eerola and Vuoskoski, 2011). That contradicts what we anticipated to see, because of the fact that, as explained on the previous paragraph, the dimensional model was the one with the impairment when emotions converge. However, as we see in this study, discrete model's outcomes were worst in that domain. That is something that can be explained and understood, because when someone is uncertain about what he feels, it is easier for him to define the valence (pleasant/unpleasant) and the arousal, than choosing an emotion from a predefined list.

One major issue of someone describing what he feels, resides in the fact that there is a difference between perceived and actual felt emotions. Gabrielsson (2001), through his study, confirmed that perceived emotions do not always harmonize with actual felt emotions. Additionally, as claimed by Lindquist et al. (2012), many times a subject either does not actually know what he feels, or he just feels embarrassed to speak the truth about it. He also may feels different things than what we expect from certain stimulators (e.g. music), because of personal experiences we cannot possibly know of in advance (Lindquist et al., 2012). For all those reasons, we decided to use dimensional conceptual model to examine the relationship, if any, between emotions and ratings.

Our Study

Having concluded that we will proceed with the dimensional emotion model, we should next examine if there is a way to transform its two neurophysiological dimensions (valence and arousal) into something measurable, capable of being used as implicit data, convertible to ratings. In the relevant literature, many studies have stated that valence and arousal, actual have a relationship with a few body physiological measures, meaning that if we can track them, we can also define the valence of an emotion or the intensity of it. The most commonly used physiological figures are skin temperature, galvanic skin response measuring skin conductance through sweat and beats per minute most notably used for heart rate variability calculations. Those measures can be recorded via properly constructed sensors, usually attached to individuals' fingers or wrists (Rosebrock, 2017; Christopoulos, 2016; Greco, 2016; Ferdinando, 2014).

So, considering the above factors and parameters, we concluded that the aim of this paper is to examine whether these certain physiological measures can be correlated with individuals' ratings, and play that way, the role of implicit data in a recommender system. To achieve that, we designed and conducted a listening experiment, in which 7 participants listened to a number of predetermined music excerpts, at the end of each, they were asked to rate them according to their preferences. At the same time, their physiological measures were being recorded through a wristband, which contained all the appropriate sensors. The experiments' results revealed that a correlation between listeners' measures and ratings indeed exists. These outcomes were then tried to be interpreted and possible biases and implications were been discussed.

Music Stimuli

Finally, we will discuss our choice of music as emotion stimuli and by extension as the ''item'' our subjects were instructed to rate. As reported by several studies throughout history regarding the influence of arts in the arousal of emotions, music has been identified as the one with the highest emotional response. Music has also been described as the most usually sought form of art when people want to express their feelings or change their emotional state (Huron and Margulis, 1993). Despite the fact that, the number of studies investigating how music affects our emotional state is relatively low, musical experiences are described as one of the stronger regulators of mood. Additionally, music plays the role of the enhancer of emotional valence and when combined with other stimuli (e.g. pictures) seemed to have bigger impact. In some cases totally outperformed other arts that used as emotion evokers (Hanser and Mark, 2013). As reported by Garlin and Owen (2006), background music can even affect consumers' behavior. Consequently, music is considered a strong emotion stimulus and therefore was chosen in this experiment.

Music Expectancy

According to various psychological studies, emotions can be induced by music through brainstem reflexes, visual imagery, memory, as well as musical expectancy (Juslin and Västfjäll, 2008; Zentner et al., 2001). The latter, refers to the anticipation of an upcoming musical event. Human brain has developed the survival mechanism of anticipation, according to which, we are able to expect what may occur in the near future so as to be prepared for it. This process causes physiological and psychological changes in our organism and evokes emotions (Blood and Zatorre, 2001). The expectation of a musical event evokes very similar processes in a listener's organism and therefore, emotions and other psychological reactions can be stimulated, such as surprise, awe, "chills," comfort or astonishment (Huron, 2006).

As claimed by Eerola (2003), there are two main process categories, in which we divide expectations' factors, the data-driven and the schema-driven process. Data-driven process considers as factors that affect expectations, all perceptual senses. Specifically, according to this process, information derived automatically by our senses affects our emotions and defines our behaviour. If such information (gathered through the data-driven process) emerges frequently, it transforms to knowledge and transferred into our long term memory, in order to able to be used again in the future. Thus, an experience is created. This transfer of information into our long term memory is called schema-driven process. So, through this process, our brain can create predefined behavioural or emotional patterns, ready to be evoked whenever a familiar stimulus emerges, or whenever we expect it to emerge (Eerola, 2003).

Music attracts attention by various factors, such as intensity, high or low tone and tempo, constancy (how often tempo fluctuates), changes in musical tension, lyrics, connected experiences etc. Our brain, when being accustomed to a continuous music theme (identified by some, or all of the previous factors), creates emotional patterns. When this music theme suddenly changes, the brain tries to interpret it and induces new emotions based on previous experiences and what we expect that we will listen (Eerola, 2003). That was also stated by Sloboda (1991), whose study presented that subjects’ observed emotional peaks occurred during unexpected musical changes. Our experiment’s design includes such music changes and therefore music expectation’s phenomenon occurrence is anticipated.

Feel Wristband

The wristband used in this study, is called Feel Wristband and was provided to us from the startup company Sentio Solutions, as part of a research collaboration with the Department of Management Science and Technology of Athens University of Economics and Business. It is an innovative IoT (Internet of Things) band, located on the wrist of a subject, with multiple integrated bio-sensors which monitor a variety of physiological signals such as skin conductance, skin temperature and heart beat rate. The Feel Wristband uses also a mobile application, with which is connected to via wifi. However, because of some implications, in our study the connection was only able through bluetooth signals. Through the mobile application the initiation and handling of the record procedure is possible. The researcher can start or stop record sessions, create flags or notes and generally control the whole procedure. After a recording session ends, the recorded data are saved in Sentio Solutions’ databases and in order to acquire them, we have to request them from the company.

Sentio Solutions Inc.

Sentio Solutions uses Feel Wristband to gather patients’ data, in order to eliminate their suffering from mood disorders. Specifically, the company’s goal is to help people with mood altering symptoms, by providing them with awareness about their emotions so as to gain a deeper understanding of their internal and external triggers. Through the recorded measures collected by the wristband, subjects are constantly informed about their emotional state and receive personalized guidance on how to better regulate their emotional experience. To gain that kind of intel, Sentio Solutions provide their customers with its wristband equipment and instruct them to wear it throughout the day, in order to monitor their activities, gather and analyze the acquired data through proprietary algorithms and finally give them tailor made recommendations and real-time coaching (Sentio Solutions 2018).

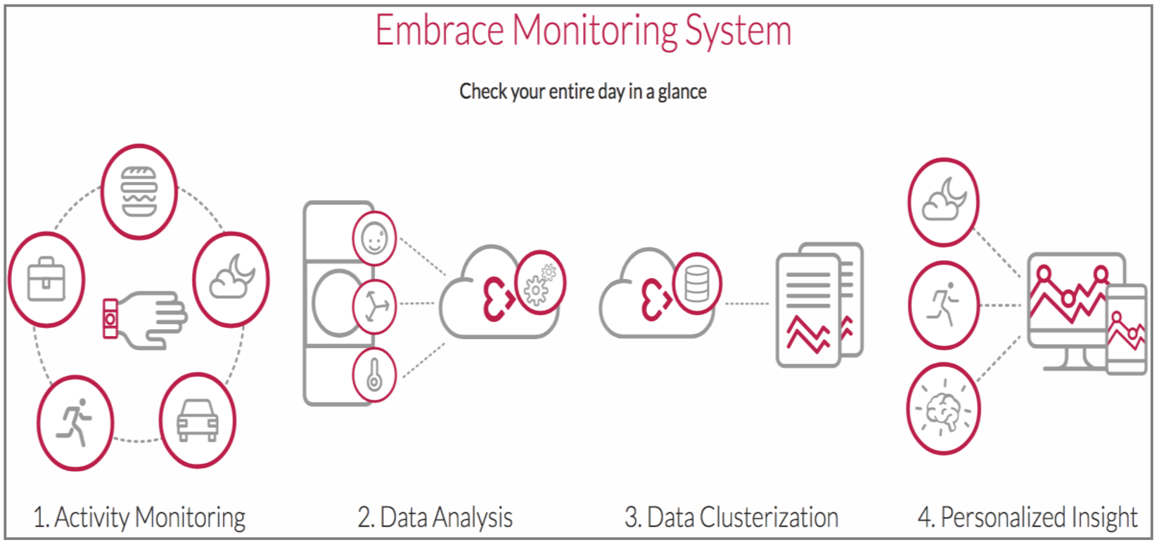


Figure 3: Sentio Solutions' Wristband Usage and Data Analysis

Physiological Measures

As it was briefly remarked in previous paragraphs, the physiological measures recorded through the Feel Wristband were:

Galvanic Skin Response (GSR)

With the term Galvanic Skin Response, also known as Electrodermal Activity (EDA), we refer to the changes of the sweat gland activity which is reflective to the emotional arousal. According to many studies, whenever emotions of an individual are evoked, tube-shaped sweet glands located on their skin, produce and excrete sweat (Bovell, 2015). This sweat gland activity alters the skin resistance. More specifically, the more sweat there is in the skin, the less skin resistance occur. In a GSR measurement, conductance is used to determine the relationship between the skin perspiration and the intensity of an emotion. Conductance is expressed as the opposite of resistance ($\text{Conductance} = 1/\text{ Resistance}$) and measured in Siemens. As a consequence, more perspiration produce higher conductance on the skin and by extension, the emotional arousal increases (Boucsein, 2012; Salimpoor et al., 2009). It is important to note that the valence of an emotion (negative or positive), does not influence the intensity of an emotional stimuli and by extension the GSR measurement. As follows, whenever someone feels joy (positive valence) or fear (negative valence) his GSR signal in both cases will only be high if his emotional arousal is high.

The GSR sensor implemented on the wristband used in this study, is consisted of silver chloride electrodes, which detect the electrical activity derived from the differences in the perspiration levels of the skin. In particular, the two electrodes which are in contact with the skin are provided with a constant voltage. Because of the constant voltage, the skin conductance is calculated by measuring the flow through the electrodes. As a result, any fluctuation in the flow, stems from the alteration of the sweat gland activity and consequently from the change in the emotional intensity.

Skin Temperature (ST)

A Skin Temperature sensor, designed for continuous temperature measurements, was also used in the experiment. Using the skin as an indicator of the body temperature, this particular sensor provided us with the temperature data of the participants while listening to music, measured in degrees of Celsius (°C). At that point it should been noted that skin temperature is affected by the environmental temperature, so the data for the experiment should be measured in a laboratory with the standard room temperature of 25°C. Except from the environment, sex difference also has an impact on the temperature of the skin. Females' temperatures decrease more or increase less than those of males' as stated in a previous study, while those temperature changes would not have occurred in the absence of the music (McFarland and Kadish, 1991). Given that, such skin temperature differences, need stimuli to be occurred and therefore are anticipated in the results of these listening experiments.

As claimed by various researches, fall in temperature occurs from the narrowing of the blood vessels (vasoconstriction) and correlated with increase in conflict. On the other hand, the dilation of the arterial blood vessels (vasodilation) results in an increase in skin temperature and is correlated with calmness and emotional security (Mittelmann and Wolff, 1943). Generally, a low skin temperature is due to increased activity of the sympathetic nervous system (SNS), which is responsible for the body’s fight response, preparing it for intense physical activity. On the contrary, a high skin temperature is a result of relaxation and a feeling of security and calmness that relaxes the body through the parasympathetic nervous system (PNS) (Richter and Wright, 2012). Consequently, low temperatures are related to emotions evoked from negative events and high temperatures from emotions evoked from positive events. That was also observed in McFarland’s (1985) listening experiment, in which, high intensity, negative emotion music resulted in a temperature decrease, while calm and positive emotion music resulted in listeners’ temperature increase.

However, this skin temperature pattern did not stay the same in every experiment regarding emotional response to music conducted. In the more recent study of Munde et al. (2012), it was argued that participants' skin temperature was higher when they were listening to low intensity negative emotion music and lower when low intensity positive emotion music was being played.

As a result, while it is safe to assume that skin temperature, indeed respond to different kind of music, the exact pattern of it remains debatable. Additionally, as it has been identified in the previous studies' results, skin temperature might be affected by the intensity of the music, rather than the valence of it. In another study, it was argued that the changes of skin temperature are a result of the arousal of each emotion, rather than the valence of it and what is more, music of both negative and positive valence, induces an increase in skin temperature (Lundqvist et al., 2009). This conclusion is consistent with the research of this study, because, as it will be described afterwards, despite of the emotional valence roused from each music excerpt, the skin temperature will always be increased and what is more, it will be correlated with the arousal and not the valence of emotions. Furthermore, in the previous studies, the selection of the music excerpts has been accomplished with regard to both the valence and the arousal of the emotion evoked by each excerpt. In this study however, the difference between the selected music excerpts was only the valence of the emotion that they will induce and not the arousal, as each and every excerpt, is considered to be of high emotional arousal for the individual that chose to listen to it.

Beats per Minute (BPM)

A Beats per Minute sensor has been also implemented on the wristband to count the number of pulses per minute. This sensor uses an optical LED light source, which shines through the wrist and measures the amount of light that reflects back because of the pulse. Consequently, the device can record the subject's heart beat rate, as well as the exact second of the minute the beat occurred. Regarding the relationship between heart beat rate and music, it has been observed by various studies that there is a correlation with the music tempo. Specifically, it has been stated that fast and loud rhythms increase pulse tempo and by extension heart beat rate (Wang and Huang, 2014). Additionally, during fast tempo music excerpts, the overall activity of the heart beat rate is increased regardless of its arousal (Bobrowski and Knipschild, 2016).

METHODOLOGY

Analysis Software

For the analysis of the data collected from the listening experiment and the export of the results, were used mainly Python 3.6 scripts. That refers to almost all outcome procedures, including the extraction and analysis of the various plots as well as the implementation of the K-Means clustering and machine learning predicting algorithms. In addition, to confirm that some outcome values were correct, such as correlation coefficients or cluster centroids, RapidMiner Studio Software was used to compare its results with those of the Python code. Regarding the cleaning and transformation of the data, apart from Python, MySQL scripts were also used, as well as Microsoft Excel Software.

Participants

Seven subjects participated on a voluntary basis in the experiment, with their age ranged from 25 to 32 years (M = 28.42, SD = 2.69). Three of them were females, while the other four were males. Their musical expertise varied from amateurs to experts, including a music producer writing his own music, whereas their preferred musical styles were rather diverse. Furthermore, all participants have stated that consider listening to music an important part of their entertainment and expression. Further exclusion criteria applied to subjects were: not being under 18 years old, with hearing problems or under medication that might influence their mood or concentration.

Finally, participants' educational levels varied between bachelor's, master's as well as PhD degrees. The small amount of subjects participating in the study renders the experiment not scientifically accurate as it lacks the desirable quantity, but as it will be presented in the results, it is sufficient enough to provide us with qualitative indications regarding the relationship between the physiological measures recorded and the musical preferences of the listeners.

Music Collection

The music tracks (40 in total) used in the study, were derived from Spotify, a digital music platform with millions of songs and other content from artists all over the world. More specifically, the most popular music categories within Greek audience have been singled out, from which the most popular artists have been selected. Afterwards, from each and every artist, the most famous songs have been chosen to be included in the study's music list. The purpose of this selection was to increase the listeners' possibility of having already heard the songs, so as to be expressed by them and by extension maximize their emotional response.

The selected songs were also cut, in order to range from 58 to 188 seconds in length (mean duration = 101.2 seconds) and classified into the following eight music categories: Dance, Pop, Greek Pop, "Λαϊκά" (Greek folk songs), " 'Εντεχνο " (containing elements of Greek folk, rock and orchestral music), Rock, Hard Metal and Rap. Each category contained five musical excerpts, the duration of which, were decided based on the fact that each excerpt should be long enough to induce subjects' emotions, but short enough not to tire or bore them.

PROCEDURE

The listening experiment was carried out individually for each subject and controlled by two assessors in a furnished and temperature controlled (25 °C) laboratory of Athens University of Economics and Business (Trias 2, Athens 113 62). Prior to attending the session, subjects had been informed about the structure and the main purposes of the experiment. On that basis, they were prompted to listen to music naturally and not suppress their emotions. On the contrary, they were advised to feel any emotion evoked to them from the music and express in whatever way they are comfortable with.

To begin with the experiment procedure, the participants were asked to sit comfortably and wear, with the proper instructions, the Feel sensor band on their wrist, in order for the physiological measures to be recorded. Since intense motion can affect and distort the data recorded from the sensors of the wristband, subjects were notified accordingly, so as to hold their banded arm as steady as possible. In front of them, was a computer screen which displayed eight Windows files, each one represent a separate music category created by the researchers. Subjects were asked to choose two out of the eight categories, depending on their liking. More specifically, they were instructed to pick their favorite and worst category, based on their musical preferences. Additionally, to eliminate environmental noise, instructors provided participants with a set of soundproof headphones to wear during the listening session. After the instructors addressed any other queries the participants had, they asked them to wear the headphones and adjust the music volume according to their liking.

Finally, the musical excerpts of the participants' favorite category were played successively through Windows Media Player. When the first music category finished, subjects were given 30 seconds to relax and consequently, the excerpts of their worst music category were played. During this procedure, at the end of each excerpt, the participants were being asked to give the assessors a rating on a scale from 0 to 5. That way, researchers were able to define listeners' preferences and inclinations regarding the music they listened to each time, in order to identify if there is a relationship between them and their recorded physiological measures. The playing order of the excerpts was defined due to the aim of the researchers to detect the changes taking place (if any) on the recorded data, while the music passes from the category listeners like (favorite) to the one they do not (worst).

