A Context-Aware Recommender System

By Means of Facial Expressions and Pairwise Comparisons

POLITECNICO
MILANO 1863

Nima Maleki

School of Industrial and Information Engineering

Politecnico di Milano

A thesis submitted for the degree of

Master of Science

Milan/Bolzano 2018

Dedicated to the most beautiful souls I ever got to know;
my wife,
and my mother.

Acknowledgements

I wholeheartedly thank my advisors Professor Marko Tkalčič who played a great role in nurturing me over the course of this thesis work, Professor Mehdi Elahi who always supported me mentally and academically and was the one who introduced me to the field of recommender systems, and of course Professor Francesco Ricci who has been the best mentor that we all could ever wish for.

I would also like to thank Professor Paolo Cremonesi, who has been a great teacher of mine, and I wouldn't be attracted to this field if it wasn't for his presence.

Finally, I thank my lovely wife and my dear family who never stopped supporting me.

Abstract

Recommender Systems (RSs) have been gaining attention constantly over the course of the last ten years, in part because of their great potential in e-commerce and multimedia purposes. RSs are a subclass of information retrieval systems that try to estimate the opinion of individuals about the products of a catalog. RSs are dependent on user preferences, which are difficult to acquire, in part due to users' natural tendency to not disclose information. One novel approach for obtaining these preferences is to infer them from users' behavior while interacting with the system. In the context of a music RS, we focus on inferring user preferences on different songs by means of applying supervised learning techniques on their facial expressions and emotions. Successively we build a RS using the inferred user preferences.

While we obtain notable accuracy and precision in the user preference predictions, we also show that such predictions can greatly reduce the RS's dependency on hard-to-obtain explicit user preferences. Moreover, we demonstrate that using the inferred user preferences can be an effective approach for tackling or avoiding the Cold Start problem.

Sommario

I Recommender System (RSs) hanno ampiamente attirato l'attenzione nel corso degli ultimi dieci anni, anche grazie al loro grande potenziale in ambito e-commerce e multimedia. RSs sono un sottoinsieme degli information retrieval systems che cercano di stimare l'opinione degli individui a proposito dei prodotti di un catalogo. RSs sono basati sulle opinioni dei clienti che sono difficili da ottenere, a causa della loro naturale tendenza a non diffondere informazioni. Un nuovo metodo di ottenere le opinioni è carpirle dai loro comportamenti mentre interagiscono con il sistema. nel caso di un RS per la musica, cerchiamo di predire le preferenze sulle diverse canzoni attraverso techniche di supervised learning applicate alle loro espressioni facciali e emozioni. Successivamente, costruiamo il RS usando le preferenze predette.

Abbiamo ottenuto buone precision e accuracy nella predizione delle preferenze, e abbiamo anche mostrato che queste predizioni possono ridurre molto la dipendenza del RS dalle preferenze esplicitamente espresse dall'utente. Inoltre abbiamo dimostrato che è possibile utilizzare le preferenze predette per arginare il Cold Start problem.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The users of the Internet consume a considerable amount of information daily and this amount has been only increasing. As the global network of information gets larger by the second, we are overwhelmed by the variety, velocity, and diversity of the content that is presented to us. Therefore, deciding on a movie to watch, play list to listen to, or hotel or diner to book becomes more difficult. The relationship between humans and the web is formed around searching, where the user looks up the content and picks the one that suits them. However, choosing content to consume on the web can also be done in the opposite direction, where the service provider itself understands the user's needs and provides the right type of content. This is when the field of Recommender Systems forms.

RSs in a general sense are software tools, techniques, or services that provide suggestions for items to be of use to a user [1]. The *item* can refer to a musical piece to listen to, a product to buy on line, an article to read, etc. Recommender Systems are deemed as one of the interesting sub-classes of Information Retrieval Systems that have brought a better experience while working with computers and have also generated massive business value throughout the years. RSs are designed to enable or facilitate the recommendation of *items* to users in a way that the user might probably be interested in the item. This is done by means of inferring the users' preferences which can be expressed in different ways (e.g. ratings) on different items of a catalog. Generally, the recommendations are obtained by comparing the users profile (which

can be modeled in various ways and using several techniques) with the descriptions of items (content-based approach) or with the profiles of other similar users (collaborative-filtering approach). In the following paragraphs, we try to introduce the concepts necessary for understanding the research questions and goals of this research.
Oftentimes, users' preferences on different items are measured and expressed via single ratings in a specific predefined range, e.g. 1 to 5 stars on Amazon.com or 1 to 10 points.

However, there is at least one downside to the conventional single ratings method: A user might like two items equally, but when it comes to choosing one over the other, they are not able to express their preference using the conventional ratings method. Since in RS we need to rank each user's favorite items, we need to differentiate between these items. At times, this cannot be achieved using the conventional ratings. For instance, a user who prefers one item over another might settle for rating them both equally, due to the limitations of the rating scale. Another example is a user who has previously rated an item with the highest rating possible, but later on finds another item that he deems even more desirable. In this scenario she has no way of expressing her preference for the second item over the first.

Despite its popularity and intuitiveness, collecting single ratings is not the only effective method for understanding users' preference. One alternative method is using pairwise scores, where instead of one item i, a pair of items (i, j) is presented to the user and the user chooses which item of the two and to what extent suits their desire. In pairwise scores or similar techniques, the user essentially picks a negative number to express preference towards item i and a positive number to do the same for item j or vice versa. This can be accomplished using explicit numbers or other graphical user interfaces (UI) elements such as radio buttons and sliders.
There have been efforts to design an RS based on pairwise preferences [2], although choosing pairwise preferences over conventional ratings brings its own challenge. The number of available user preferences drops, since each user gives one rating per two items and not one rating per item in a catalog. This means that the sparsity of datasets

built on pairwise preferences is normally even higher than that of more conventional RSs with absolute ratings. The increased sparsity of the matrix of preferences makes predicting the missing preferences more difficult, since we have less information per item.

Another important facet of RSs is the factor of context. The context is the implicit situational information that describes the environment in which the user is interacting with the system [3]. Recommending items without taking into account the users' situation can be ineffective, since the needs of a user change drastically as the context changes. Imagine a user that likes Hard Rock music. If he is training at the gym, there is a good chance that he will appreciate listening to a rock song. However, this might not be the case while he is studying or replying to a sensitive e-mail. These kinds of problems are addressed by $context-aware$ RSs, which calculate the user utility for an item based on a set of additional variables- context variables.

In the same analogy, the song that the user might like when he has just began to study can differ from the song that he likes in the forth consecutive hour of studying. Here, not only the activity is important, but also the user's mood or fatigue plays a role. Another important fact to notice is that Asking users to provide explicit ratings is *intrusive* and disrupts the user interaction. That is why RSs use other information available from the users alongside the explicit ratings to enrich their knowledge about users and increase their prediction power. In fact, a significant amount of what organizations know about their online users is observed implicitly rather than asked explicitly. Examples of implicit signals are clicks on a product, the amount of time a user spends in a page, and how long does he or she scroll in the page, which can all eventually contribute to the quality of RS.

Implicit information have an important advantage over explicitly collected preferences: While being easy to collect, they are not intrusive. One can observe users' behavior during their interaction with the system while the user is not bothered by direct questions.
However, the implicit signals generally represent the user's preferences

less accurately with respect to explicitly expressed preferences. Another point is that, in contexts and scenarios that are well defined, pairwise preferences have several advantages over the conventional absolute ratings: they help the users to reflect more on their preferences and require less cognitive effort when the items that are compared are comparable. Two items are comparable when we can compare them on a feature-basis, which is especially true when the user task is clear. This is why in such a scenario, using pairwise preferences can be preferable over absolute ratings.

1.2 Problem Definition

We are interested in answering the following three **research questions**:

1. **Can we infer users' preferences on music from their facial expressions during the listening by means of a supervised machine learning technique?**
2. **Can a recommender system make high quality recommendations using *only* the user preferences inferred by the above-mentioned model, compared to the same RS that uses explicit user preferences?**
3. **How much more implicitly acquired pairwise scores, as compared to explicitly acquired, are needed for achieving an equal quality of recommendations?**

To answer them, in this thesis we build a context-aware recommender system that is based on pairwise preferences and emotional responses.

1.3 Research Methodology

The research methodology followed in this thesis consists of 5 parts: (1) reviewing the research literature, (2) choosing a relevant context to act in, (3) designing, implementing, and running an online experiment to gather data, (4) modeling and evaluation of the preference inference, (5) evaluating the recommender system

First, we conducted a literature review to identify the main techniques and characteristics of recommender systems. We tried to focus on the literature that is specifically related to the work described in this dissertation, such as preference elicitation techniques, context-aware recommendations, and emotional studies and music.

Second, we conducted pre-studies alongside findings in the literature, to find a reasonably widespread application for the proposed recommender system.

Third, we designed and implemented an online web interface capable of gathering data from participants in specific ways in line with the second step. We used this interface to collect data about users' musical preferences, emotions, facial expressions, musical abilities, and personality traits.

Then, we used statistical and machine learning approaches to infer users' musical preferences using the data gathered in the third part. The evaluation showed a high accuracy.

As a final step we also built a recommender systems using explicit scores and scores inferred from facial expressions and compared the results.

1.4 Contributions

In this research we used an unobtrusive approach for gathering implicit signals in the form of facial expressions using an emotion detection algorithm via a web interface. We presented ways of inferring a user's explicit preference using his or her implicit facial signals. We have shown that these results are significant.

We collected a dataset of such data, which can be useful for future works. We have shown that there are differences in the accuracy of the score inference based on personality. Using personality as a feature can hence improve the inference accuracy.

We have demonstrated the usefulness of the above-mentioned predictions in reducing the need for asking a user for explicit feedback, and also in tackling the cold start problem.

1.5 Thesis Outline

The rest of this thesis is organized as follows:

- Chapter 2 provides an overview of the related research literature and the state of the art. We explore research efforts in the fields of context-aware recommendation, preference elicitation, and emotions and their relationship with preferences.

- In Chapter 3 we layout the background of our research work and justify our design choices. We explain our pre-studies and instruments in detail.

- Chapter 4 contains the details of the descriptive analysis, dataset characteristics, predictive modeling and recommender system specifications. We also inspect and compare the results.

- Finally in Chapter 5 we sum up the results and their implications, and discuss future possible research directions.

Chapter 2

State of The Art

In this chapter we review the relevant literature and the recent advances in the subfields that are connected to our research work.

2.1 Context-Aware Recommender Systems

As briefly pointed out in the introduction of this thesis, an important facet of recommender systems is the context. By context, we generally mean any of the possible circumstances in which the RS is to be utilized. An RS that takes at least one of these circumstances into consideration when making recommendations, can be called a context-aware recommendation system [4]. Formally, context has been defined in a number of ways by various researchers, like [5] and [6], while the most frequently cited definition is the one proposed by Dey and Abowd [3]. Yong Zheng has created a framework for detecting the context in recommendation [7]. The framework enables us to define the context based on the user, the item, and the action itself (i.e., listening or watching).

Overall, there can be a trade-off between an RS's generalizability and its context-awareness; the more we try to pinpoint a specific use case and scenario for the RS, the more niche it becomes. Then, such niche RSs should later be integrated with other systems that take care of the rest of the context space. Furthermore, an RS that acts in a broad range of contexts, may face the data sparsity issue. In such a scenario, the user preferences are expressed under different circumstances for different items, hence the RS will be built using fewer user preferences per context and per user. The data sparsity problem is also a reason why we chose to work only on one context.

2.2 Preference Elicitation Techniques

Preferences can be acquired as single judgments about an item (e.g. a rating on a scale from 1 to 5) or as pairwise judgments when comparing two or more alternatives. While the former approach is widely used in recommender systems, the latter has received little attention. Few examples are [8] and [9]. However, research in behavioral economics showed that people do not hold well-defined opinions about their value, choice or attitude judgments [10]. Boeckenholt [10] further argued that providing pairwise preferences is preferred over single ratings when there are reasons to believe that the users have difficulties assessing their preferences about an item. In the music domain this is particularly true as one would have a hard time giving a single judgment about a particular song. However, when asked to compare two songs for a given usage scenario, the user would be able to provide a more reliable judgment about her preferences.

As asking users to explicitly express their music preferences is intrusive, researchers have started to focus on the implicit acquisition thereof. Parra and Amaitrian [11] have shown that using existing traces of human behavior in the domain of music listening makes good predictions of the actual music ratings. Popular implicit behavioral signals used to infer preferences are play-counts [11] and listening time [12].

In order to explore other implicit signals that could carry information about the music preferences we turned to emotions. There is a lot of research showing that emotions and music preferences are correlated [13, 14, 15]. In fact, music listening and experiencing emotions are tightly coupled. There is a long tradition of research that shows that people feel and express emotions when they listen to music [16]. Researchers in music psychology distinguish between expressed, perceived and induced emotions [16]. Furthermore, people also use music for regulating their emotions [14].

2.3 Emotions and Preferences

Measuring emotions through questionnaires is an intrusive task. The field of affective computing addresses the problem of detecting emotions unobtrusively from various modalities, such as video cameras, voice or physiological sensors [17]. Valenti et al. have sought to perform the tasks of face detection, facial feature tracking, and emotions classification by facial recognition and using an integrated system [18]. Black and Yacoob have utilized local parametrized models of image motion for detecting motions of human face, and consecutively the human facial expressions [19]. Bourel et al. have proposed an approach for the robust recognition and extraction of facial expressions from video sequences, which can tolerate higher levels of noise and face occlusion with respect to methods predating it [20]. Bartlett et al. have taken advantage of Support Vector Machines to perform the task of detecting spontaneous human facial actions in real-time [21]. Baltrušaitis et al. have taken into account also the upper body gestures in real-time facial expression detection [22]. These studies and others have laid the foundations for some off-the-shelf emotion detection solutions that are capable of inferring the emotional state of a subject in video streams. One such solution is Affectiva, which we utilized in this research.