RESULTS

Data Handling

After the experiment was over and all participants have successfully completed the procedure mentioned above, recorded data had to be gathered from Sentio Solutions Inc. and properly denoized and transformed in order to be able to be analyzed. Initially, all missing values and outliers were removed, so as not to affect the experiment results. Physiological measures such as electrodermal activity and skin temperature, apart from the subjects' mood and emotions, are quite sensitive also to internal (inner body changes and status), as well as external environment circumstances (room temperature, interaction with the assessors etc.). Given that, sometimes it is only natural, outlier data to be detected and erased. Furthermore, because of the fact that sometimes the connection between the wristband and the mobile was lost for a few seconds, breaks or missing values spotted on the data had to be cleaned. In fact, in the cases of two out of seven participants' sessions, the disconnection lasted several minutes resulted in severe data loss. Therefore, their recorded measures excluded from the analysis, in fear of outcomes' distortion. In transforming and denoizing the data, we mainly used Python's Pandas and Numpy libraries and various techniques such as splitting, concatenating, merging etc., a few examples of which are presented in the following figures:

```
mat = loadmat('C:/det/1st_test/data/Subject4_dance.mat')

df = pd.DataFrame(mat['        dance'][0])
#print(list(df))

dfP1 = pd.DataFrame(df.iloc[:,0][0])
dfP2 = pd.DataFrame(df.iloc[:,5][0])
dfP3 = pd.DataFrame(df.iloc[:,6][0])
dfP4 = pd.DataFrame(df.iloc[:,2][0])
dfP5 = pd.DataFrame(df.iloc[:,9][0])
dfB = pd.DataFrame(df.iloc[:,7][0])
dfG = pd.DataFrame(df.iloc[:,8][0])
dfST = pd.DataFrame(df.iloc[:,1][0])
dftime = pd.DataFrame(df.iloc[:,18][0])
list_ = [dt.utcfromtimestamp(row[0]).strftime('%H:%M:%S') for index, row in dftime.iterrows()]
dftime = pd.DataFrame(np.array(list_))

#Merge
df1 = dfP1.merge(dfP2,left_index=True,right_index=True,suffixes=(1,2)).merge(dfP3,left_index=True,right_index=True).merge(dfP4,
df1.rename(columns={'01':'P1','02':'P2','03':'P3','04':'P4','05':'P5','06':'BPM','07':'GSR','08':'ST',0:'Time'}, inplace=True)
print(df1.head(5)) #test
```

Figure 4: Data Transformation 1st Script Example

```
frames = [df1, df2]
df = pd.concat(frames)
#print(df) #test

#Check for NaN
print(df.isnull().any())
```

Figure 5: Data Transformation 2nd Script Example

```
m_r = pd.read_csv('M_data.txt', sep="|", header=None)
m_r.columns = ['Song','Starting','Ending','Rating','Familiarity']


dfmr = pd.read_csv('..../Subject1_Ratings.csv')
dfmr_start = pd.DataFrame(dfmr['Starting'])
dfmr_stop = pd.DataFrame(dfmr['Ending'])

dfmr_start_stop = dfmr_start.merge(dfmr_stop,left_index=True,right_index=True,suffixes=(1,2))

dfm = dfm.set_index(['Time'])
#print(dfm.head()) #test

list_ = []
array_ = np.array(dfmr_start_stop)
for i in range(len(array_)):
    list_.append(dfm['GSR'].loc[dfmr_start_stop['Starting'][i]:dfmr_start_stop['Ending'][i]].mean())

dfm_means = pd.DataFrame(list_)
dfm_means.columns = ['GSR_Means']
#print(dfm_means) #test

dfmr_rates = pd.DataFrame(dfmr['Rating'])
dfmr_rates.columns = ['Rates']
#print(dfmr_rates) #test

dfm_means.to_csv('.....\Subhect1_gsr_means.csv', index=False)
dfmr_rates.to_csv('.....\Subject1_rates.csv', index=False)
```

Figure 6: Data Transformation 3rd Script Example

Time Series Plots

When the data were ready for use, time series plots for every measure of each participant were created. Specifically for all the remaining five subjects, graphs with their GSR, ST and BPM data progression through time were exported, for the purpose of identifying their status before and after the music category change. Some script examples used for the plots, are shown below:

```
#Plots
time = list(dfj['Time'])
x = matplotlib.dates.datestr2num(time)

plt.plot_date(x, yj_GSR, markersize=1, lw=0.5, ls='--', c='r')
plt.xlabel('Time')
plt.ylabel('GSR')
plt.title('Subject 1 GSR Time Plot')
plt.show()
plt.plot_date(x, yj_ST, markersize=1, lw=0.5, ls='--', c='r')
plt.xlabel('Time')
plt.ylabel('ST')
plt.title('Subject 1 ST Time Plot')
plt.show()
```

Figure 7: Plots Generation 1st Script Example

```
# Subplots

fig = plt.figure()
ax1 = fig.add_subplot(311)
ax2 = fig.add_subplot(312)
ax3 = fig.add_subplot(313)

ax1.plot(x, y_GSR, c='r', markersize=2, lw=0.5, ls='--')
ax1.set_xlabel('Time')
ax1.set_ylabel('GSR')

ax2.plot(x, y_ST, c='r', markersize=2, lw=0.5, ls='--')
#ax2.set_xlabel('Time')
ax2.set_ylabel('Skin Temp')

ax3.plot(x, y_BPM, c='r', markersize=2, lw=0.5, ls='--')
#ax3.set_xlabel('Time')
ax3.set_ylabel('BPM')

plt.suptitle('Bill')
#plt.legend(loc='upper left')
plt.show()
```

Figure 8: Plots Generation 2nd Script Example

Despite of some minor differences, the results of all subjects were the same. GSR and ST physiological measures were significantly higher when a music excerpt was considered of negative valence by the listener. More precisely, emotional arousal (defined by the GSR measure) and skin temperature were far higher when the listeners have given low ratings, which means that whatever their emotions were, they were felt more intensively during the listening of the worst category music, than the ones felt during the listening of the favorite category music. During the experiment, while the participants were listening to their worst music category, boredom or irritation (or both) was observed in their movements and facial expressions. Some of them were even asked if they were permitted to change or skip the music excerpt that was being played. Considering that and in response to the results, we can assume that negative emotions, like annoyance, have a bigger impact on GSR and ST variables. The above statements are also observed in the in the figures given below, containing two example GSR and ST time plots of subjects 1 and 2.

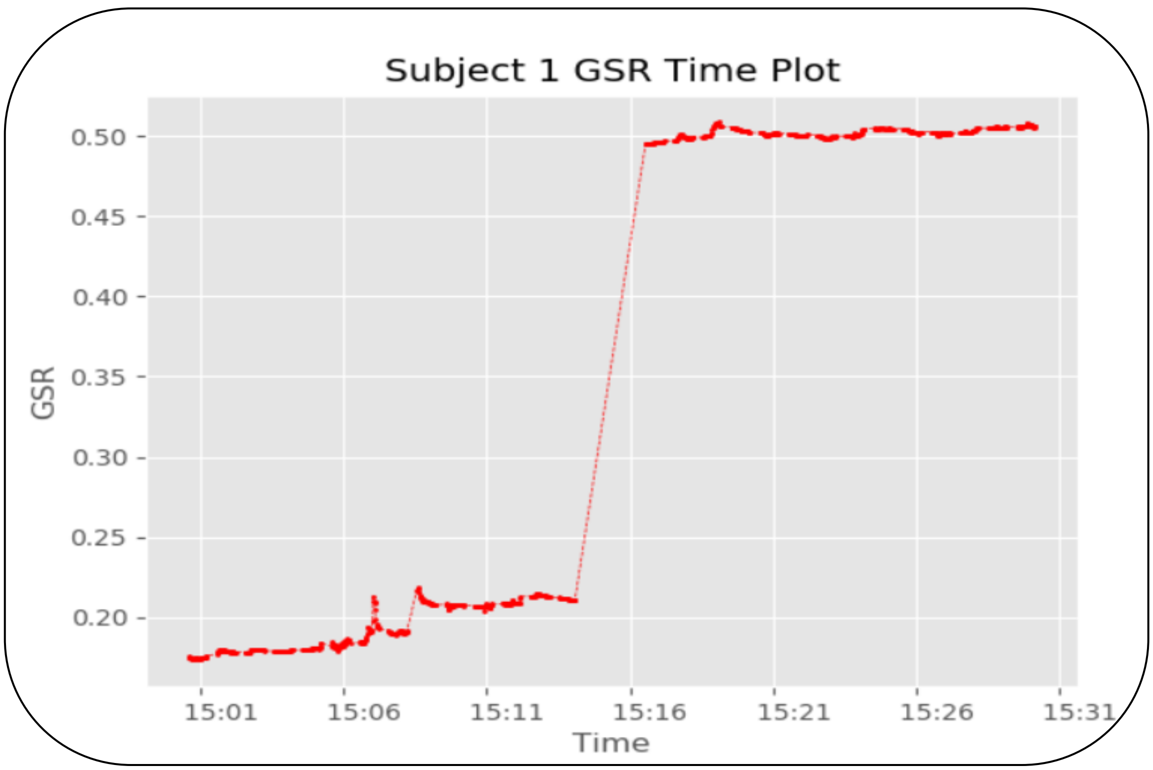


Figure 9: GSR Time Plot of Subject 1

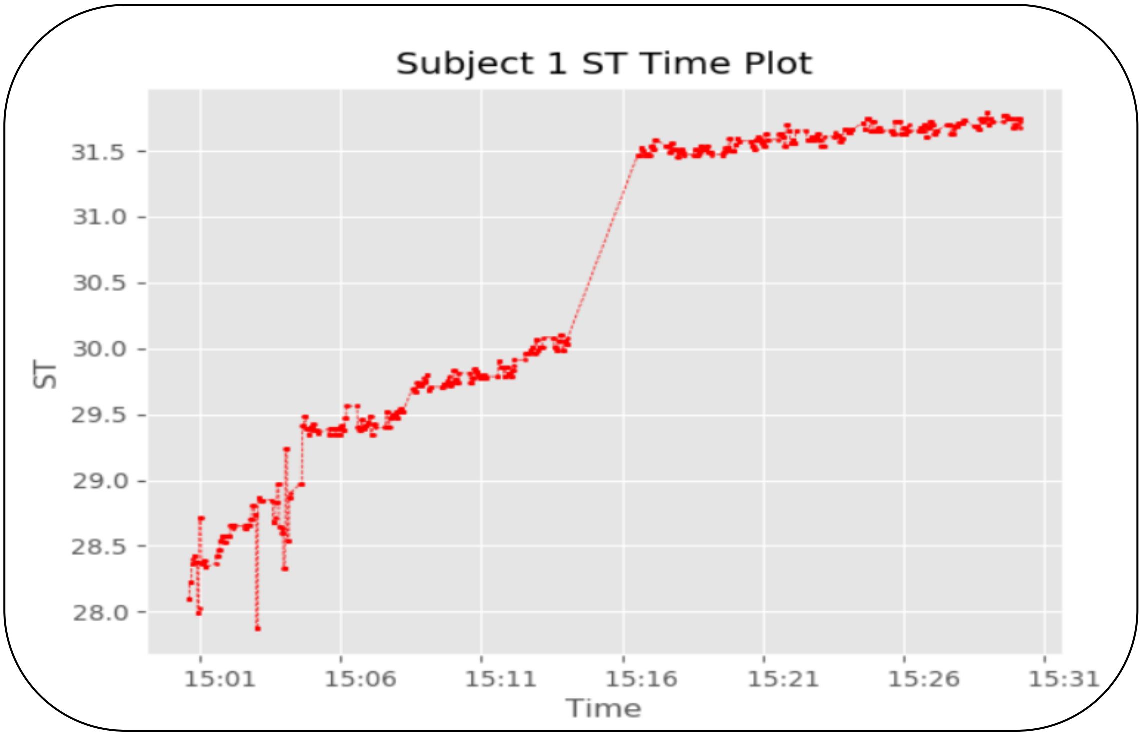


Figure 10: ST Time Plot of Subject 1

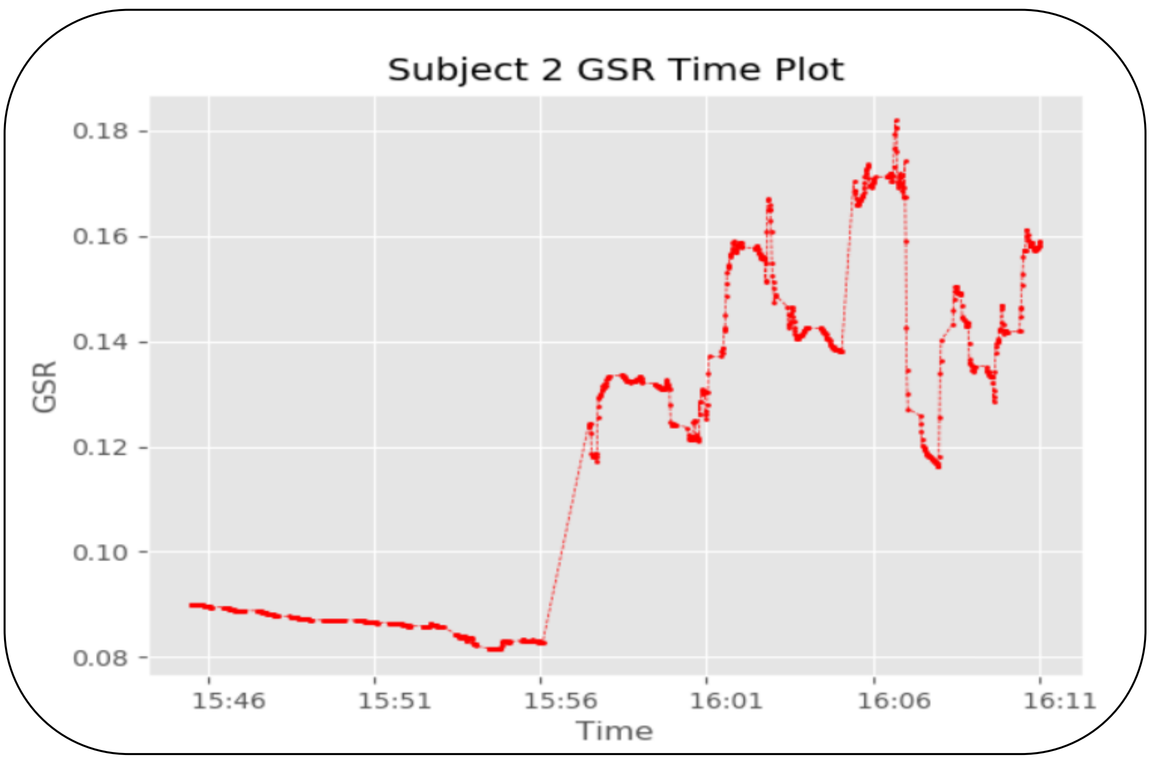


Figure 11: GSR Time Plot of Subject 2

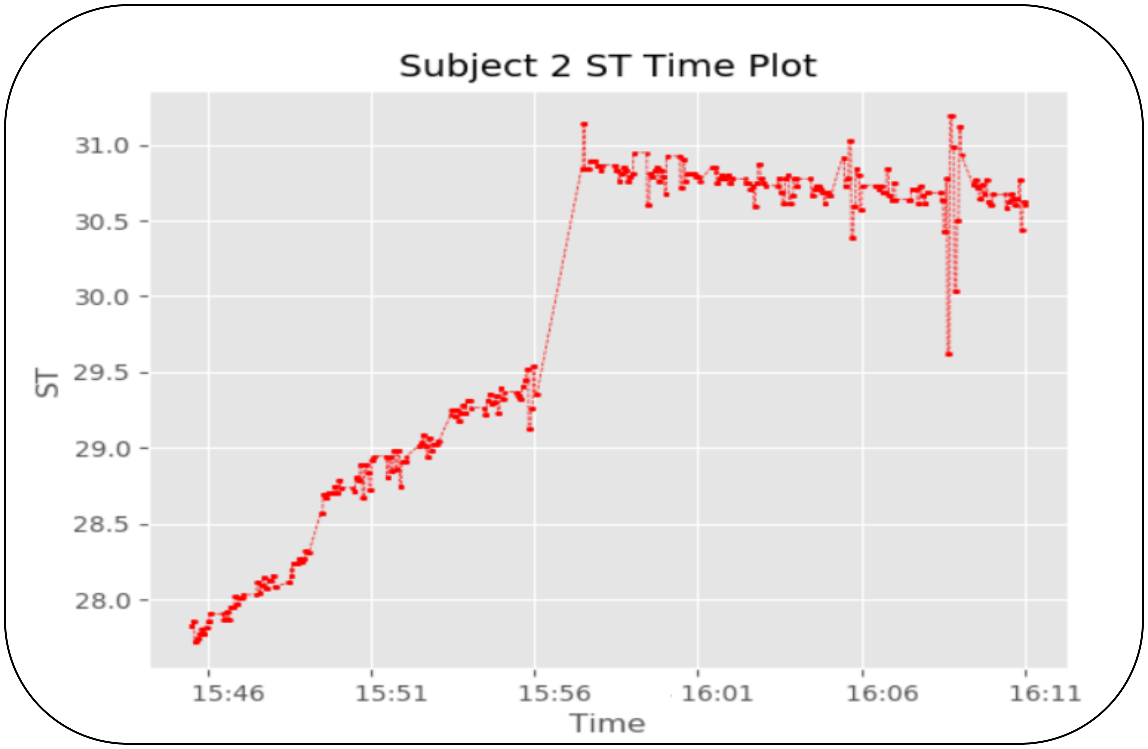


Figure 12: ST Time Plot of Subject 2

As it is observed in the time plot examples of subjects 1 and 2, both GSR and ST measures advance steeply after a certain point in time, which reflects music change from the category the subjects liked to the one they did not. Another fact that can be identified in all graphs is that after the music category's change, data start to be recorded from an already higher point. Thus, in the time gap intervened between the alternation of the two music categories, where no music excerpt was being played, GSR and ST measures were highly increased. That can only be explained through the music expectancy theory. As explained in the introduction (Music Expectancy), human brain is able to anticipate what may occur in the near future and that way can be prepared for it. This procedure influences the body and evokes emotions. In the same manner, the anticipation of a musical event also evokes emotions and apparently affects the physiological measures of GSR and ST. It should be noted that such hypothesis makes sense in our occasion, because in the experiment all participants have personally chosen both their favorite and worst music category and knew what follows far in advance. For this reason, another experiment should be conducted, using a randomized system of music playing, so as to be impossible for the participants to know beforehand, what they are going to listen to. Only by comparing these results with the ones we already have, is possible to reach a safe conclusion as to whether music expectancy affects the listener or not and deduce to how much.

On the other hand, as opposed to the GSR and ST data, there were no significant differences in BPM during the listening experiment. As stated by the recorded data of all five participants, BPM measure's progression through time did not displayed any upward or downward trend. The following time plots depict the progression of subjects' 1 and 2 BPM.