Research on emotion detection from video recordings of facial expressions of emotions has shown that people are diverse in terms of facial expressivity. Cohn et al. [23] observed that individuals are diverse in the strength of their facial expressions and that these differences are stable over time. Tkalcic et al. [24] achieved an improvement in emotion prediction accuracy from facial expressions by clustering users in three different groups according to their expressivity and training separate models for each cluster.

Facial expressions were also used to infer stable user traits, such as personality [25], showing that people with diverse personalities have different facial expressions on the same stimuli. Another study showed that there is correlation between user characteristics, such as personality and music education, and the emotional perception of music [15].

2.4 Collaborative Filtering

Generally, recommender systems can be divided into two categories: content-based filtering and collaborative filtering. Content-based recommendation systems, as shown by De Gemmis et al. [26] and others, leverage on a user's past likings when making new recommendations. Collaborative Filtering (CF) methods provide recommendations of items based on patterns of ratings or usage without need for external information about either items or users [27].

The Collaborative Filtering approach requires a platform to collect feedback, both implicit and explicit, from users. With these information the Recommender System is then able to suggest items to users in a personalized fashion. The classical approach in Collaborative Filtering grounds on the concept of neighborhood, in which a user will like either items that are similar to items that has been previously liked, or items that has been liked by similar users. In order to find items that could be relevant for the user, the system relies on the computation of similarity metrics between pairs of users that exploits the information coming from the set of items liked by the user in combination with the sets of liked items of the other users in the system. In such a way the system suggests items to the user that are unknown for him, but are rated high by other similar users in the system. This type of Collaborative Filtering is user-based.

The user-based Collaborative Filtering idea was further developed into the item- based Collaborative Filtering [28], in which items that are similar to those liked by the user are suggested as recommendations. Both the user-based and item-based Collaborative Filtering techniques are affected by the cold-start problem. This problem emerges when providing recommendations for new users or new items. For new users, the system does not have information about their preferences in order to make recommendations, while also new items might have less known characteristics. [29, 30].

In order to avoid this problem, Recommender Systems implement a preference elicitation task in their bootstrap phase, in which the user has to provide some feedback to a subset of items in the system. By exploiting the acquired data from the user, the system is then able to train a prediction model in order to provide a set of items reflecting the user preferences. Together with the classical user-based and item-based neighborhood approaches, latent factor techniques, such as Matrix Factorization (MF), have emerged in the panorama of the recommendation techniques. Matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. When the correlation of user and item factors reaches high levels, a recommendation is made [31].

Chapter 3

Background, Pre-Studies and Setup

3.1 Research Questions And Hypotheses

In this chapter we explain the steps taken towards a context-aware recommender system that utilizes pairwise preferences and takes into account (1) the activity that the user is performing at the moment of recommendation, and (2) the user's current mood based on his or her facial expressions as the context. The evaluation results of the aforementioned RS helped us in answering the following three **research questions**:

1. **Can we predict users' preferences about items only by analyzing their facial expressions and by means of a supervised machine learning technique?**
2. **Can a recommender system make high quality recommendations using *only* the user preferences predicted from facial expressions?**
3. **How much more implicitly acquired pairwise scores, as compared to explicitly acquired, are needed for achieving an equal quality of recommendations?**

We have made a number of hypotheses in this research, which will be listed below, and will be further discussed and justified throughout this chapter. We conjecture that since musical pieces trigger emotions in subjects and those emotions are manifested with body responses (one of which is facial expressions), there should be a correlation between a subject's body responses and his or her preference for the musical piece. We hypothesize that different groups of people with different personalities should not be equally emotionally expressive and thus, should be easier or more difficult to predict. Furthermore, as suggested by Rentfrow and Gosling [32], there is a link between personality traits and musical taste.

We further speculate that building a recommender system based on inferred user preferences via body responses should be feasible, and although it will not match the quality of a RS based on real preferences, there can be a compromise between asking users for explicit feedback and inferring them via facial responses.

Prior to answering the research questions earlier defined, we need to consider that building a recommender system that recommends various types of items and takes into account various contexts comes with complexities such as the sparsity problem, which we have pointed out earlier.

Hence, we decided to initially narrow down our focus to a setup of one type of recommender systems, with one context, and one item category. This would give us a platform for expanding the setup to other contexts and item types in future works.

Before proceeding, the following needed to be decided:

1. The preference elicitation approach.
2. The specific application and use case of the recommender system and the specific item type (music, film, books, etc.) to be recommended.
3. The context in which the final recommender system is going to be used.

In order to decide on the points mentioned above, we reviewed the relevant literature for hints and also conducted two pre-studies. The final decisions and the reasonings driving them are described in the three upcoming sections of this chapter.

3.2 Preference Elicitation Approach

Every recommender system is dependent on the process of preference elicitation. Traditionally, preference elicitation is done either using (i) explicitly entering ratings of some numerical form (e.g., one to five stars) or (ii) using implicit signals, such as clicks, previews, purchases etc. Explicit preferences contain information about the true opinion of users, especially if they have been expressed after the users have been engaged with the item. However, explicit ratings are difficult to acquire. Users are often not willing to provide feedback, and asking them to do so explicitly is intrusive and expensive. On the other hand, implicit ratings are easier to acquire and are natural byproducts of a user's interactions with the system. However, they are in general less accurate in determining the user's opinion.

Independently on whether feedback is acquired implicitly or explicitly, it can be acquired in different forms. The most widespread form of feedbacks are absolute ratings, such as a rating on a scale or thumbs up and thumbs down. There is, however, another form of user preference; pairwise comparisons [9]. In pairwise comparisons the users are exposed to a pair of items and they have to indicate which one they prefer over the other. This preference indication, usually on a numerical scale, is called the pairwise score.

Pairwise preferences have several advantages, one of which is enabling the user to reflect more and easier on their preferences. Considering two items that are of the same nature and can be described by the same features, and are hence "comparable", the user requires less cognitive effort to arrive to a conclusion on his or her preference. In situations where the user's task is clear and well defined, often the items are comparable as well. This gives us a reason to choose pairwise scores as our preference elicitation method, since we do want to focus on one specific type of item and application.

3.3 Recommender System Use Case

We are particularly interested in studying the correlation between the users' emotions and preferences. Ideally, we would infer the users' explicit preferences on items from the implicit feedback that we receive from them, in particular in the form of the emotions induced in the user. Hence, we would have to choose an item type to be recommended that has correlation with the subject's emotions.

Music is a particular application where emotions are important. We hypothesize that emotions induced in the user by listening to music are indicative of his opinion on the musical piece and thus can be used to infer explicit pairwise preferences. Research has shown that music listening evokes emotional responses of different forms [14]. These emotional responses are coupled with bodily responses, among which, facial expressions are one of the strongest predictors of the current emotion [33]. Based on this assumption, two pieces of music that evoke different emotions should evoke different facial expressions, too. Hence, we conjecture that the difference in facial expressions of a user while listening to two songs is related to the pairwise preference of that user over the two songs.