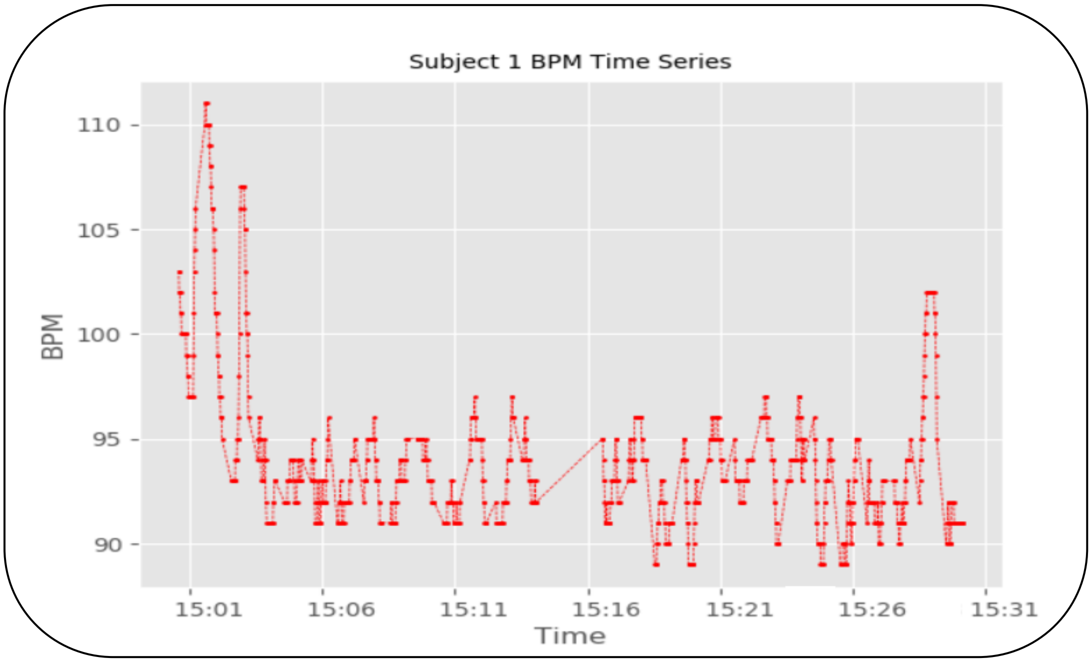


Figure 13: BPM Time Plot of Subject 1

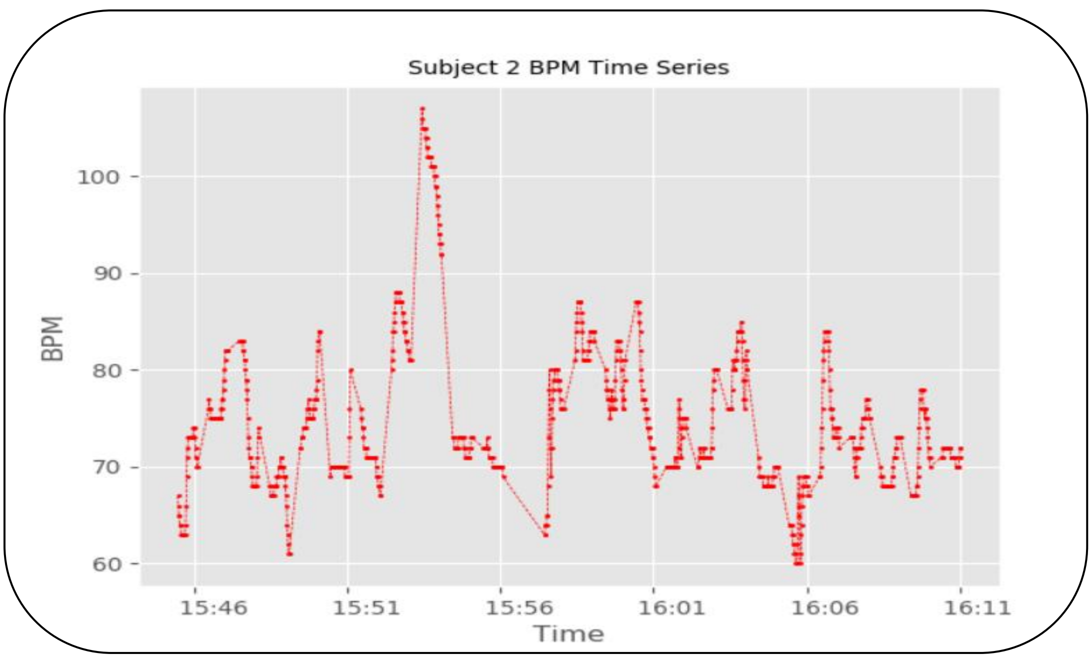


Figure 14: BPM Time Plot of Subject 2

Physiological Measures – Ratings Correlation

To further analyze BPM's influence, we measured the correlation between the ratings of each music excerpt listened per participant, and the respective BPM mean recorded on each excerpt. To achieve that, we used Python's SciPy library. The following figures depict the relevant script and correlation results of Subject 1.

```
import numpy as np
import pandas as pd
from scipy.stats import *

#Subject 1

#Load data
df1 = pd.read_csv('.../Subject1.csv')
yb_BPM = df1['BPM']


#Calculate BPM Means for every excerpt
mg1 = yb_BPM.iloc[0:88].mean()
mg2 = yb_BPM.iloc[89:190].mean()
mg3 = yb_BPM.iloc[190:248].mean()
mg4 = yb_BPM.iloc[249:390].mean()
mg5 = yb_BPM.iloc[391:539].mean()
mg6 = yb_BPM.iloc[540:626].mean()
mg7 = yb_BPM.iloc[627:760].mean()
mg8 = yb_BPM.iloc[761:820].mean()
mg9 = yb_BPM.iloc[821:955].mean()
mg10 = yb_BPM.iloc[956:1082].mean()

bpm = np.array([mg1, mg2, mg3, mg4, mg5, mg6, mg7, mg8, mg9, mg10], dtype=np.float64)
ratings = np.array([5, 2, 4, 4, 5, 2, 1, 1.5, 3.5, 2.5], dtype=np.float64)

r_value,p_value,std_err = linregress(bpm,ratings)
print('\nScipy Results:')
print('Coefficient of Determination = ', round(r_value**2, 3))
print('Corellation = ', round(r_value, 3))
print('P_Value = ', round(p_value, 3))
```

Figure 15: Subject's 1 Ratings-BPM Means Correlation script

```
Scipy Results:
Coefficient of Determination =  0.107
Corellation =  0.327
P_Value =  0.356
```

Figure 16: Subject's 1 Ratings-BPM Means Correlation Results

As we can see from the results in the above figure, Coefficient of Determination (R^2) was nearly zero (0.107), while correlation (r) between BPM Means and Ratings was also not strong enough (0.356). Additionally, P_Value was 0.356. The fact that P_Value exceeds the level of significance of 0.05 indicates weak evidence against the null hypothesis of $r = 0$ and therefore we fail to reject it. That means that r is not statistically significant and there is no linear relationship between BPM and Ratings. Similar results were observed for all subjects and null hypothesis of $r = 0$ was not rejected in all five cases.

As stated in the Introduction, BPM measures' activity is affected by fast and loud music tempos. More precisely, its fluctuation is increased with fast enough rhythms. Most of the musical excerpts used in this study had fast tempos and their volume, during participants' listening, was also pretty high. As a result, the BPM outcomes are comprehensible.

The same calculation as the one between BPM and Ratings was also performed for the GSR and ST measures for every participant. The correlations (r) exceeded -0.67 in all five cases, confirming the strong negative correlation configured between GSR – ST and Ratings. The P_Value was lower than 0.035 in all five cases and therefore we rejected the null hypothesis. Consequently, some of the variation in Ratings is explained by variation in both GSR and ST. It should be noted that in all cases, the ST-Ratings correlation was stronger than the GSR-Ratings one. In next figures are presented the relevant scatter plots of Subject 1, depicting the negative relationship between Ratings and GSR, ST physiological measures.

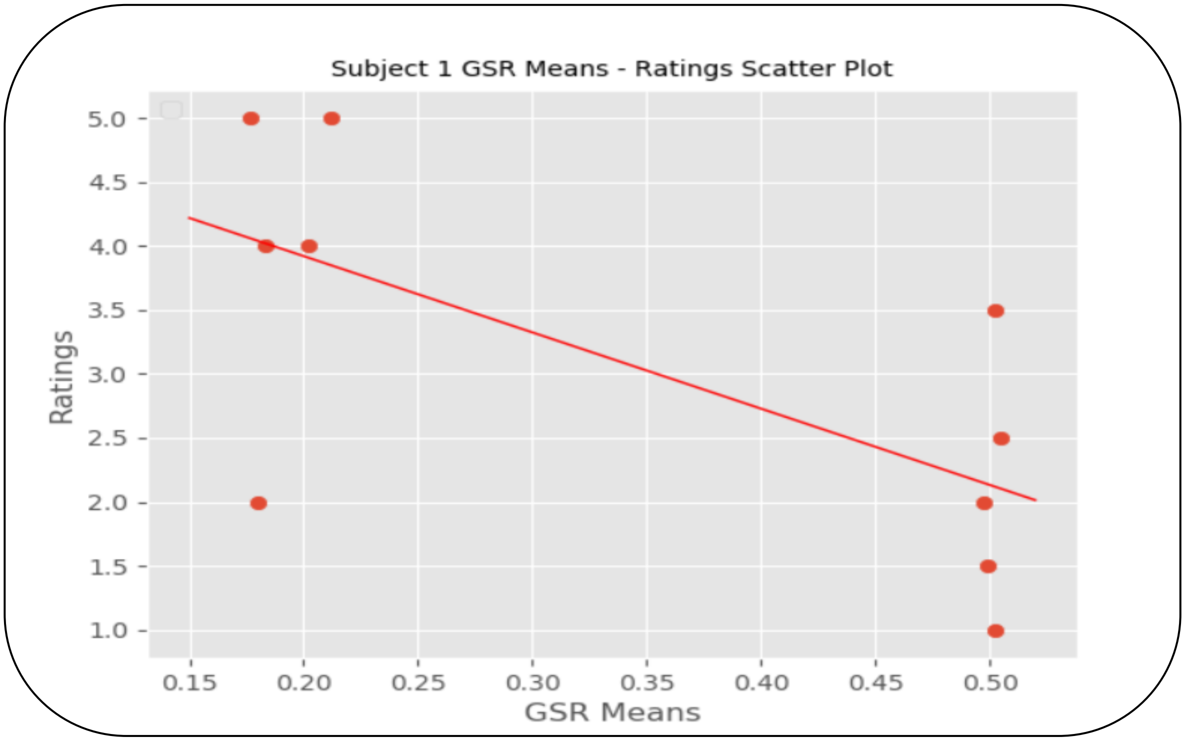


Figure 17: Ratings-GSR Means Scatter Plot of Subject 1

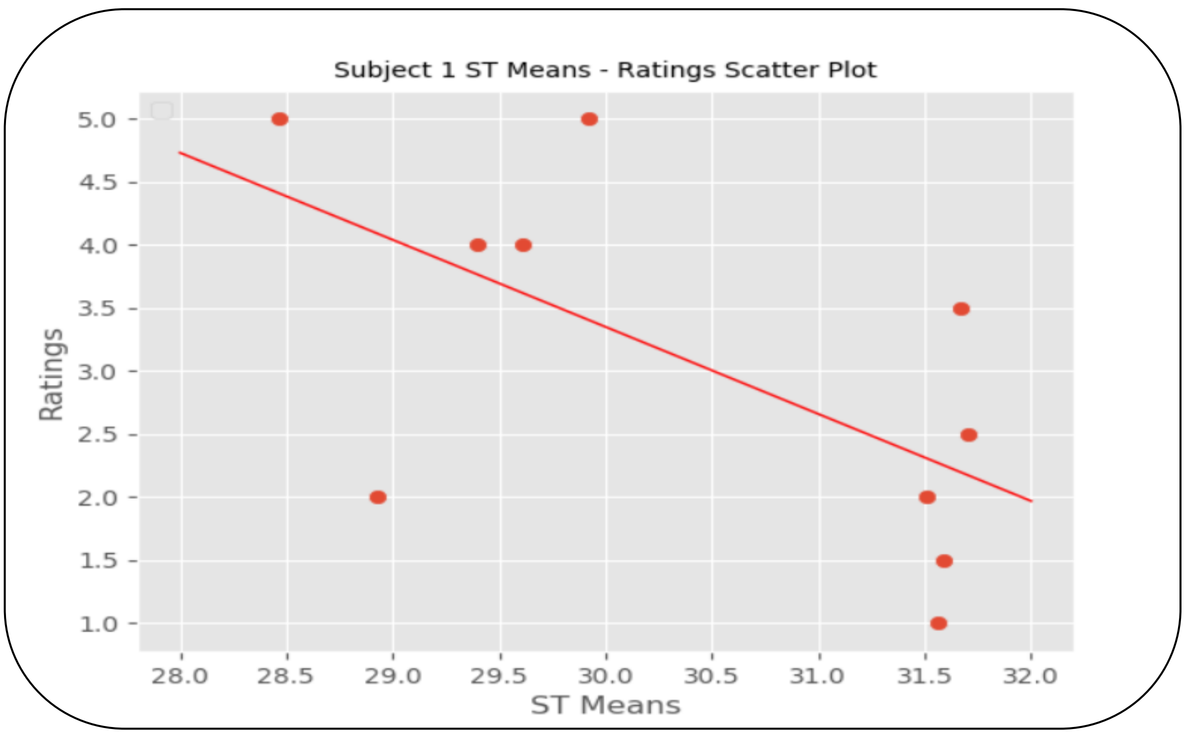


Figure 18: Ratings-ST Means Scatter Plot of Subject 1

In order to examine in more depth this negative relationship between ratings and GSR and ST respectively, we constructed two scatter plots for all five participants at the same time:

GSR – Ratings Means Scatter Plot

X Axis: Participants' means of GSR data recorded during the listening of each category

Y Axis: Participants' means of ratings for each category (favorite and worst)

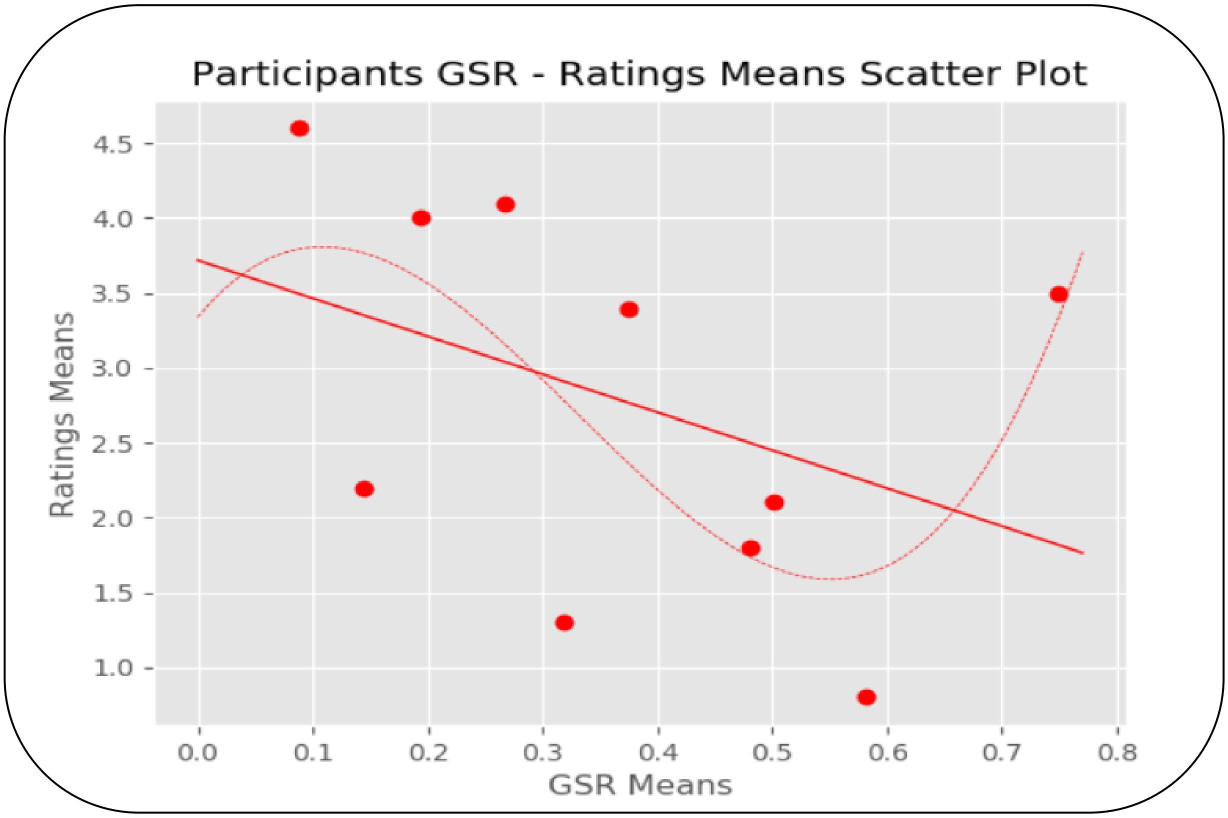


Figure 19: All Subjects' GSR Means – Ratings Means Scatter Plot

GSR – Ratings Means Scatter Plot

X Axis: Participants' means of ST data recorded during the listening of each category

Y Axis: Participants' means of ratings for each category (favorite and worst)

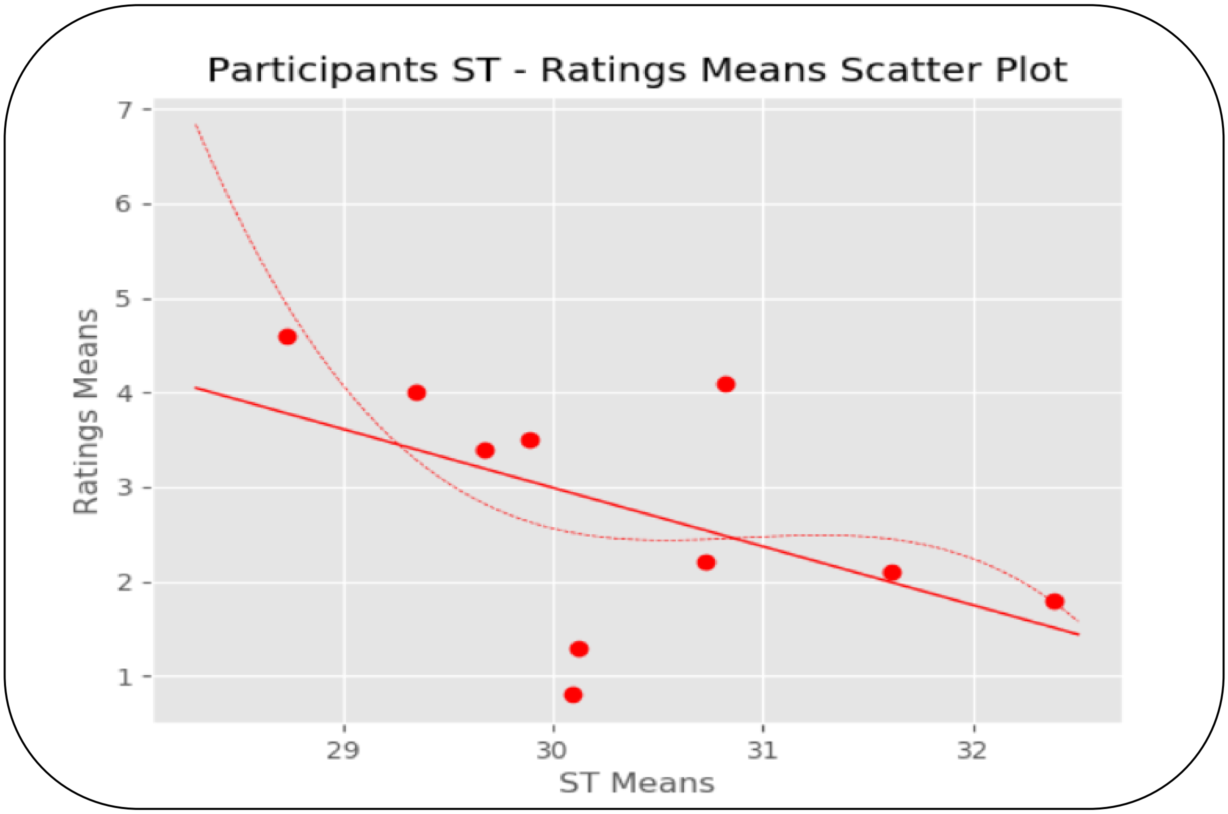


Figure 20: All Subjects' GSR Means – Ratings Means Scatter Plot

As it is observed in the figures 19 and 20, while the means of ratings are increased, both GSR and ST means are decreased. It should be noted that, although a negative correlation between these measures clearly exists, it is not as strong comparing to the correlations calculated for each participant separately. That is natural because every participant's GSR and ST data have their own slightly different scale of Siemens and °C respectively. What is more important, is that despite of those differences, the negative trend between listeners' music preferences and their physiological measures of GSR and ST can still be formed and identified. We will revisit and analyze GSR-ST and Ratings correlation more meticulously in Prediction section.

GSR-ST Correlation

Although by this point it was obvious, scatter plots between GSR and ST were also created for every participant, to define the relationship between those two measures. As it is shown below, a strong positive correlation was identified between the two measures for both subject 1 and 2. Similar outcomes also apply to the other 3 subjects.

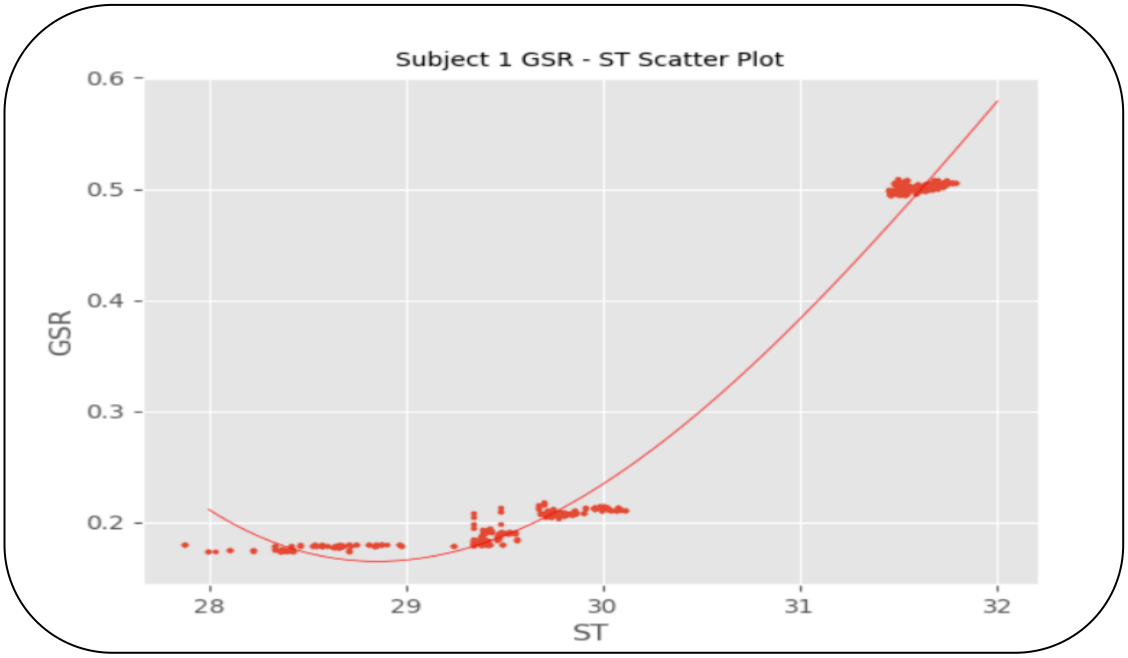


Figure 21: GSR – ST Scatter Plot of Subject 1

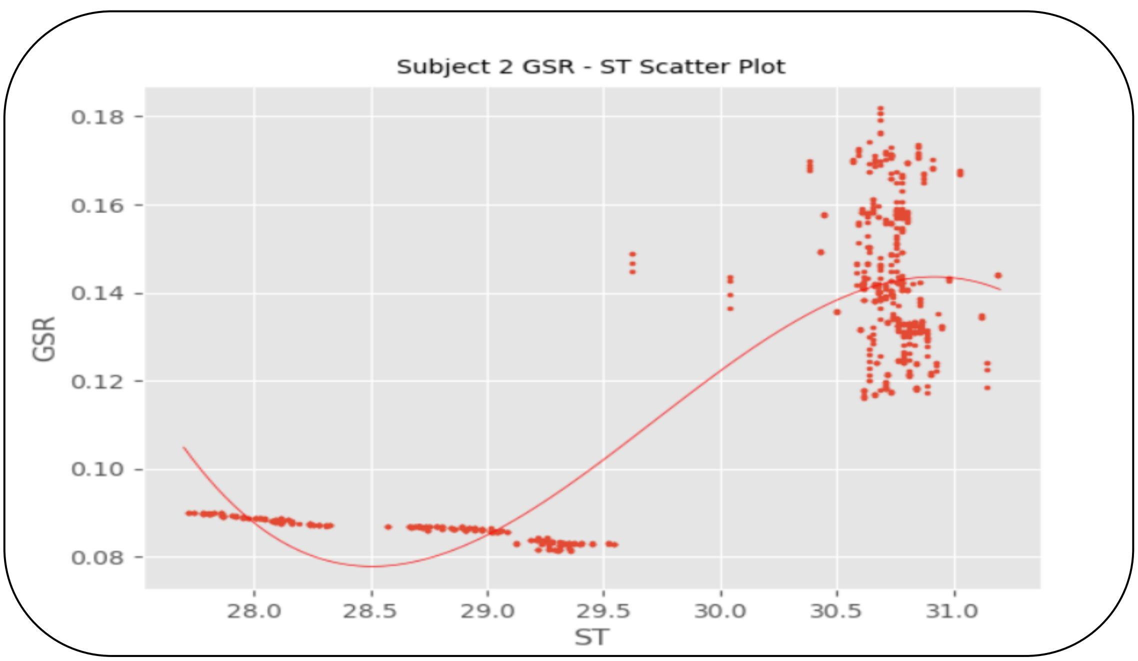


Figure 22: GSR – ST Scatter Plot of Subject 2

To confirm this positive relationship, we used the script presented bellow to extract the exact correlation and coefficient of determination for each subject.

```
# GSR - ST Correlation
print('GSR-ST Correlation: \n', df[['GSR', 'ST']].corr())

from scipy.stats import *
r_value= linregress(y_GSR,y_ST)
print('\nCoefficient of Determination = ', r_value**2)
```

Figure 23: GSR – ST Correlation Script

Subject 1 GSR - ST Correlation		
	GSR	**ST**
GSR	1	0.9605
ST	0.9605	1
Coefficient of Determination		0.92257

Subject 2 GSR - ST Correlation		
	GSR	**ST**
GSR	1	0.8377
ST	0.8377	1
Coefficient of Determination		0.70182

Figure 24: GSR – ST Correlation Results for Subjects 1 & 2

Scatter plots between GSR and BPM, as well as ST and BPM, for every subject were also created and the relevant correlations were calculated to investigate if there is a relationship between BPM and either of the other measures. However no such relation was identified, as in all cases the correlation was close to zero. Given that, we will no longer include BPM physiological measure in our calculations and models, as further analysis regarding it seems fruitless.

PREDICTIONS

Machine Learning

One of the fastest and more sufficient ways of making predictions currently exists, is Machine Learning. By that term, we refer to the science of training computers to learn automatically over time how to respond to us with a prediction if we feed them with specific data and information. Basically, machine learning is a set of algorithms we created to improve software' prediction capabilities. These algorithms receive data and using various statistical methods, provide us with a prediction output. While we keep feeding them with data, they will continue to train and spawn new, updated outcomes, adapted to the information they receive (Shalev-Shwartz and Ben-David, 2014).

Machine learning algorithms (or models) are divided to two major categories, supervised and unsupervised. The main difference is that, in a supervised model instructors are needed to determine which specific variables and parameters it will take under consideration to generate an output. Unsupervised algorithms do not need to be specifically trained in order to produce an output. They automatically identify which parameters, variables and connections they will use to generate the final outcome (Ayodele, 2010).

Clustering

Clustering is a Machine Learning technique in which, given a set of data points, we can use a clustering algorithm to classify each data point into a specific group (Omran et al., 2007). Unsupervised K-Means clustering was applied between GSR and ST, to analyze their relationship and define the clusters which would be created by the algorithm itself. K-Means algorithm, classifies n observations of the data set into k clusters. Each observation is included in the cluster with the nearest mean (considered the centroid). The algorithm try to minimize the distance (measured with various methods, e.g. Euclidean distance) between the centroid of the cluster and the given data point by appending it to any closer cluster and terminate when the lowest distance is achieved (Morissette and Chartier, 2013). In the next figure it is shown the Python code used to generate the unsupervised K-Means model, utilizing the Scikit-learn's library MeanShift.

```
#Unsupervised Clustering
from sklearn.cluster import MeanShift

ms = MeanShift()
ms.fit(npcl)
labels2 = ms.labels_
centroids2 = ms.cluster_centers_

clusters2 = len(np.unique(labels2))

for i in range(len(npcl)):
    plt.plot(npcl[i][0], npcl[i][1], colors[labels2[i]], markersize = 3)
plt.scatter(centroids2[:,0], centroids2[:,1], marker='x', c='k', s=50, linewidths=10, label='Centroids')
plt.ylabel('GSR')
plt.xlabel('ST')
plt.title('Subject 2 GSR-ST Unsupervised Clustering', fontsize = 10)
plt.legend(loc='upper left')
print('\nUnsupervised Clustering\nClusters: ',clusters2,'\n')
print('Centroids: \n',centroids2)
plt.show()
```

Figure 25: Unsupervised K-Means Model Scipt

As someone can observe in the following figures (Subject's 1 and 2 Unsupervised Clustering), three clusters were created in both cases. The first two clusters (blue and red color) depict the subject's physiological measures' status when the favorite music category was played, while the third one (green color) depict their measures' status when the worst music category was played.

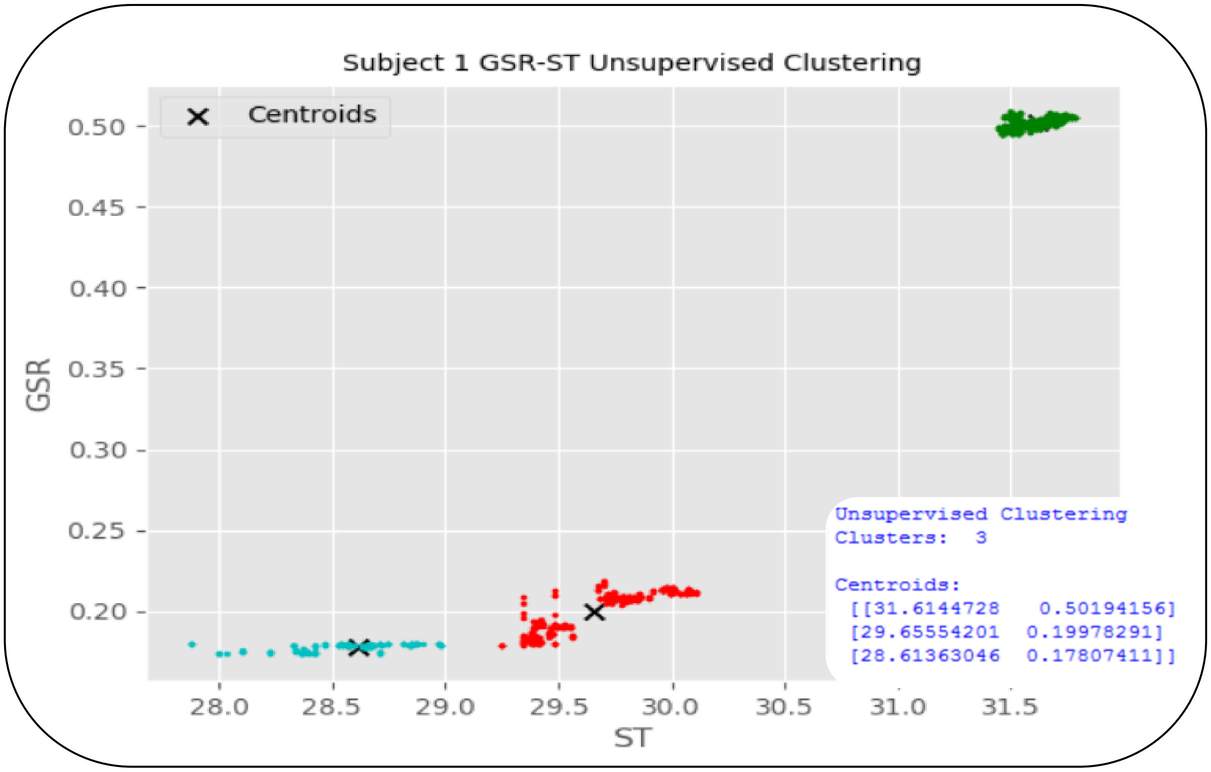


Figure 26: GSR – ST Unsupervised Clustering for Subject 1

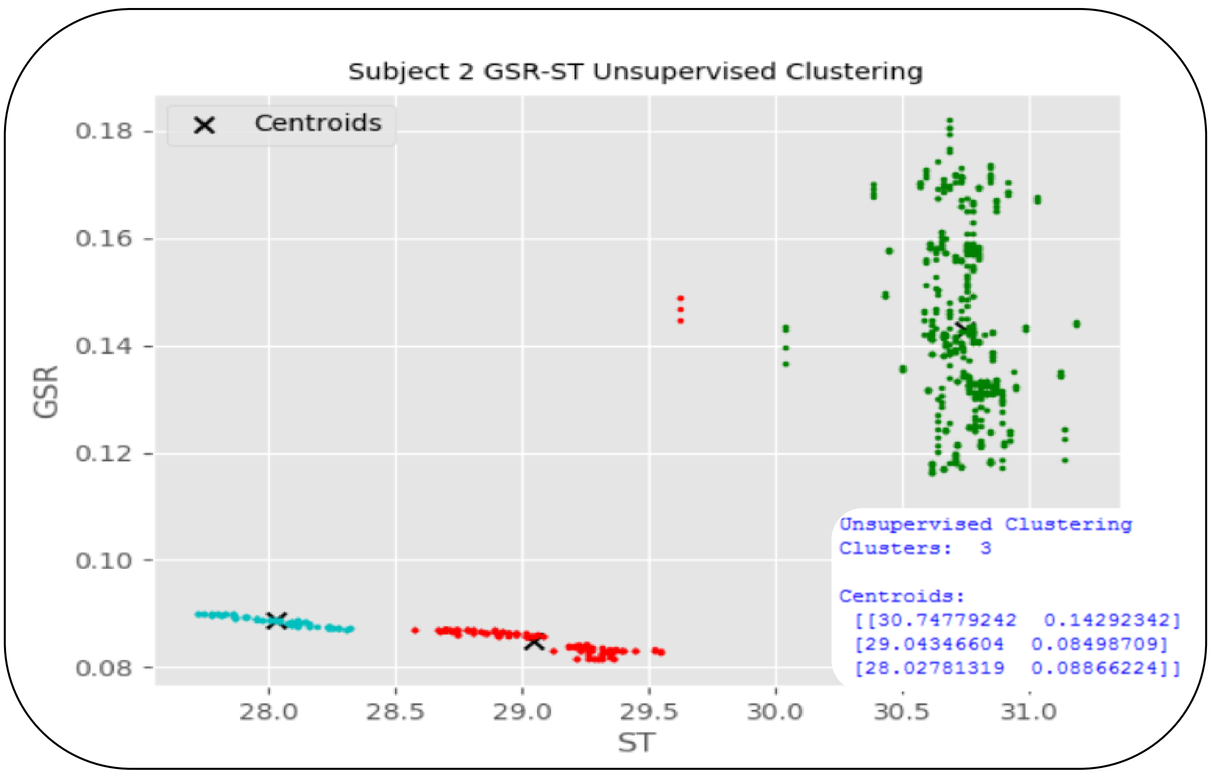


Figure 27: GSR – ST Unsupervised Clustering for Subject 2

Since the first two clusters have very much in common (especially regarding GSR), a K-Means supervised clustering (for each participant) was also executed, with predefined number of clusters equals to 2. To generate the supervised K-Means model we used Scikit-learn's library KMeans. Bellow, are presented the relevant script and scatter plot:

```
#Supervised Clustering
from sklearn.cluster import KMeans

clusters1 = 2
cl = KMeans(n_clusters=clusters1, max_iter = 100)
cl.fit(npcl)
centroids1 = cl.cluster_centers_
labels1 = cl.labels_

colors = 10*['g.','r.','c.','k.','m.','y.','b.','w.']

for i in range(len(npcl)):
    plt.plot(npcl[i][0], npcl[i][1], colors[labels1[i]], markersize = 3)
plt.scatter(centroids1[:,0], centroids1[:,1], marker='x', c='k', s=50, linewidths=10, label='Centroids')
plt.ylabel('GSR')
plt.xlabel('ST')
plt.title('Subject 1 GSR-ST Supervised Clustering', fontsize = 10)
plt.legend(loc='upper left')
print('Supervised Clustering\nClusters: ',clusters1,'\n')
print('Centroids: \n',centroids1)
plt.show()
```