Moreover, music is an application where items can be compared on a feature-basis. These make music a good application for the pursued recommender system and a good duo with pairwise comparisons. In our setup, we allow the user to listen to two musical pieces one after the other, which further helps the users to compare the items.

3.4 Music Database

Having decided on a music recommender system, we needed to have a database of musical pieces in order to conduct experiments. One such dataset is Moodo [34], which has been created considering the relation between music listening and emotions. The database contains a total of 200 songs. 80 songs are from the royalty free online music service *Jamendo*, representing a diverse variety of standard genres. 80 songs were included from a dataset of film music excerpts, 20 from a database of folk music and 20 from a contemporary electro-acoustic music collection.

Apart from its diversity of genres, the Moodo dataset has a few desirable characteristics.

Firstly, the songs contained in the dataset are mostly obscure and not widely known to the general public. The obscurity of the songs is important, because it minimizes the preference bias caused by familiarity of items. The popularity might cause the judgments to be biased. This would affect the distribution of ratings per song and would limit our exploration ability.

Secondly, this dataset contains 200 song snippets, each lasting approximately 15 seconds. The relatively short length of the snippets make the dataset a good fit for pairwise comparisons, since the user can navigate the songs back and forth rapidly and draw a conclusion without much cognitive effort. This would allow the users to compare more songs in the same amount of time.

3.5 The Context

As we have pointed out earlier, the preferred song for a user can change completely in different circumstances, i.e. a song suitable for a physically intense activity such as jogging might not be favorable while performing a mentally demanding task such as reading a book. In line with our strategy of pursuing a recommender system in one specific setup, we decide to find one type of "activity" and one specific "time period" for the recommender system to recommend for.

In order not to have sparse data for different contexts we opted for the use case of one single activity/situation where music is consumed. We aimed at an activity that people do often and also listen to music when they do it. In order to identify such an activity we ran a pre-study.

3.5.1 Activities and Music Pre-study

This pre-study involved gathering data from online participants and analyzing them.

3.5.2 Data Acquisition

In addition to personal information such as age, gender, and education, the participants were asked

1. *which periods of day do they normally listen to music in* and in those periods,

2. *which activities do they take part in that involves music listening* and

3. *what type of music do they listen to* during those activities.

The collected data includes various features that are summarized in Table 3.1. The first part questions the periods of the day in which the participant listens to music on a regular basis, and the second part asked for one to three activities that the participant performs in each period and the type of music that he or she normally listens to during the activity. Each participant could choose between one to seven periods of the days, and for each period, he or she had to name one to three activity-music pairs.

We decided to let the participant deliver his or her answers to activity and music questions as simple text, rather than using input methods with predefined entries such as drop-down menus or radio buttons, in order to not introduce a bias. In other words, introducing predefined categories might affect the participants opinion, which is something that we wanted to avoid.

We performed post-coding [1] on the open answers that the participants gave to activity-music pairs.

[1] Coding is an analytical process in which data, in both quantitative form (such as questionnaires results) or qualitative (such as interview transcripts) is categorized to facilitate analysis. post-coding is a type of coding which involves mapping of open questions on completed questionnaires.

Table 3.1: Structure of the dataset from the pre-study on activities and music types

Personal Information	Periods of the day in which the participant listens to music	Activity which involves listening to music and type of music
ID	morning - e.g. waking up, preparing breakfast	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
Age	morning commute to work or school	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
Gender	daytime - e.g. studying, working	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
Education	daytime - e.g. studying, working	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
	other commuting	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
	afternoon - e.g. hanging out with friends or family, home chores	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3
	evening - e.g. reading a book, preparing to go to bed, before falling asleep	Activity 1, Music 1 Activity 2, Music 2 Activity 3, Music 3

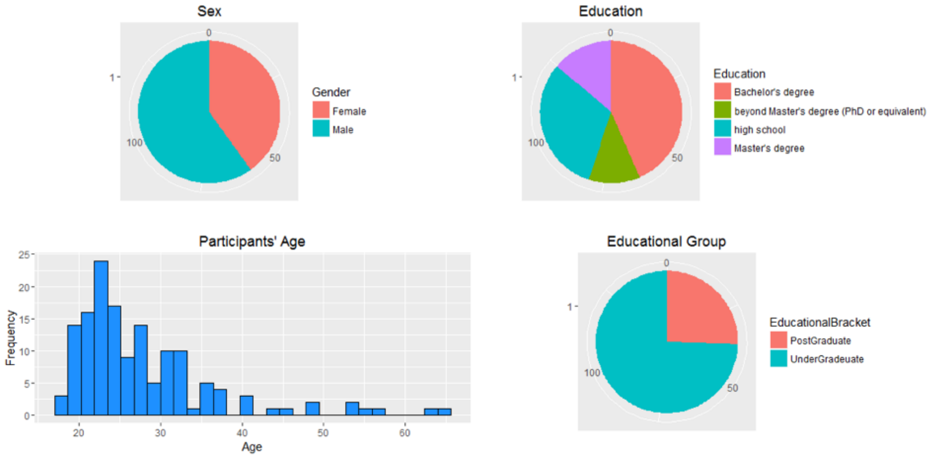


Figure 3.1: Personal information of participants and their distributions

3.5.3 Results

Overall, 145 people answered the questionnaire, out of which 85 were males and 60 females. In the sample 75 per cent were undergraduate students and 25 per cent postgraduates. The demographical outcomes of the pre-study data acquisition are summarized in Figure 3.1.

Based on the acquired data, we can conclude that more than 75 per cent of our participants listen to music while doing their daily tasks such as **working and studying**. Moreover, periods of the day which involve **commuting** are also popular in general Figure 3.2. Interestingly, these are still true even if only people with a Masters degree or higher are considered, as seen in Figure 3.3. This indicates that we need to explore the third period and the periods that involve transportation more deeply.

We also investigated which activities are the most popular in each period, and more importantly in the more popular periods. In the period which involves performing daily tasks, working, studying, and coding (programming) are the most frequent activities, as summarized in Figure 3.4. Moreover, people generally use the same means of transportation in all three periods of the day which involve commuting. During those three periods, most people tend to drive and listen to music as it can be seen in Figure 3.4 and Figure 3.5. Considering this, before investigating the associated genres, we merged all of the three periods of the day, as shown in Figure 3.6. Driving, Commute, Walk-

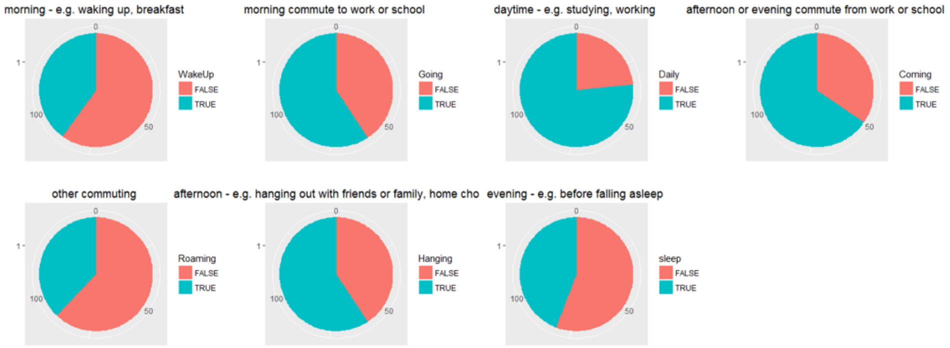


Figure 3.2: Periods of the day and the proportion of people who have reported that they listen to music in them. TRUE (Cyan) is the part of the study population that DOES listen to music in a period of the day.