Figure 28: Supervised K-Means Model with 2 Clusters Script

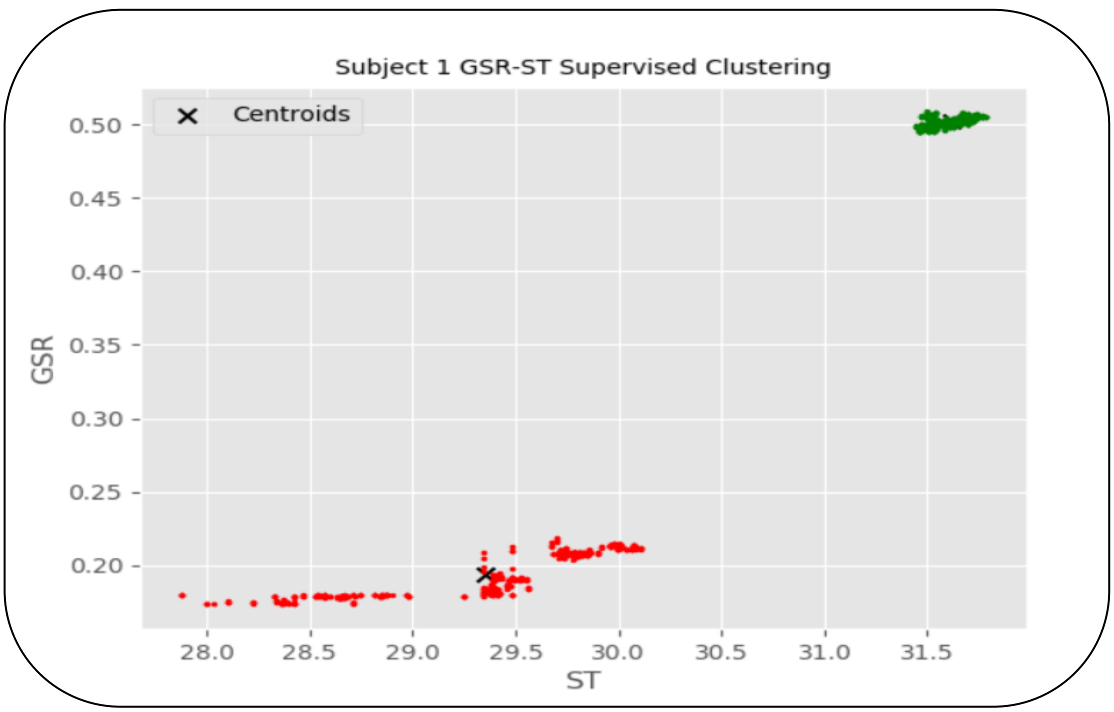


Figure 29: GSR – ST Supervised Clustering for Subject 1

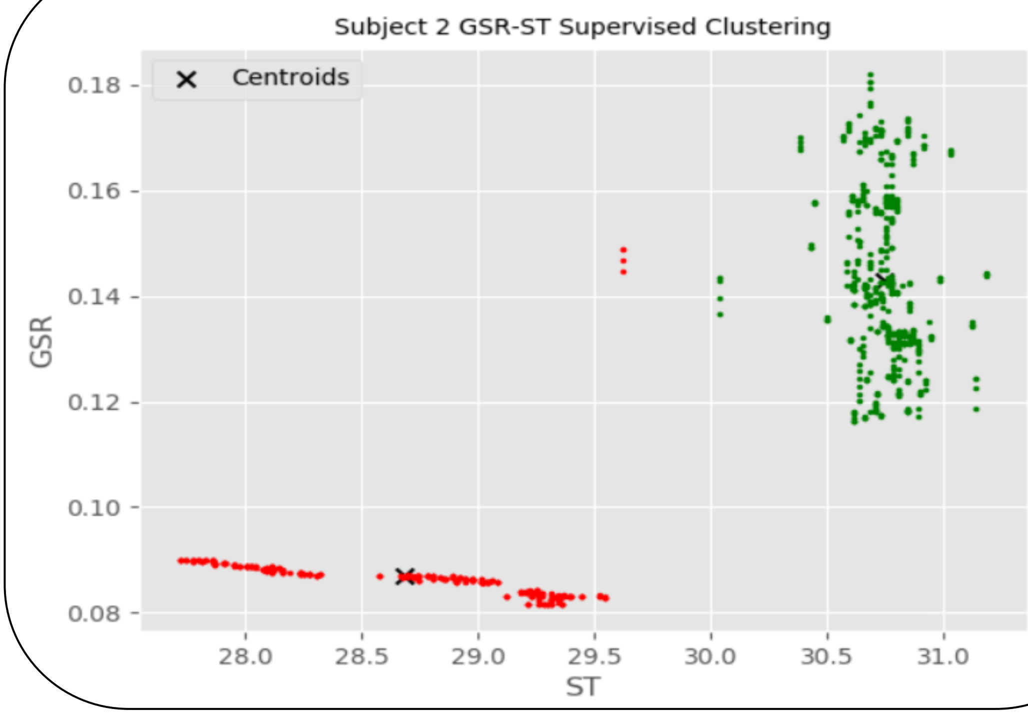


Figure 30: GSR – ST Supervised Clustering for Subject 2

In both scatter plots, we can easily comprehend that the third cluster consists of much higher GSR and ST values, than the other two. It is easier for us to understand the changes occurred in the specific measures when the subjects' emotional valence passed from positive to negative. That way, for each subject, every time we need to, depending on their recorded GSR and ST data, we are able to classify their preferences. Specifically, if GSR and ST data are high enough and belong to the second (green) cluster, we can assume that the subject does not like the music and gives a low rating. Of course, as it was mentioned above, music expectancy must be taken under consideration and further clustering should be implemented in the recorded data of an experiment with a randomized music list, to see if those changes will still occur.

To conclude, we used the supervised K-Means algorithm to investigate the relationship of GSR, ST and Ratings values in a three dimensional scatter plot, using all subjects' mean values at the same time. Bellow you can see the script used for the 3d plot.

```python
#GSR - ST - Ratings Means Scatter Plot

from mpl_toolkits.mplot3d import axes3d
ax.scatter(centroids1[:,0], centroids1[:,1], centroids1[:,2], c='k', marker='x', s=80, linewidths=1, label='Centroids')
#ax.scatter(centroids2[:,0], centroids2[:,1], centroids2[:,2], c='k', marker='x', s=80, linewidths=1, label='Centroids')
ax.scatter(gsr, st, np.log10(rat), c='c', marker='o', s=5, linewidths=1)
ax.set_xlim3d(0,max(gsr))
ax.set_ylim3d(0,max(st))
ax.set_zlim3d(17,19) #1e17,1e19
ax.set_xlabel('Ratings')
ax.set_ylabel('ST')
ax.set_zlabel('GSR')
ax.set_zlim(-0.02, 0.02)
ax.set_title('GSR-ST-Ratings Clustering', fontsize=12)
plt.legend(loc='upper left')
plt.show()
```

Figure 31: GSR-ST-Ratings Means 3d Scatter Plot Script

In the next figure you can see the relevant scatter plot. In its X Axis there are the subjects' means of GSR values of each music category, while in Y Axis there are the subjects' means of ratings for each category. Finally, the Z Axis contains the subjects' means of ST values of each music category.

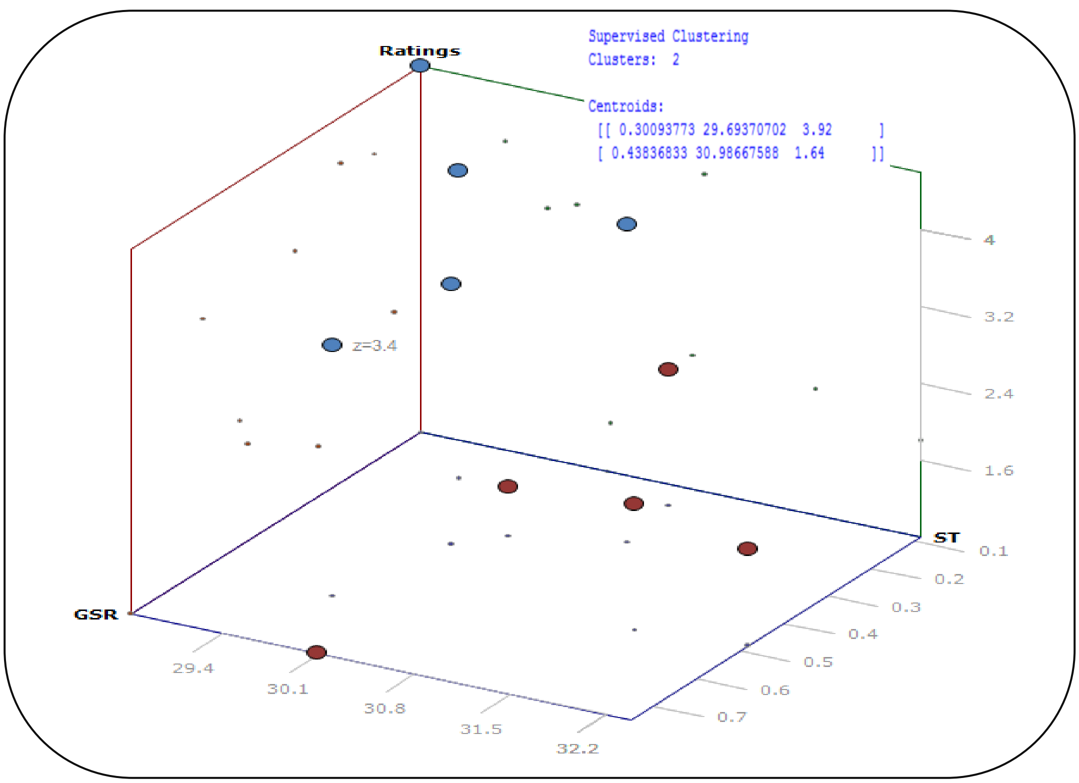


Figure 32: GSR-ST-Ratings Means 3d Scatter Plot for All Subjects

Our data inputs, were the ten data points (five for each music category), containing the GSR and ST means for each Rating mean. As we can observe in the figure 32, the algorithm divided those data points into two groups. The lower data group, consisting of the red observations, represents the values recorded when the subjects listened to their worst category, while the upper data group, consisting of the blue observations, represents the values recorded when the subjects listened to their favorite category, as the red observations have lower rating values (Ratings Centroid = 1.64) than the blue ones (Ratings Centroid = 3.92).

K Nearest Neighbor Algorithm

Having created two separate clusters with the K-Means model, next we will use K Nearest Neighbor (k-NN) Classification model to predict in which cluster (class), we are going to include a new observation. Specifically, based on our existing GSR-ST-Ratings data points, we are going to train the algorithm to decide whether or not a listener likes music, depending on both his recorded measures. For this purpose, we used K-NN model built-in function of Scikit-learn’s library. However, the accuracy of the model was 45%, which means that nearly 5 out of 10 data points will be included in the wrong cluster and by extension, the predicted rating for five users will be wrong. Such low prediction accuracy, is owed to the fact that the clusters are based on data points that represent the values of five different individuals, each one having a different measuring scale. For example, when someone’s skin temperature is considered ‘’high’’ for values bellow 30 °C and all the others’ ‘’high’’ temperature exceed 30 °C, an outlier will be formed that will confuse the model and decrease its predictive accuracy. Consequently, for predicting ratings based on GSR and ST, it is not efficient to use all subjects’ values at the same time.

For that reason, we run the model on each participant separately and surprisingly discovered that the accuracy exceeded 90% in all five cases. In the figures bellow, is presented the Python script, used to perform the algorithm for subject 1.

```
import numpy as np
import pandas as pd
import scipy
import sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn import neighbors
from sklearn import preprocessing
from sklearn.cross_validation import train_test_split
from sklearn import metrics
from sklearn.cluster import KMeans

#Subject 1

#Load data
df1 = pd.read_csv('.../Subject1.csv')
yb_GSR = dfb['GSR']
yb_ST = dfb['ST']

#Calculate GSR Means for every excerpt
mg1 = yb_GSR.iloc[0:88].mean()
mg2 = yb_GSR.iloc[89:190].mean()
mg3 = yb_GSR.iloc[190:248].mean()
mg4 = yb_GSR.iloc[249:390].mean()
mg5 = yb_GSR.iloc[391:539].mean()
mg6 = yb_GSR.iloc[540:626].mean()
mg7 = yb_GSR.iloc[627:760].mean()
mg8 = yb_GSR.iloc[761:820].mean()
mg9 = yb_GSR.iloc[821:955].mean()
mg10 = yb_GSR.iloc[956:1082].mean()
#Calculate ST Means for every excerpt
ms1 = yb_ST.iloc[0:88].mean()
ms2 = yb_ST.iloc[89:190].mean()
ms3 = yb_ST.iloc[190:248].mean()
ms4 = yb_ST.iloc[249:390].mean()
ms5 = yb_ST.iloc[391:539].mean()
ms6 = yb_ST.iloc[540:626].mean()
ms7 = yb_ST.iloc[627:760].mean()
ms8 = yb_ST.iloc[761:820].mean()
ms9 = yb_ST.iloc[821:955].mean()
ms10 = yb_ST.iloc[956:1082].mean()
```

Figure 33: Calculating Subjects' 1 GSR and ST Means for Every Excerpt Script

```
#Create a dictionary
d = {'GSR Means': [mg1, mg2, mg3, mg4, mg5, mg6, mg7, mg8, mg9, mg10],
     'ST Means': [ms1, ms2, ms3, ms4, ms5, ms6, ms7, ms8, ms9, ms10],
     'Ratings Means': [5, 2, 4, 4, 5, 2, 1, 1.5, 3.5, 2.5]}
#Convert to Data Frame
df = pd.DataFrame(d)
#Relocate Columns
df = df[['GSR Means', 'ST Means', 'Ratings Means']]
#Check data
print(df)

#Define K-Means Model
def kmeans_model(data, k):
    clusters = k
    cl = KMeans(n_clusters=clusters, max_iter = 100)
    cl.fit(data)
    centroids = cl.cluster_centers_
    labels = cl.labels_
    return labels

df_labels = pd.DataFrame(kmeans_model(df,2))

x = df.iloc[:,0:2]
y = df_labels

#Split data into Train & Test Set
X = preprocessing.scale(x)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 17)

#Train K-NN Model
clf = neighbors.KNeighborsClassifier()
clf.fit(X_train, y_train)
#print(clf)

#Evaluate Model with Test Set
y_actual = y_test
y_pred = clf.predict(X_test)

#print(metrics.classification_report(y_actual, y_pred))

accuracy = metrics.classification_report(y_actual, y_pred)
print('Accuracy Table: \n', accuracy)
```

Figure 34: K-NN Model Script

Regarding the script, after we loaded subjects' 1 data and calculated its measures means for each music excerpt, we used supervised K-Means, to create two clusters and convert ratings to labels (class 0 for high ratings / class 1 for low ratings). Then we defined the GSR and ST measures as the features the prediction will be based on (x value), and ratings as labels (y value). Afterwards, we shuffled the data and split them into test set (30% of total sample) and train set (the remaining 70%). It should be noted that during the train-test splitting procedure, we just divided the prime data set into the mentioned percentages, without using cross-validation method. Although in classification models such method is frequently suggested because it reduces overfitting, our experiment did not have enough data to support it. Finally we trained the K-NN model based on the train set and used test set to calculate its accuracy. The accuracy for this particular subject is 100%, which means that every observation of the test set was assigned with the correct label and consequently, the algorithm predicted successfully whether the subject liked the music excerpt or not.

All things considered, we confirmed that K-NN algorithm is way more efficient in its predictions when used for every participant separately, with an accuracy value nearly double, compared to the one of the all subjects' observations case.

Regression Algorithm

As long as we tested the prediction rate with K NN, we examined the implementation of the Regression model in the previous data and defined its accuracy. To begin with, a Regression model tries to describe the relationship between two or more values, by fitting an equation to observed data points. Those data points (or vectors) consist of a dependent variable (in our case Ratings means) and one or more independent variables (in our case GSR and ST means), which try to explain the dependent one. The best fit of the equation to the observations provide us with high accuracy predictions (note that the fit should not be very high, as overfitting issues emerge) (Rencher and Schaalje, 2008).

Like the K-NN model, first we examined the relationship between GSR, ST and Ratings means for all participants at the same time. At length, GSR and ST values will be the independent, explanatory variables and the Ratings values will be the dependent one. The linear regression model implemented through Python's StatsModels library. Part of the script, in conjunction with the results, are presented in the next two figures.

```
import statsmodels.api as sm

x = df_all.iloc[:,0:2]
y = df_labels

X = sm.add_constant(X)

#Model
model = sm.OLS(y,X).fit()
predictions = model.predict(X)

# Print out the statistics
print(model.summary())
```

Figure 35: StatsModels Regression Model with OLS Method Script

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      0   R-squared:                       0.551
Model:                            OLS   Adj. R-squared:                  0.422
Method:                 Least Squares   F-statistic:                     4.288
Date:                Wed, 20 Feb 2019   Prob (F-statistic):             0.0609
Time:                        18:11:44   Log-Likelihood:                -3.2589
No. Observations:                  10   AIC:                             12.52
Df Residuals:                       7   BIC:                             13.43
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.5000      0.127      3.947      0.006       0.200       0.800
x1            -0.0779      0.132     -0.591      0.573      -0.390       0.234
x2            -0.3416      0.132     -2.589      0.036      -0.654      -0.030
==============================================================================
Omnibus:                        0.387   Durbin-Watson:                   3.083
Prob(Omnibus):                  0.824   Jarque-Bera (JB):                0.456
Skew:                           0.021   Prob(JB):                        0.796
Kurtosis:                       1.955   Cond. No.                         1.33
==============================================================================
```

Figure 36: OLS Regression Results

The algorithm we used, applied Least Squares methods to find the best fit line for the given observations. The line (equation) created, had as goal to minimize the sum of squares of the errors (or residuals) generated from the differences in the observed (actual) values and the anticipated ones.

In the OLS (Ordinary Least Squares) Results we can see several parameters, like R-squared of the model, Adj. R-squared, F-Statistics etc. The R-squared figure (or Coefficient of Determination) will be used as a guideline to measure the model accuracy. Its value was 0.551, which means that GSR and ST variables explain only the

55% of the Ratings' observed variation. This figure is higher than the respective K-NN accuracy (0.45), but not high enough to make us consider Regression as an efficient model to describe those variables' relations.

Furthermore, we can see the Slopes of both independent variables with their respective P_Values and Confidence Intervals. Slope shows how much dependent variable (Ratings Means) is going to move, if this particular independent variable increased by 1 unit. GSR Means variable (x1), has $\text{Slope} = -0.0779$ and $\text{P\_value} = 0.573 > 0.05$ and therefore is not statistically significant. On the other hand ST Means variable (x2), with $\text{Slope} = -0.3416S$ and $\text{P\_value} = 0.036 < 0.05$ is statistically significant. As a consequence ST variables are better predictors for the dependent variable than the GSR ones.

Next, we run the same model on each participant separately and like the K-NN model, accuracy exceeded 90% in all five cases. As an example, we present Subject's 1 OLS Results in the figure bellow.