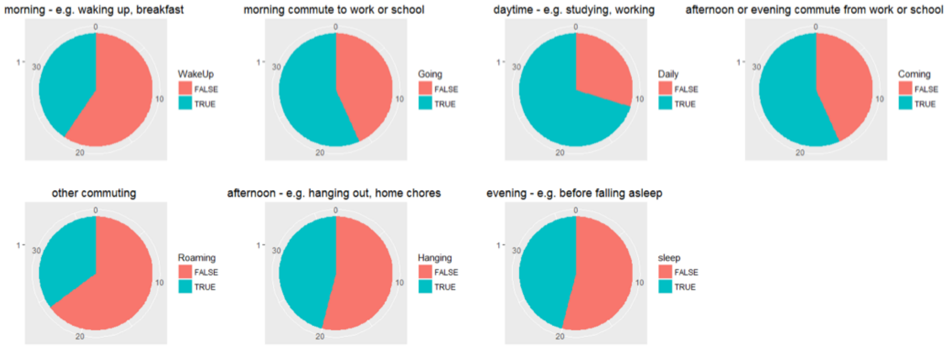


Figure 3.3: Holders of Master's degree or higher: Periods of the day and the proportion of people who have reported that they listen to music in them. TRUE (Cyan) is the part of the postgraduate population that DOES listen to music in a period of the day.

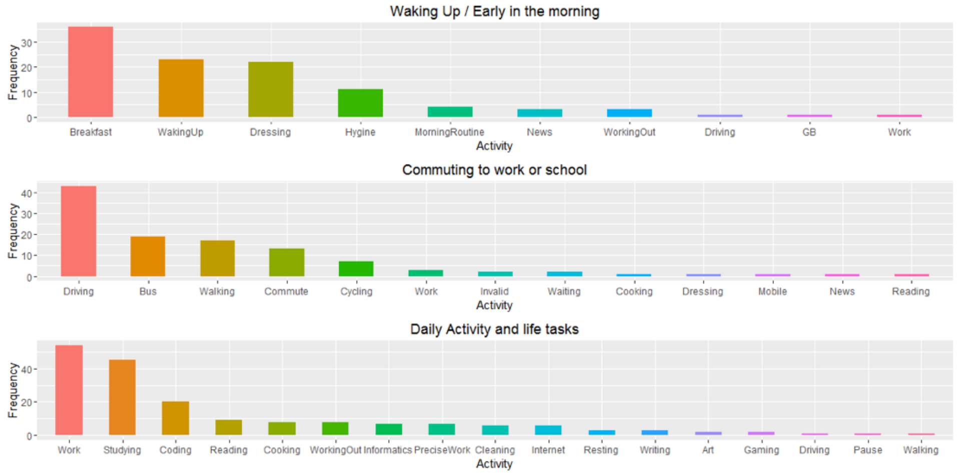


Figure 3.4: Activities and periods of the day which involve music listening (1)

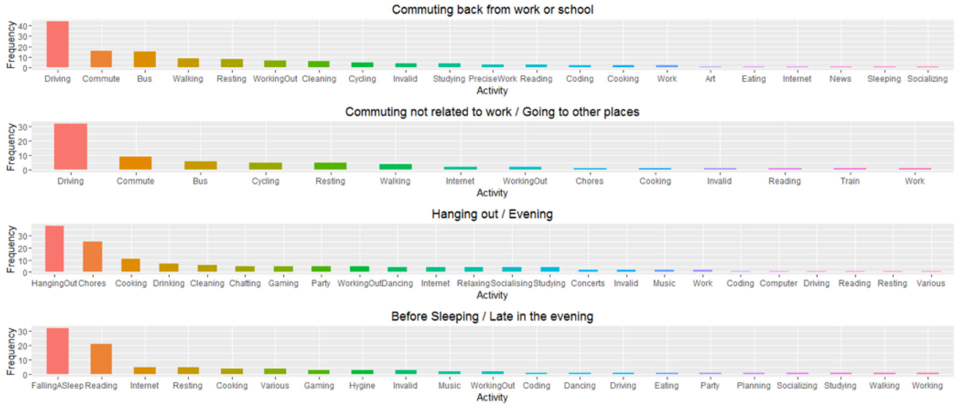


Figure 3.5: Activities and periods of the day which involve music listening (2)

ing, and Bus are the most frequent activities if we consider all the three periods.

Understanding the frequent genres associated with the aforementioned activities is a side goal of this pre-study. In Figure 3.7 we can see the general popular genres of music, where Rock and Pop are clearly more popular than most of the other genres.
However, during daily tasks the same does not hold. The most popular genres while working are Rock, Pop, Chill, Radio, and Anything (i.e. when the participant has declared no specific genre):Figure 3.8. Whereas people tend to listen to Chill, Classical, and then Rock while studying, which can hint that generally more mentally demanding ac-

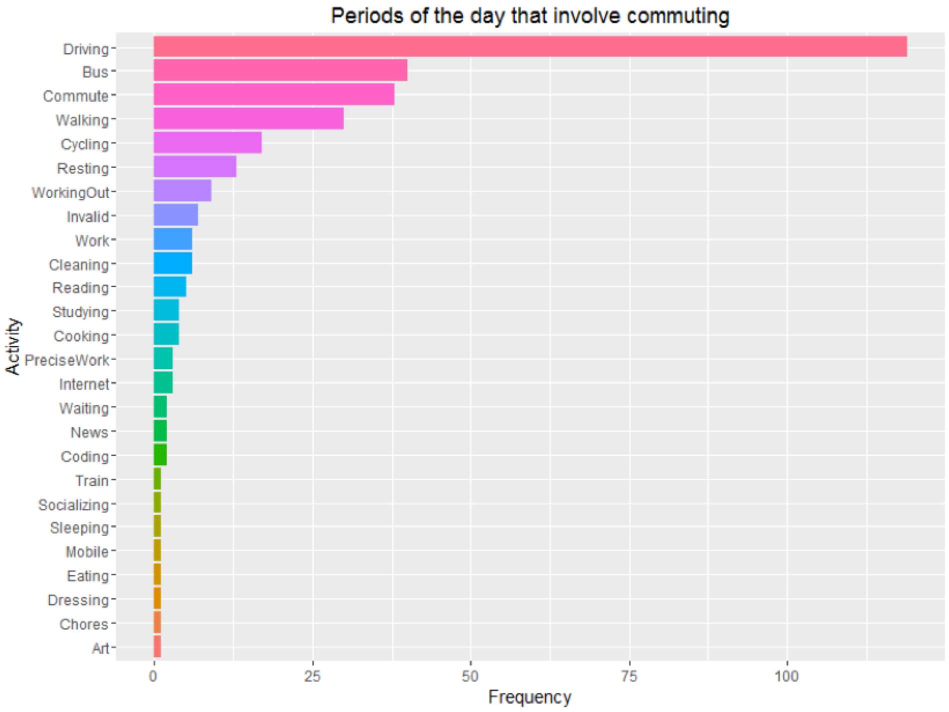


Figure 3.6: Periods of the day that involve commuting for the participants

tivities are associated with less mentally demanding genres, as seen in Figure 3.9. Coding (programming) is also a frequent activity during which people tend to listen to less mentally demanding musical styles such as Chill, Electronic, Dance, Rock, and Trance, as shown in Figure 3.10.

Considering these observations, we chose cognitively demanding tasks for the user activity of the recommender system, because users do it frequently and they also often listen to music during this type activity.

3.6 General Music Popularity

Another question that needed to be answered before attempting to acquire pairwise comparisons from users is *how frequently should each song snippet from the Moodo dataset be presented to users?*.

Having a roughly uniform distribution of pairwise scores among songs would yield few overlapping songs. Overlapping songs are needed

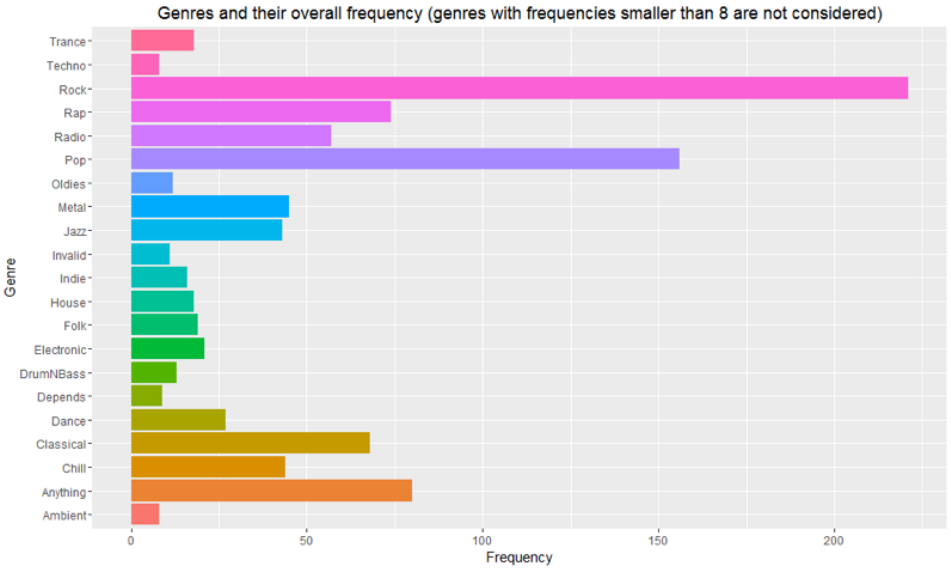


Figure 3.7: Overall popularity of genres in the pre-study

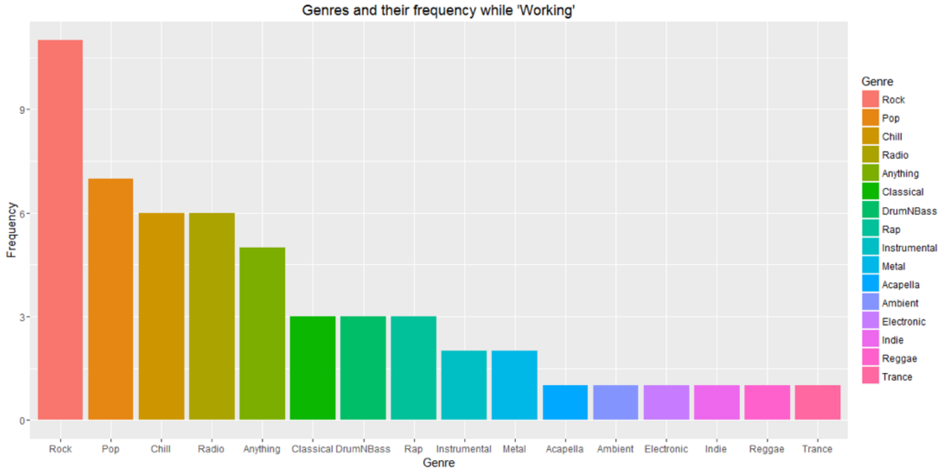


Figure 3.8: Most frequently listened to genres while working

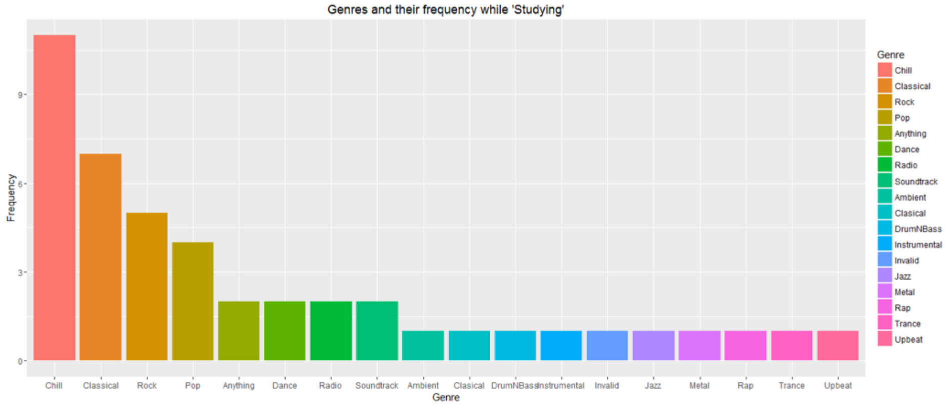


Figure 3.9: Most frequently listened to genres while studying

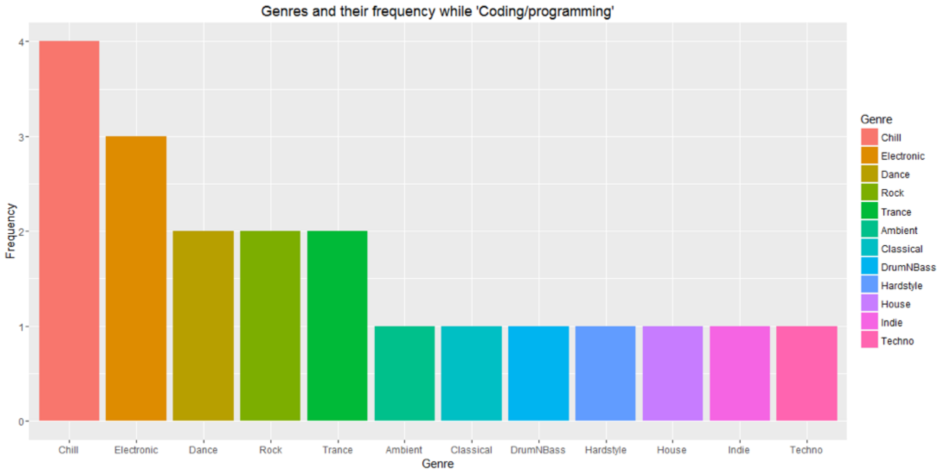


Figure 3.10: Most frequently listened to genres while programming

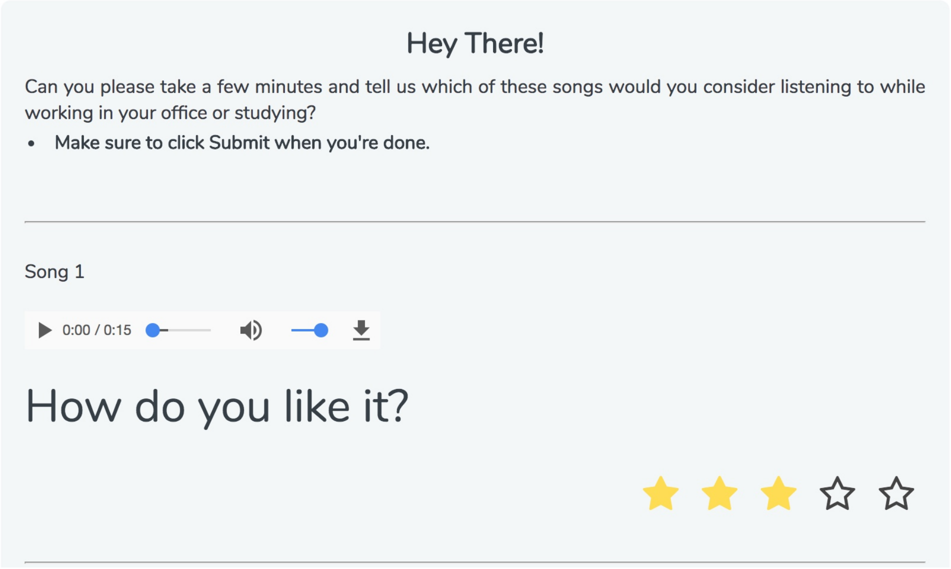
because the recommender system uses the MF algorithm, which works better if there are many overlapping items. Hence, we decided to make some songs occur more often.

In uniform distribution scenario, we would neglect the fact that in a music service, some songs are more popular than others and are thus more often listened to and *rated*. We decided to force a short-head, long-tail scenario by making some songs appear more often.

3.6.1 Experiment

In order to find the most popular songs of the dataset and forcing a short-head, long-tail scenario, an online pre-study was performed (Figure 3.11), in which we asked the participants to listen to n of the 200 song snippets in the Moodo dataset and provide a rating from 1 to 5 stars to each song. The participants were asked to take into account the context (cognitively demanding activities) in which the music excerpt is listened to.

The popularity curve of the songs was not steep enough to result in the concentration that we are looking for. In order to overcome this problem, we needed to generate a distribution function that falls quickly near the y axis and is almost flat in the tail. A good option is the exponential probability distribution function, which has the exact same behavior, specifically with higher λ (Figure 3.12). Hence, we place the



Hey There!

Can you please take a few minutes and tell us which of these songs would you consider listening to while working in your office or studying?

- Make sure to click Submit when you're done.

Song 1

How do you like it?