```
Subject 1
                            OLS Regression Results
==============================================================================
Dep. Variable:                      0   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.998
Method:                 Least Squares   F-statistic:                     2893.
Date:                Wed, 20 Feb 2019   Prob (F-statistic):           6.13e-11
Time:                        19:08:54   Log-Likelihood:                 26.335
No. Observations:                  10   AIC:                            -46.67
Df Residuals:                       7   BIC:                            -45.76
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.5000      0.007     76.117      0.000       0.484       0.516
x1             0.6036      0.027     22.643      0.000       0.541       0.667
x2            -0.1079      0.027     -4.048      0.005      -0.171      -0.045
==============================================================================
Omnibus:                        3.302   Durbin-Watson:                   2.017
Prob(Omnibus):                  0.192   Jarque-Bera (JB):                1.044
Skew:                           0.772   Prob(JB):                        0.593
Kurtosis:                       3.346   Cond. No.                         7.99
==============================================================================
```

Figure 37: OLS Regression Results of Subject 1

In this specific subject's case the R-squared is 0.999 and both GSR and ST Means variables have P_values nearly zero and therefore a possible change in them affect the dependent variable.

In addition, as depicted in the following script, we used Scikit-learn's LinearRegression function to calculate accuracy for each and every subject. The results were the same as the ones provided by the StatsModels library.

```
from sklearn.linear_model import LinearRegression

#Train Regression model
clf = LinearRegression()
clf.fit(X_train, y_train)
#print(clf)

#Accuracy
accuracy = clf.score(X_test, y_test)
print('Subject 1 Regression Accuracy = ', accuracy)
```

Figure 38: Scikit-learn Regression Model Script

Finally, we wanted to investigate each subject's Ratings relation with the respective GSR and ST Means separately. Instead of using some library's built in model like before, we created "manually" the Regression algorithm used to determine these relations, because we wanted to feed the algorithm with GSR or ST means and in response take a predicted rating. The next script presents the code used for that purpose.

```python
gsr = np.array([mg1, mg2, mg3, mg4, mg5, mg6, mg7, mg8, mg9, mg10], dtype=np.float64)
st = np.array([ms1, ms2, ms3, ms4, ms5, ms6, ms7, ms8, ms9, ms10], dtype=np.float64)
y = np.array([5, 2, 4, 4, 5, 2, 1, 1.5, 3.5, 2.5], dtype=np.float64)

def regression_model_prediction(xs, ys, pred_x, flag):
    slope = ( (mean(xs)*mean(ys)) - mean(xs*ys)) / ((mean(xs)*mean(xs)) - mean(xs*xs) )
    intercept = mean(ys) - slope*mean(xs)
    print('Slope = ', slope)
    print('Intercept = ', intercept)

    regression_line = [(slope*x)+ intercept for x in xs]

    pred_x = pred_x
    pred_y = (slope*pred_x)+intercept
    print('\nPredicted Rating = ', round(pred_y, 1))

    def squared_error(ys_actual, ys_predicted):
        return sum((ys_actual - ys_predicted)**2)

    def R2(ys_actual, ys_predicted):
        y_mean_line = [mean(ys_actual) for y in ys_actual]
        squared_error_regr = squared_error(ys_actual, ys_predicted)
        squared_error_y_mean = squared_error(ys_actual, y_mean_line)
        return 1 - (squared_error_regr / squared_error_y_mean)

    R_squared = R2(ys, regression_line)
    print('\nCoefficient of Determination = ', round(R_squared, 3))
    if flag == True:
        xp = np.linspace(0.15,0.52,100)
    else: xp = np.linspace(28,32,100)

    p1 = np.polyfit(xs,ys,1)
    plt.plot(xp, np.polyval(p1,xp),'r-',linewidth=1)

    p3 = np.polyfit(xs,ys,3)
    plt.plot(xp, np.polyval(p3,xp),'r--',linewidth=0.5)

    plt.scatter(xs,ys)
    plt.scatter(pred_x, pred_y, s=100, color='c')
    plt.ylabel('Ratings Means')
    plt.xlabel('GSR Means')
    plt.title('Subject 1 GSR Means - Ratings Scatter Plot', fontsize = 10)
    plt.legend(loc='upper left')
    plt.show()

regression_model_prediction(gsr, y, 0.48, True)
```

Figure 39: Manual Regression Model Script

For the calculation of Linear Regression Line we used the formula:
$y = Intercept + Slope *$
x, where y the dependent variable and x the independent variable.

For the Slope and Intercept we used: $Slope = \frac{mean(x)*mean(y) - mean(x*y)}{mean(x)^2 - mean(x^2)}$ and
$Intercept = mean(y) - Slope * mean(x)$ respectively.

The Coefficient of Determination calculation based on the formula:
$$R^2 = 1 - \frac{Error\ Sum\ of\ Squares}{Total\ Sum\ of\ Squares}$$

Regarding GSR Means – Ratings Relation

Initially, we trained our model with Subject's 1 GSR Means – Ratings observations. After that, to test its accuracy, we fed it with the GSR value of 0.48 which, based on our previous assumptions, is considered quite high. The predicted rating of the model was 2.3. The relative scatter plot is presented bellow.

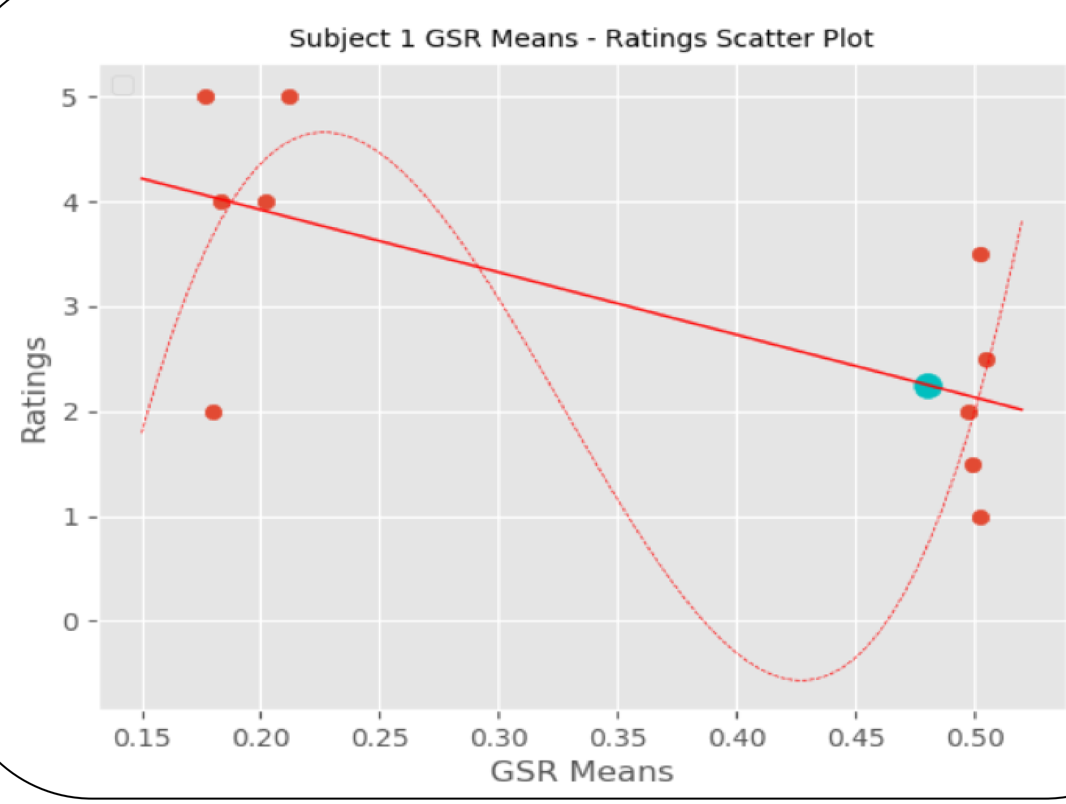


Figure 40: GSR Means – ST Scatter Plot of Subject 1, GSR Value Fed = 0.48

Afterwards, we fed the model with the GSR value of 0.16, which is considered pretty low and model's responded rating was 4.2.

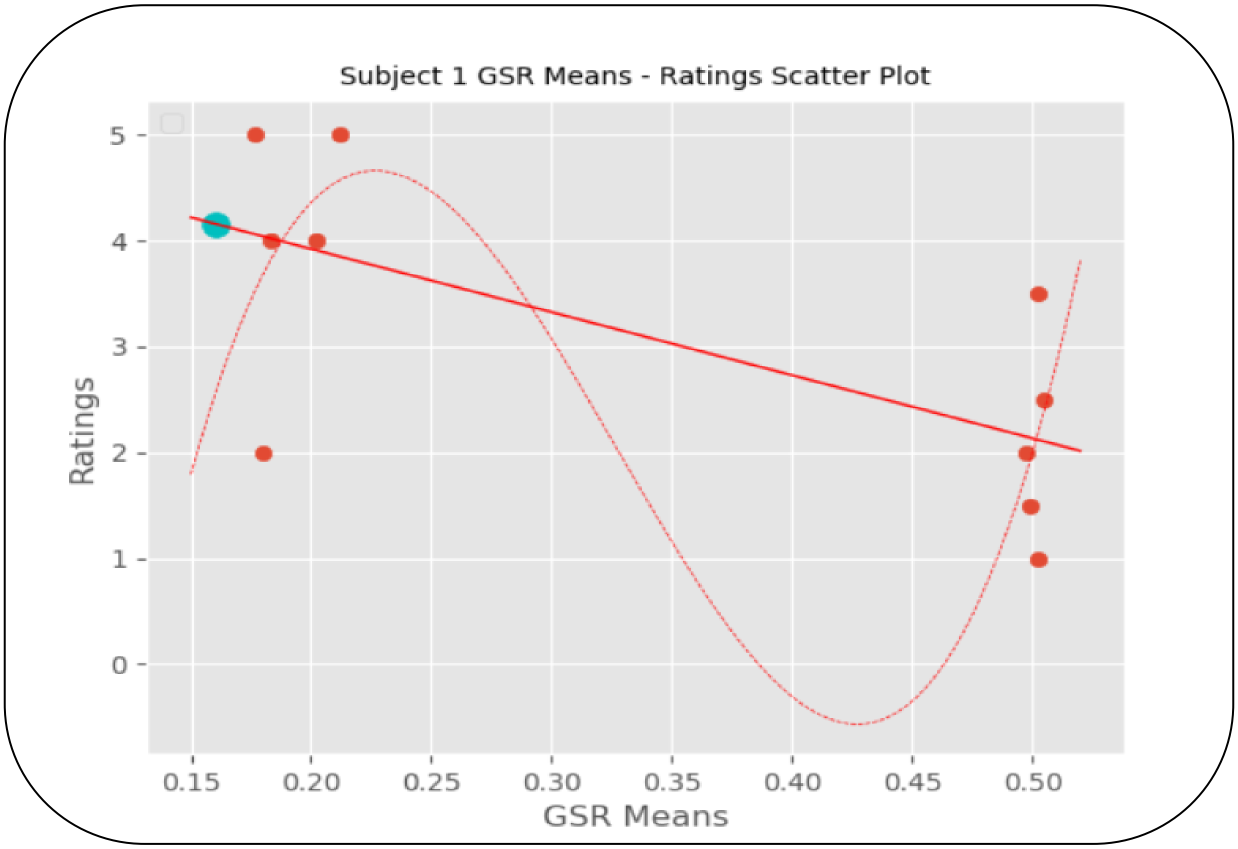


Figure 41: GSR Means – ST Scatter Plot of Subject 1, GSR Value Fed = 0.16

Regarding ST Means – Ratings Relation

Consequently, we trained our model with Subject's 1 ST Means – Ratings observations and fed it with the ST value of 31 which is considered a high one. The predicted rating was 2.7. The relative scatter plot is presented bellow.

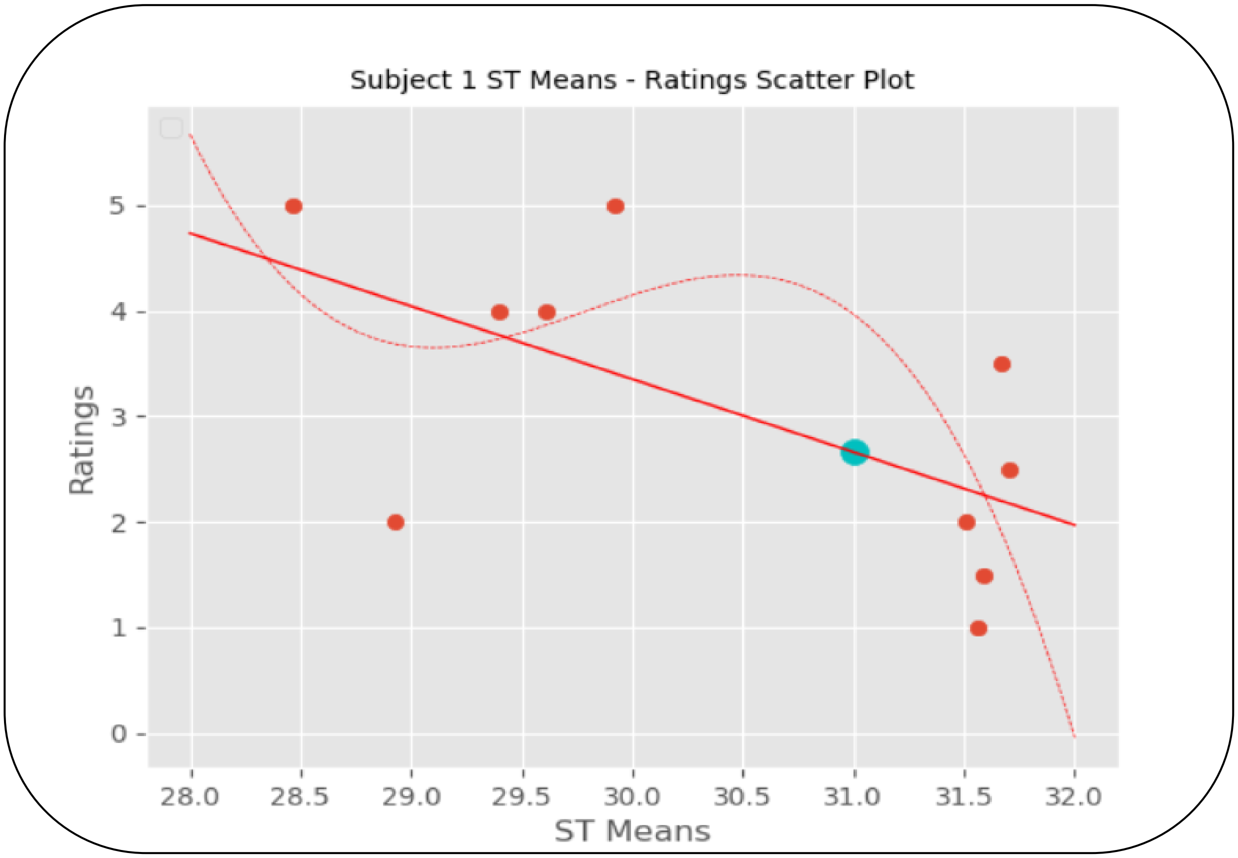


Figure 42: GSR Means – ST Scatter Plot of Subject 1, ST Value Fed = 31

Afterwards, we fed the model with the ST value of 29, which is considered low and the predicted rating was 4.

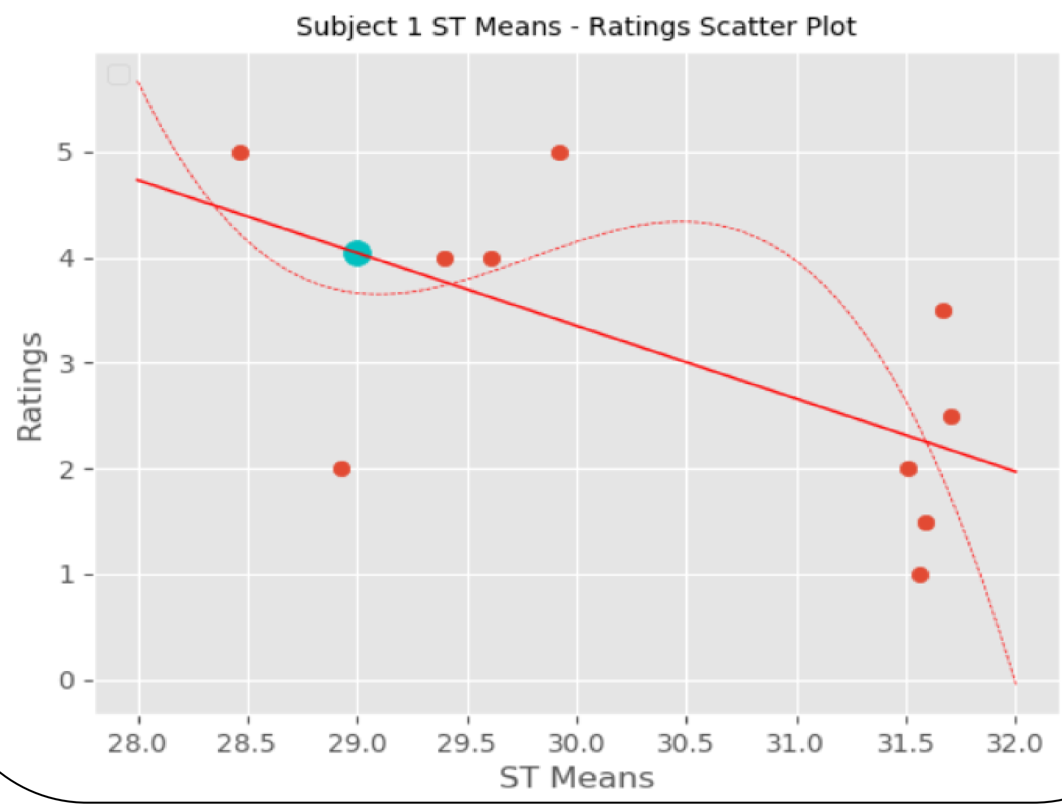


Figure 43: GSR Means – ST Scatter Plot of Subject 1, ST Value Fed = 29

After the feeding of various other values (both GSR and ST) to our model for all subjects, we concluded that its prediction rate is sufficient enough to be considered as a good predictive algorithm for the ratings of our subjects. However, because of the fact that the quantity of the initial observations, in all five cases, was low, we cannot be certain yet about model's efficiency. We have to use it with more data, examine again its accuracy and conclude to whether it is useful or not.

Neural Networks Algorithm

With the term Neural Network (NN) we refer to a machine learning model, which is based on human brain's functions. More precisely, this model creates an artificial neural network that uses different layers of mathematical processing to make sense of the information that it was fed. The input layer receives the input data which represent the observations used to train the model. Next, the observations are transferred to the hidden layer, which, through some mathematical functions, transform them and pass them into the output layer. There are neural networks which use more than one hidden layer and in that case, they are called deep learning models. Each layer is connected to another through weights. The higher the weights the higher the influence one layer has on another connected one. Neural networks are used to solve various problems and predict outcomes using huge amounts of data, fast and efficiently. Additionally, one of their biggest advantages is that they adapt to changing input, so they always generate the best possible outcomes without needing to be redesigned or alter the output criteria (Kriesel, 2007).

In Python, the most common way to construct and run a neural network model, is through the open source library TensorFlow which, instead of matrixes to represent data, uses tensors. As opposed to many other machine learning algorithms, neural networks' prediction efficiency highly depends on the amount of input data. The higher the data quantity, the higher the predictive accuracy. In our study, inputs were very few in number and consequently such algorithms are not suggested for predictions. However we tried a neural network with one hidden layer which takes as inputs a subject's measures of GSR and ST Means and return a prediction of whether or not the subject liked the song. The used TensorFlow script is depicted bellow.