Figure 3.11: A screenshot of the web interface developed for acquiring general ratings on Moodo songs

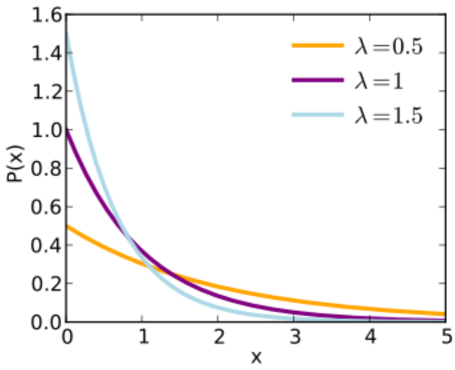


Figure 3.12: The exponential probability distribution function

songs on the x axis, and let $1/x$ define their probability of occurrence in the data acquisition phase where we ask for pairwise comparisons.

In order to rank the songs based on their popularity we used the delta correction:

$$\Delta = (\mu - \mu_g)/\varepsilon$$

$$\hat{r} = \mu - \Delta$$

Where μ_g is the *global average rating*, ε is the *number of ratings*. For each song, μ is the *average stars* it has received, while $\hat{r}$ is the *corrected* *rating.*

We rank the items based on $\hat{r}$. This will allow both the number of ratings, and quality of ratings to be considered. The resulting distribution is shown in Figure 3.14.
We chose this approach because ranking the songs based only on the average rating (Figure 3.13), positions songs with few, but high ratings to appear as popular, and ranking the songs based only on the sum of ratings of each song, positions songs with many low ratings high in the ranking.

3.7 Preference Acquisition Interface

In this section, we briefly describe the implementation of the web interface, which was used to acquire pairwise scores from participants, the data of which is the backbone of the rest of the research. We also comment on some design choices and specific characteristic.

Since we are particularly interested in relating the implicit facial reactions of the user to his or her explicit preference on different songs, the interface had to be capable of capturing user's facial expressions in real-time. Moreover, pairwise scores might be completely unknown to some people, at least as a way of expressing one's opinions in a web interface. Therefore, the interface had to be simple enough for the user to understand and express his or her preference.

A side objective of this study was to create a research tool for the community to be reused in similar contexts with minimal codebase changes. In this way, even if we could not answer the research questions adequately, other researchers would be able to use the interface to collect data in other setups and for different activities. The added value of extracting facial features, related to emotions, allows to pursue new research directions: (i) development of unobtrusive acquisition of user ratings from facial expressions and (ii) new multidimensional models of user preferences.

Although the interface is not maintained in production status, a video walk-through of the functionality of the Affectiva API and the