```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.preprocessing import LabelEncoder
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')

def read_dataset():
    df = pd.read_csv('.../Subject1_gsr_st_ratings.csv')
    X = df[df.columns[0:2]].values #Features
    y = df[df.columns[2]] #Label (test data)

    #Encode the dependent var (y - Label)
    encoder = LabelEncoder()
    encoder.fit(y) #encode the ratings to 0 for Not Like & 1 for Like (boolean)
    y = encoder.transform(y)
    Y = one_hot_encode(y) #one_hot encoding: Not Like->[1,0], Like->[0,1]
    #print(X.shape)
    return(X, Y)

def one_hot_encode(labels):
    n_labels = len(labels)
    n_unique_labels = len(np.unique(labels))
    one_hot_encode = np.zeros((n_labels, n_unique_labels))
    one_hot_encode[np.arange(n_labels), labels] = 1
    return one_hot_encode

#Read & shuffle the data set:
X, Y = read_dataset()
X, Y = shuffle(X, Y, random_state=1)

#Convert the dataset into train and test sets:
train_x, test_x, train_y, test_y = train_test_split(X, Y, test_size=0.20, random_state=415)

#Parameters
lr = 0.01
hm_epochs = 10 #number of iterations that will be done in order to minimize the error
cost_history = np.empty(shape=[1], dtype=float) #loss function
n_dim = X.shape[1] #the number of columns of X var (features), the values that we input into the model
print('n_dim :', n_dim)
n_class = 7
model_path = '...//Model' #where do we store the model
```

Figure 44: Neural Network Model with TensorFlow Script Part 1

```
#Hidden Layers:
n_nodes_hl1 = 5

#Tensor parameters:
x = tf.placeholder(tf.float32,[None, n_dim]) #input values->they're gonna be fed with the data set
w = tf.Variable(tf.zeros([n_dim, n_class])) #tf.zeros -> initialized with zeros
b = tf.Variable(tf.zeros([n_class]))
y_ = tf.placeholder(tf.float32, [None, n_class]) #actual values (known data)

#Define the model:
def nn_model(x):
    hidden_1_layer = {'weights':tf.Variable(tf.random_normal([n_dim, n_nodes_hl1])),
                      'biases':tf.Variable(tf.random_normal([n_nodes_hl1]))}

    output_layer = {'weights':tf.Variable(tf.random_normal([n_nodes_hl1, n_class])),
                    'biases':tf.Variable(tf.random_normal([n_class]))}

    # (input_data * weights) + biases

    layer_1 = tf.add(tf.matmul(x, hidden_1_layer['weights']), hidden_1_layer['biases'])
    layer_1 = tf.nn.sigmoid(layer_1) # sigmoid the (input_data * weights) + biases

    output_layer = tf.matmul(layer_1, output_layer['weights']) + output_layer['biases']

    return output_layer


#Initialize all variables:
init = tf.global_variables_initializer()
#saver = tf.train.Saver() #Saver object in order to save our model

#Call your model
y = nn_model(x)

#cross_entropy_with_logits as cost function: the difference between the prediction and the y data (actual_data*w + b)
#logits -> prediction (nn output data)
#labels -> known data
cost_function = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(logits = y, labels = y_)) #logits = y, labels = y

#stochastic gradient descent
training_step = tf.train.GradientDescentOptimizer(lr).minimize(cost_function)
```

Figure 45: Neural Network Model with TensorFlow Script Part 2

```python
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    #Calculate cost & accuracy for each epoch:
    mse_history = []
    accuracy_history = []
    for epoch in range(hm_epochs):
        sess.run(training_step, feed_dict={x: train_x, y_: train_y})
        cost = sess.run(cost_function, feed_dict={x: train_x, y_: train_y})
        cost_history = np.append(cost_history, cost)
        #tf.argmax -> returns the index of the max value
        #The difference between the actual and the model output (the prediction)
        correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
        accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
        #print('Accuracy: ',(sess.run(accuracy, feed_dict={x: test_z, y_: test_y})))
        pred_y = sess.run(y, feed_dict={x: test_x})
        mse = tf.reduce_mean(tf.square(pred_y - test_y))
        mse_ = sess.run(mse)
        mse_history.append(mse_)
        accuracy = (sess.run(accuracy, feed_dict={x: test_x, y_: test_y}))
        accuracy_history.append(accuracy)

        print(epoch+1,' epoch completed out of ', hm_epochs,'\nCost: ',cost,' / MSE: ',mse_,'/ Accuracy: ',accuracy,'\n')

        #Print the final accuracy & mse:
        if epoch+1 == hm_epochs:
            accuracy_ = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
            print('Model Accuracy: ',(sess.run(accuracy_, feed_dict={x: test_x, y_: test_y})))
            print('Final MSE: %4f' % sess.run(mse))
    save_path = saver.save(sess, model_path)
    print('\nModel saved in file: %s' % save_path)
```

Figure 46: Neural Network Model with TensorFlow Script Part 3

As expected, the prediction accuracy of the model was close to zero. No matter how much we were altering model's parameters (such us epochs and learning rate) or adding and subtracting new hidden layers the results were always the same. To solve that issue, we decided to construct our on model, with much simpler structure, able to predict with an adequate precision users' preferences. The following figure depicts this particular code.

```python
#Load data
df = pd.read_csv('C:/det/1st_test/Bill_gsr_st_ratings.csv')

#Label target data
def label(data, k):
    clusters = k
    cl = KMeans(n_clusters=clusters, max_iter = 100)
    cl.fit(data)
    centroids = cl.cluster_centers_
    labels = cl.labels_
    return labels

df_labels = pd.DataFrame(label(df,2))
gsr = df.iloc[:,0:1]
st = df.iloc[:,1:2]

#Merge
df = gsr.merge(st,left_index=True,right_index=True,suffixes=(1,2)).merge(df_labels,left_index=True,right_index=True)
df.rename(columns={'01':'GSR Means','02':'ST Means',0:'Ratings'}, inplace=True)

data = np.array(df).tolist()

# Our Neural Network Structure:

#   o   Output(type)
#  / \  Weigths(w1, w2) & Bias(b)
# o   o Inputs(length, width)

w1 = np.random.rand()
w2 = np.random.rand()
b = np.random.rand()

def sigmoid(x):
    return 1/(1 + np.exp(-x))

#Sigmoid derivative (prime):
def sigmoid_p(x):
    return sigmoid(x)*(1 - sigmoid(x))
```

Figure 47: Manual Neural Network Script Part 1

```
#Training Loop:
w1 = np.random.rand()
w2 = np.random.rand()
b = np.random.rand()

data = np.array(df).tolist()

step = 0.2

costs = []
for i in range (50000):
    ri = np.random.randint(len(data)) #Random Index
    #point = data[ri]
    z = data[ri][0]*w1 + data[ri][1]*w2 + b #NN Output, based on inputs (len, wid)
    prediction = sigmoid(z) #NN sigmoid Output
    target = data[ri][2]
    cost = np.square(prediction - target)
    #print("For flower with measurements :", data[ri], ", the cost is: ",round(cost,3), "!")

    #Derivatives:

    #Derivative of cost with respect to prediction:
    dcost_pred = 2*(prediction-target)
    #Derivative of prediction with respect to z (data):
    dpred_z = sigmoid_p(z) #Has been calculated previously!
    #Derivative of z with respect to w1:
    dz_w1 = data[ri][0]
    #Derivatibe of z with respect to w2:
    dz_w2 = data[ri][1]
    #Derivatibe of z with respect to b:
    dz_b = 1

    #Derivative of cost with respect to w1:
    dcost_w1 = dcost_pred * dpred_z * dz_w1
    #Derivative of cost with respect to w2:
    dcost_w2 = dcost_pred * dpred_z * dz_w2
    #Derivative of cost with respect to b:
    dcost_b = dcost_pred * dpred_z * dz_b

    w1 = w1 - step*dcost_w1
    w2 = w2 - step*dcost_w2
    b = b - step*dcost_b
```

Figure 48: Manual Neural Network Script Part 2

```
mf = [0.5, 31]

z = mf[0]*w1 + mf[1]*w2 + b
prediction = sigmoid(z)
if prediction > 0.5:
    col = 'does not like'
    mf.append(1)
else:
    col = 'likes'
    mf.append(0)
print('NN Prediction: ', round(prediction,6))
print('\nThe subject with GSR = ', mf[0],'and ST = ',mf[1],'', col, ' the song')
```

Figure 49: Manual Neural Network Script Part 3

Our aim was to construct a model which will be fed with the subject's measures of GSR and ST and return values from a 0 to 1 scale. The values close to 0 will indicate that the subject liked the song, while values close to 1 that the subject did not.

First we loaded and transformed the data and labeled the ratings column to values 0 for low ratings and 1 for high ratings. Then we defined our model which has 1 hidden layer and is connected to input layer with two randomly generated weights and one bias value. The connection between the two layers was established through the mathematical linear function: $GSR * weigth1 + ST * weight2 + bias$. Then a sigmoid function was constucted, based on the formula: $sigmoid(x) = \frac{1}{1+e^{-x}}$, where x replaced with the previous linear function. Consequently, we created a hidden layer that take measures from the input layer, transform them into the desired 0 to 1 scale values (with the sigmoid function) and feed them to the output layer. The role of the bias is to prevent the outcome values from being totally depended on the weights, for there are cases in which weights, randomly, may take as initial value the number zero. In such cases, in the absence of a constant (bias), all outcomes would have been zero.

Finally, we fed the created model with the measures 0.5 and 31 for GSR and ST respectively. The prediction was that the subject with these particular values will not like the song. That outcome is in total agreement with the study's previous results, as both mentioned GSR and ST values are considered pretty high.

```
NN Prediction:  1.0

The subject with GSR =  0.5 and ST =  31  does not like  the song
```

Figure 50: NN Prediction Results for GSR, ST = [0.5, 31]

On the other hand, when we fed our model with the values 0.2 and 28, the respond was that the subject likes the songs, which is also compatible with our experiment's results, as both values are considered to be among the lower ones.

```
NN Prediction:  1.0239514842385295e-16

The subject with GSR =  0.2 and ST =  28  likes  the song
```

Figure 51: NN Prediction Results for GSR, ST = [0.2, 28]

CONCLUSION

Throughout the study, we have exported several graphs (either time series or scatter plots) to determine the relationship between participants' recorded physiological measures of GSR, ST and BPM, as well as the existed correlation with the Ratings. Additionally, a few machine learning algorithms were preformed, not only to identify the mentioned correlations, but also to provide us with an answer in our query: can we predict users' ratings solely from these particular measures? In other words, can we consider GSR and ST values as implicit ratings? Drawing a conclusion is not that simple, as, although our study results headed us towards the answer yes, there are still many questions that remain unanswered and many factors that need to be considered and further examined in order for us to be able to give an ultimate conclusion.

The first confirmed outcome, after analyzing the appropriate graphs and calculating the relevant correlations, was that there is a strong negative relationship between subjects' both GSR and ST values and their respective ratings. Whenever the subjects were listening to the music category they disliked, their mentioned values were rapidly increased. The question here is, to what extent those increases were marked due to the subjects' averseness of the songs they were hearing and to what due to the averseness of the songs they were expecting to hear. To paraphrase the question, how much (if any) the music expectancy phenomenon influenced their recorded measures? As mentioned before, to reply to this question we should design and conduct another experiment, in which the music excerpts will being played in randomized order, so as the subjects not knowing in advance what they are going to listen. Furthermore, the quantity of the listeners was quite small and therefore, the experiment is not considered scientifically adequate and we cannot render its results as proven. For that reason the conduct of an additional experiment, with a higher number of participants, is of primary importance.

As far as the machine learning models applied are concerned, we measured their accuracies and concluded that predictions should be made for each and every user separately, in order to achieve high model efficiency. Between the used models, K-NN and Regression are considered the ones with the highest efficiency as they had bigger accuracies. Neural Network is probably useful too, but we need to test it against a much bigger amount of input data to be sure of its competence.

Overall, based exclusively on this study's available data, it may be argued that the valence and the arousal of a user's perceived emotion, expressed by the measures of GSR and ST, can be replace its ratings and therefore considered as implicit rating data. That is of course with the use of the relevant sensors, recording those measures and with the appropriate machine data processing. Despite of the fact that a bigger number of participants is needed and the music expectancy bias, the evidences of the negative measures – ratings correlation are strong. These evidences, as well as the created prediction algorithms, signify that recorded data from a wristband can be converted into ratings, or at least into an indication of users' likes and dislikes. If those findings are also verified in a bigger data scale, a new era in recommender systems will emerge, as acquiring individuals' preferences, without asking to provide them, will no longer be a difficult task. Of course Internet of Things technology has to be further expanded and established as one of the main trends, but that is just a matter of time, as IoT already considered as the future of connectivity.

REFERENCES

Ayodele, T. (2010). *Types of Machine Learning Algorithms.* United Kingdom: University of Portsmouth. Available: doi: 10.5772/9385.

Barrett, L.F. and Wager, T.D. (2006). The Structure of Emotion Evidence From Neuroimaging Studies. *Current Directions in Psychological Science*. 15(2), 79-83. Available: https://doi.org/10.1111/j.0963-7214.2006.00411.x

Blood, A.J. and Zatorre, R.J. (2001). Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion. *Proceedings of the National Academy of Sciences of the United States of America*. 98(20), 11818-11823. Available: DOI: 10.1073/pnas.191355898

Bobrowski, P. and Knipschild, A. (2016). Effects of Music on Emotion, Heart Rate, Respiration, and Electrodermal Activity. *National Conference On Undergraduate Research.*

Boucsein, W. (2012). *Electrodermal Activity.* 2 nd edition. Germany: Springer

Bovell, D. (2015). The human eccrine sweat gland: Structure, function and disorders. *Journal of Local and Global Health Science*. 1(5). Available: doi: 10.5339/jlghs.2015.5.

Burke, R.D., Felfernig, A., Göker, M.H. (2011). Recommender Systems: An Overview. *Ai Magazine*. 32, 13-18. Available: doi: 10.1609/aimag.v32i3.2361.

Christopoulos, G.I., Uy, M.A., Yap, W.J. (2016). The Body and the Brain: Measuring Skin Conductance Responses to Understand the Emotional Experience. *Organizational Research Methods*. 22(1), 394-420. Available: https://doi.org/10.1177/1094428116681073

Eerola, T. and Vuoskoski, J.K. (2011). A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*. 39(1), 18-49. Available: https://doi.org/10.1177/0305735610362821

Eerola, T. (2003). *The dynamics of musical expectancy: cross-cultural and statistical approaches to melodic expectations.* Jyväskylä: University of Jyväskylä

Ferdinando, H., Ye, L., Seppänen, T., Alasaarela, E. (2014) Emotion Recognition by Heart Rate Variability. *Australian Journal of Basic and Applied Science*. 8(10), 50-55. Available: https://www.researchgate.net/publication/267212566_Emotion_Recognition_by_Heart_Rate_Variability

Gabrielsson, A. (2001). Emotion perceived and emotion felt: Same or different? Musicae Scientiae. 5(1), 123-147. Available: https://doi.org/10.1177/10298649020050S105

Garlin, F.V. and Owen, K. (2006). Setting the tone with the tune: A meta-analytic review of the effects of background music in retail settings. *Journal of Business Research.* 59(6), 755-764. Available: doi: 10.1016/j.jbusres.2006.01.013.

Greco, A., Lanata, A., Citi, L., Vanello, N., Valenza, G., Scilingo, E.P. (2016). Skin Admittance Measurement for Emotion Recognition: A Study over Frequency Sweep. *Electronics.* 5(46). Available: doi: 10.3390/electronics5030046

Hanser, W. and Mark, R.E. (2013). Music Influences Ratings of the Affect of Visual Stimuli. *Psihologijske Teme*. 22(2), 305-324. Available: https://www.researchgate.net/publication/279527090_Music_Influences_Ratings_of_the_Affect_of_Visual_Stimuli

Huang, C. and Yin, J. (2010). Effective association clusters filtering to cold-start recommendations. *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*. 5, 2461-2464. Available: DOI: 10.1109/FSKD.2010.5569294

Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation.* Cambridge, MA, UK: The MIT Press.

Huron, D. and Margulis, E.H. (1993). Musical Expectancy and Thrills. Handbook of Music and Emotion: Theory, Research, Applications. *ResearchGate.* 21, 575-604. Available: doi: 10.1093/acprof:oso/9780199230143.003.0021.

Juslin, P.N. and Västfjäll, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behav Brain Sci.* 31(5), 559–621. Available: doi:10.1017/S0140525X08005293

Kowalska, M. and Wróbel, M. (2017). Basic Emotions. *Encyclopedia of Personality and Individual Differences.* Available: DOI: 10.1007/978-3-319-28099-8_495-1

Kriesel, D. (2007). *A Brief Introduction to Neural Networks.* Available: http://www.dkriesel.com

Lindquist, K.A., Wager, T.D., Kober, H., Bliss-Moreau, E., Barrett, L.F. (2012). The brain basis of emotion: a meta-analytic review. *The Behavioral and brain sciences.* 35(3), 121-143. Available: doi: 10.1017/S0140525X11000446

Lundqvist, L.O., Carlsson, F., Hilmersson, P., Juslin, P.N. (2009). Emotional responses to music: experience, expression, and physiology. *Psychology of Music*. 37(1), 61-90. Available: https://doi.org/10.1177/0305735607086048

McFarland, R.A. and Kadish, R. (1991). Sex differences in finger temperature response to music. *Int J Psychophysiol*. 11(3), 295-298. Available: https://doi.org/10.1016/0167-8760(91)90024-R

McFarland, R.A. (1985). Relationship of skin temperature changes to the emotions accompanying music. *Biofeedback & Self Regulation*. 10(3), 255-267. Available: http://dx.doi.org/10.1007/BF00999346

Mittelmann, B. and Wolff, H.G. (1943). Emotions and skin temperature: observations on patients during psychotherapeutic (psychoanalytic) interviews. *Psychosomatic Medicine. 5,* 211-231. Available: http://dx.doi.org/10.1097/00006842-194307000-00002

Morissette, L. and Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology.* 9(1), 15-24. Available: DOI: 10.20982/tqmp.09.1.p015

Nichols, D.M. (1998). Implicit Rating and Filtering. *Lancaster University.* Available: https://www.ercim.eu/publication/ws-proceedings/DELOS5/nichols.pdf

Omran, M.G., Engelbrecht, A.P., Salman, A.A. (2007). An overview of clustering methods. *Intell. Data Anal*. 11(6), 583-605. Available: DOI: 10.3233/IDA-2007-11602

Papagelis, M., Plexousakis, D., Kutsuras, T. (2005) Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. *Springer.* 224-239. Available: DOI: http://dx.doi.org/10.1007/11429760_16

Posner, J., Russell, J.A., Peterson, B.S. (2005). The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Dev Psychopathol.* 17(3), 715-734. Available: doi: 10.1017/S0954579405050340

Reisenzein, R. (2007). What is a definition of emotion? And are emotions mental-behavioral processes?. *Social Science Information*. 46(3), 424-428. Available: doi: 10.1177/05390184070460030110.

Rencher, A.C. and Schaalje, G.B. (2008). *Linear Models in Statistics.* 2 nd edition. New Jersey: John Wiley & Sons, Inc.

Richter, M. and Wright, R.A. (2012). *Autonomic Nervous System.* New York: Springer.

Rosebrock, L.E., Hoxha, D., Norris, C., Cacioppo, J.T., Gollan, J.K. (2017). Skin conductance and subjective arousal in anxiety, depression, and comorbidity: Implications for affective reactivity. *Journal of Psychophysiology*. 31(4), 145-157. Available: http://dx.doi.org/10.1027/0269-8803/a000176

Sachan, A. and Richhariya, V. (2013). Reduction of Data Sparsity in Collaborative Filtering based on Fuzzy Inference Rules. *International Journal of Advanced Computer Research.* 3(10). Available: https://pdfs.semanticscholar.org/cfcb/3fb182a421b9540c6c61ff0c077421893c1d.pdf

Salimpoor, V.N., Benovoy, M., Longo, G., Cooperstock, J.R., Zatorre, R.J. (2009). The Rewarding Aspects of Music Listening Are Related to Degree of Emotional Arousal. *PLoS One.* 4(10), e7487. Available: doi: 10.1371/journal.pone.0007487

Scherer, K. (2009). The dynamic architecture of emotion: Evidence for the component process model. *Cognition and Emotion*. 23(7), 1307-1351. Available: doi: 10.1080/02699930902928969.

Sentio Solutions Inc. (2018). *Myfeel.* Sentio Solutions Inc. Available: https://www.myfeel.co/

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* New York: Cambridge University Press.

Sloboda, J.A. (1991). Music Structure and Emotional Response: Some Empirical Findings. *Psychology of Music*. 19(2), 110-120. Available: https://doi.org/10.1177/0305735691192002

Vos, P., De Cock, P., Munde, V., Petry, K., Van Den Noortgate, W., Maes, B. (2012). The tell-tale: What do heart rate; skin temperature and skin conductance reveal about emotions of people with severe and profound intellectual disabilities?. Res Dev Disabil. 33(4), 1117–1127. Available: doi: 10.1016/j.ridd.2012.02.006

Wang, H.M. and Huang, S.C. (2014). Musical Rhythms Affect Heart Rate Variability: Algorithm and Models. *Advances in Electrical Engineering,* 2014(851796). Available: https://doi.org/10.1155/2014/851796

Zentner, M., Grandjean, D., Scherer, K.R. (2001). Emotions Evoked by the Sound of Music: Characterization, Classification, and Measurement. *Emotion.* 8(4), 494–521. Available: doi: 10.1037/1528-3542.8.4.494

APPENDIX 1: Music Excerpts' List shorted by Music Category

Musical Excerpt Title	Artist/s	Duration (mins)	Music Category
Say My Name	David Guetta, Bebe Rexha & J Balvin	2.07	Dance
Lost In Istanbul	Brianna	2.07	Dance
6 Days	Mahmut Orhan & Colonel Bagshot	1.35	Dance
Resurrection	PPK	2.27	Dance
Lions In The Wild	Martin Garrix & Third Party	1.47	Dance
Tomame	Ελένη Φουρέιρα	1.21	Greek Pop
Όσο Θα Λείπεις	KINGS & Γιάννης Πλούταρχος	2.12	Greek Pop
80s	Locomondo	2.09	Greek Pop
Γιατί	Μέλισσες	1.54	Greek Pop
Καραμέλα Αιγαίο	Ελένη Φουρέιρα	1.58	Greek Pop
Nemesis	Arch Enemy	1.39	Hard Metal
Hells Bells	ACDC	2.08	Hard Metal
Master Of Puppets	Metallica	3.08	Hard Metal
Raining Blood	Slayer	1.28	Hard Metal
Paikiller	Judas Priest	2.12	Hard Metal
I Like It	Cardi B, Bad Bunny & J Balvin	2.08	Pop
Havana	Camila Cabello	1.16	Pop
Girls Like You	Maroon 5	1.5	Pop
X Equis	Nicky Jam & J Balvin	1.28	Pop
New Rules	Dua Lipa	1.1	Pop
Ferrari	SNIK	1.36	Rap
Rockstar	Post Malone & 21 Savage	1.46	Rap
Not Afraid	Eminem	1.34	Rap
Humble	Kendrick Lamar	1.18	Rap
Gucci Mane	Migos & Slippery	1.59	Rap
Hotel California	Eagles	2.44	Rock
Sweet Child 'O Mine	Guns N' Roses	2.04	Rock
House Of The Rising Sun	The Animals	1.11	Rock
The Trooper	Iron Maiden	2	Rock
Wind Of Change	Scorpions	2.05	Rock
Οι Ψυχές Και Οι Αγάπες	Βασίλης Παπακωνσταντίνου	2.23	Έντεχνο
Κακές Συνήθειες	Μιλτιάδης Πασχαλίδης	1.41	Έντεχνο
Σου Μιλώ Και Κοκκινίζεις	Γιάννης Χαρούλης	2.02	Έντεχνο
Μ' Αρεσει Να Μη Λεώ Πολλά	Υπόγεια Ρεύματα	1.55	Έντεχνο
Εν Λευκώ	Νατάσσα Μποφίλιου	1.54	Έντεχνο
Τώρα Τι Να Το Κάνω	Νίκος Οικονομόπουλος	1.16	Λαϊκά
Απορώ	Βασίλης Καρράς	1.27	Λαϊκά
Ξημερώματα	Κωνσταντίνος Αργυρός	1.26	Λαϊκά
Γύφτισσα Μέρα	Νότης Σφακιανάκης	2.39	Λαϊκά
Στην Καρδιά	Θέμης Αδαμαντίδης	1.21	Λαϊκά

APPENDIX 2: Subjects' GSR, ST and Ratings Means per Music Category

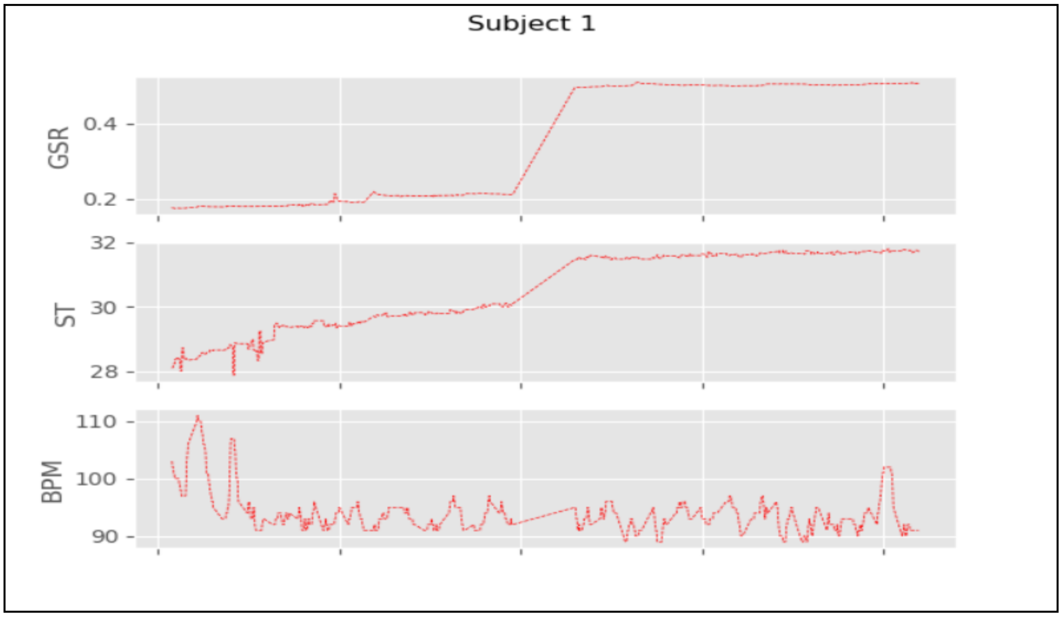
GSR Means	ST Means	Ratings Means	Category Type
0.087180206	28.73566723	4.6	Favorite
0.143430744	30.72967273	2.2	Worst
0.267245803	30.81849834	4.1	Favorite
0.318051935	31.41874105	1.3	Worst
0.193700199	29.35034217	4	Favorite
0.501408541	31.61167393	2.1	Worst
0.581281986	29.67568745	3.4	Favorite
0.748498907	30.09291653	0.8	Worst
0.375280448	29.88833993	3.5	Favorite
0.480451506	32.38037516	1.8	Worst

APPENDIX 3: Subjects' Ratings per Music Category

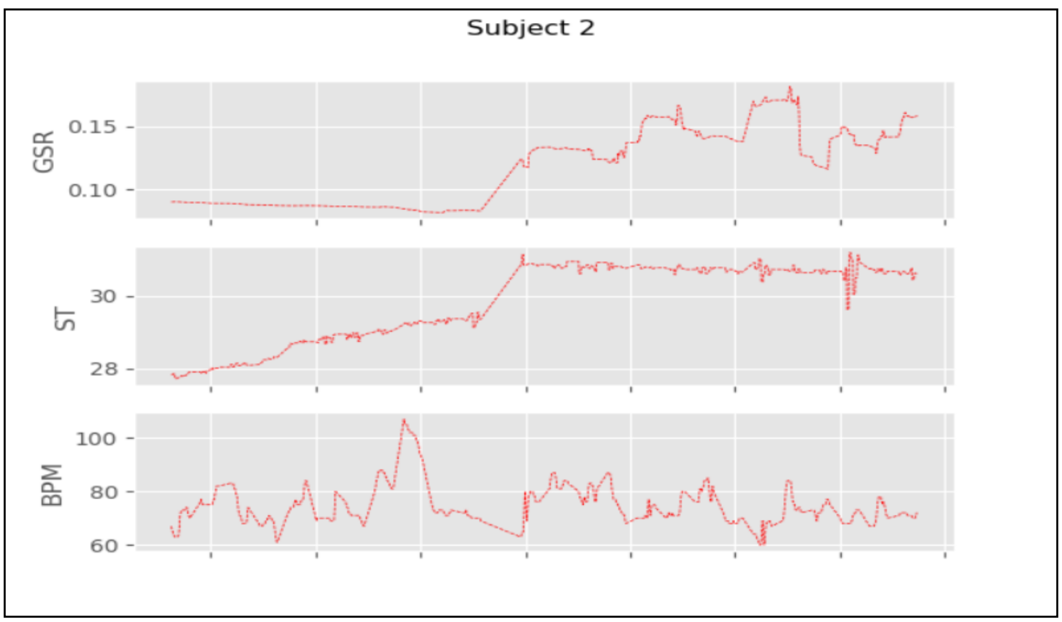
Category Type	Subjects	Category	Ratings					Means	STDEV
Favourite	Subject 1	Έντεχνο	5	2	4	4	5	4	0.547722558
	Subject 2	Rock	5	4	4	5	5	4.6	1.224744871
	Subject 3	Λαϊκά	4	5	4.5	3	4	4.1	0.741619849
	Subject 4	Dance	2	2	4	4	5	3.4	1.341640786
	Subject 5	Rock	4	4	2.5	3	4	3.5	0.707106781
Worst	Subject 1	Pop	2	1	1.5	3.5	2.5	2.1	0.961769203
	Subject 2	Λαϊκά	4	3	0	4	0	2.2	2.049390153
	Subject 3	Έντεχνο	2.5	1	0	1	2	1.3	0.974679434
	Subject 4	Έντεχνο	0	1	3	0	0	0.8	1.303840481
	Subject 5	Hard Metal	2	2.5	1	2.5	1	1.8	0.758287544

APPENDIX 4: Subjects' GSR, ST and BPM Time Series

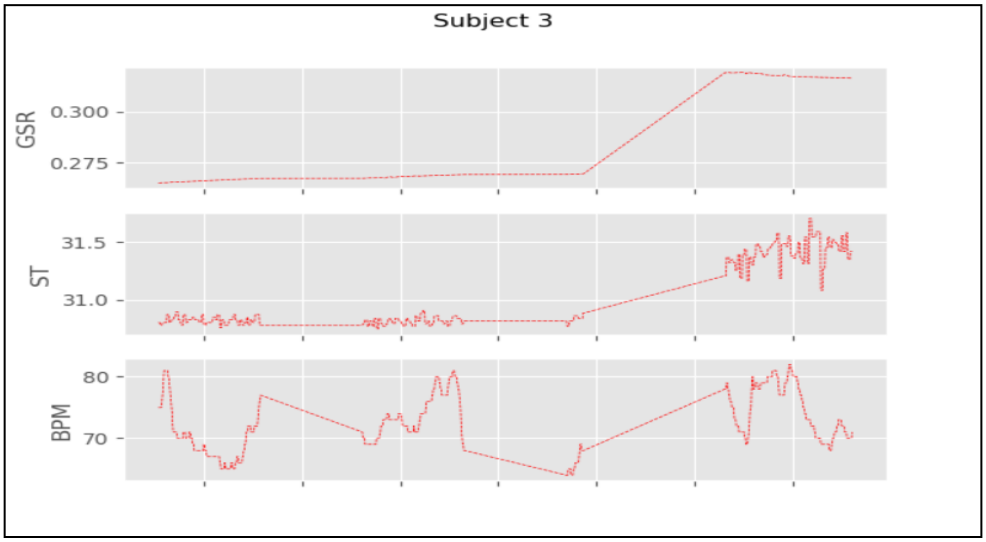
Subject 1



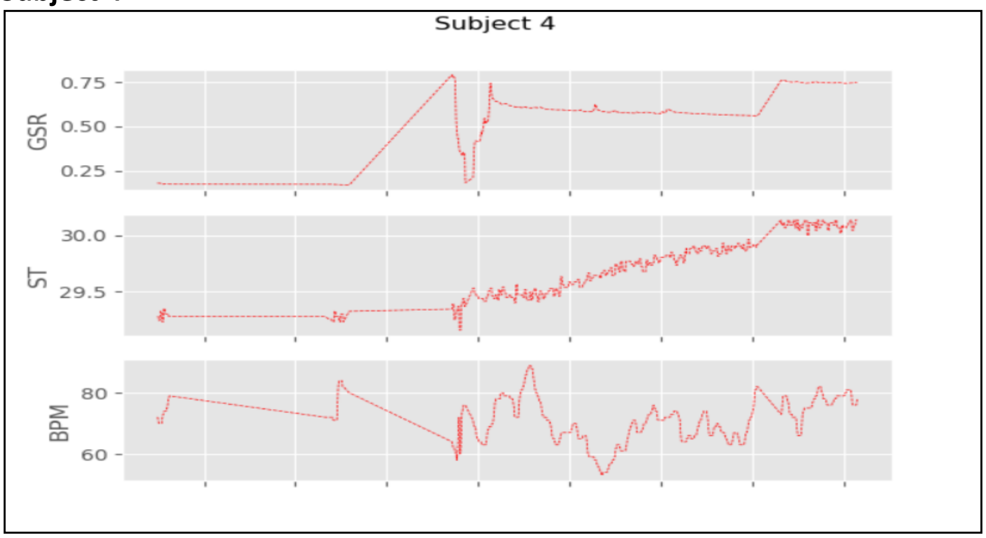
Subject 2



Subject 3



Subject 4



Subject 5