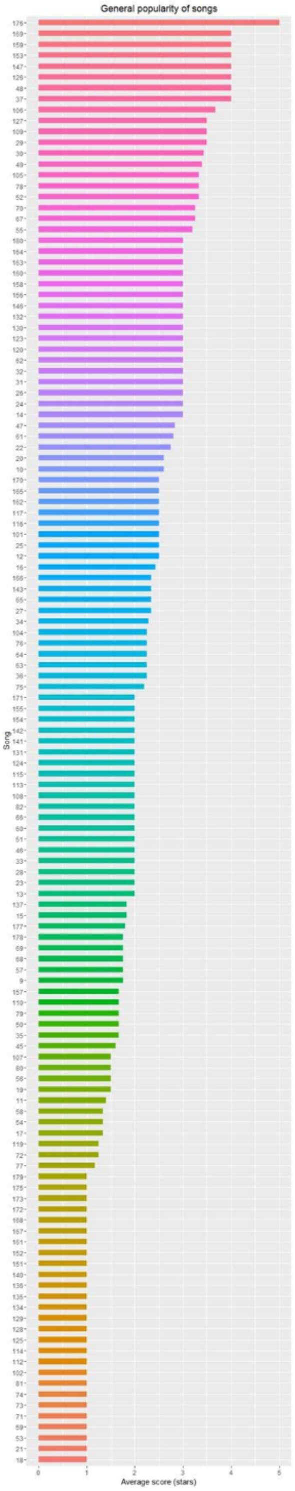


Figure 3.13: The general popularity of songs

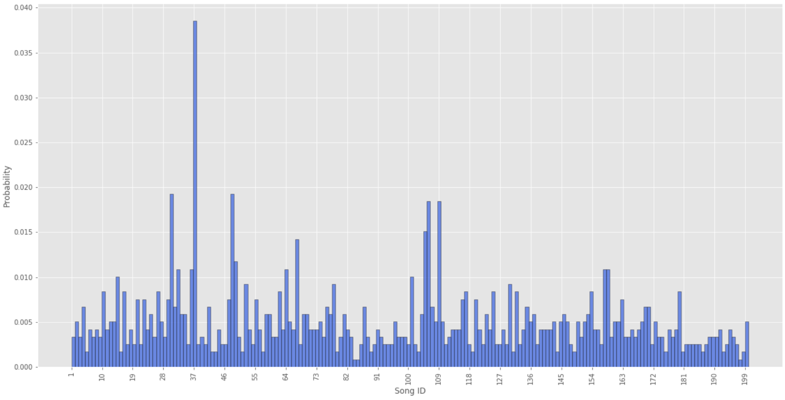


Figure 3.14: The distribution function of the songs' appearance in the pairwise comparison acquisition phase

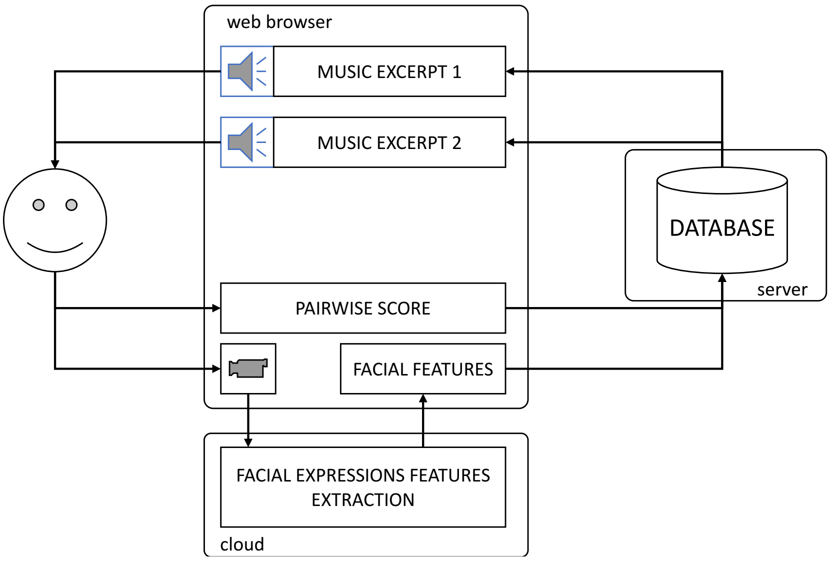


Figure 3.15: Web Interface flow: the user listens to two music snippets and then provides a pairwise score. During the listening to the music, a local instance of the API processes the user's face characteristics and movements on the client-side and sends the data to the server.

acquisition of pairwise comparisons are available at the URL `https://goo.gl/XxjBjL`. The codebase is written in the CodeIgniter framework which is built on top of PHP. On the client side, most of the code is in Javascript. An illustration of the information flow in the interface can be seen in Figure 3.15.

Here we will review the different sections of the interface:

3.7.1 Personal Data Collection

In the initial page, the subjects are instructed about the experiment and are informed about the type of information that will be collected from them. The subjects are also informed that they have the option to leave the experiment at any time, and they have the right to request the deletion of their data. The experiment begins only after the user has read the above and agreed and has given the consent by clicking. Then the user is directed to a page where he or she has to compile a form with their personal information, such as age, education level, and gender. Moreover, the user is asked to report his current valence-arousal [35]

The features resulting from this first part of the questionnaire are:

1. *Age*
2. *Gender*
3. *Educational Level*
4. *Business Sector or Industry*
5. *Valence-Arousal*

3.7.2 Music Listening, Scoring, and Facial Features Collection

The next page, is the most fundamental page of the interface. Users are asked to imagine a scenario where they are performing a *cognitively demanding* activity, hence the context of the future recommender system. Then, they are presented with a pair of song snippets from the Moodo dataset, and are asked to express their preference for one song over the other, in the given context. The song players are situated on the left and right hand sides, to induce a better feeling of distinction between them in the user, as seen in Figure 3.16. The pairwise preference is expressed using the handle and the bar in the middle, which is in fact a customized HTML5 `<input type="range">` tag, to blend in with the rest of the interface and give a natural feeling of selection between to songs. The handle can be dragged into 5 different positions, and as it does, the textual representation of the user's current pairwise score changes. Finally, the user submits his or her score. This is then repeated with 9 other distinct pairs of songs, resulting in a total of 10 pairwise comparisons per user. While the user is listening to the song snippets, his facial features are being analyzed.

The facial expressions are acquired through the webcam video stream. Each time a user starts comparing two songs, a local instance of the Affectiva API is instantiated on the client side, which processes the webcam stream completely locally, and sends the numerical features of the user's face in different moments of time over the HTTP to our server, in the form of JSON objects. The application server then appends the JSON objects to the database.

This design has two main implications: (i) Neither the database and application servers are burdened with processing the multiple video streams, and (ii) The user's privacy is better safeguarded, since his or her video stream is not shared with anyone and is not transmitted on the web.

The sampling frequency is not constant through time but in practice it fluctuates around 11 frames per second. In other words, for every second of music listening, we will have on average 11 snapshots of the user's facial features. In addition to facial features data, other features are returned by the Affectiva API.

Each music comparison, which is defined by the tuple $(user, song1, song2, score)$ contains several database entries. The pairwise score $score$ is on the scale $\{-2, -1, 0, 1, 2\}$, where $-2(2)$ means the user preferred the left-most (right-most) song and 0 means both songs were equally suitable or non suitable for the context. Each entry within the same comparison contains the facial features acquired at a sampling point while the user was comparing the two items.

Table 3.2: Summary of the features returned by the Affectiva API. The low-level facial features and the emotions are integers on the scale $[0, 100]$.

Emotions	**Facial Expressions**	**Appearance**
Anger	Attention	Age
Contempt	Brow Furrow	Ethnicity
Disgust	Brow Raise	Gender
Fear	Cheek Raise	Glasses
Joy	Chin Raise	
Sadness	Dimple	
Surprise	Eye Closure	
	Eye Widen	
	Inner Brow Raise	
	Jaw Drop	
	Lid Tighten	
	Lip Corner Depressor	
	Lip Press	
	Lip Pucker	
	Lip Stretch	
	Lip Suck	
	Mouth Open	
	Nose Wrinkle	
	Smile	
	Smirk	
	Upper Lip Raise	

From the Affectiva API we collect a total of 32 features for each video frame. These features are summarized in Table 3.2.

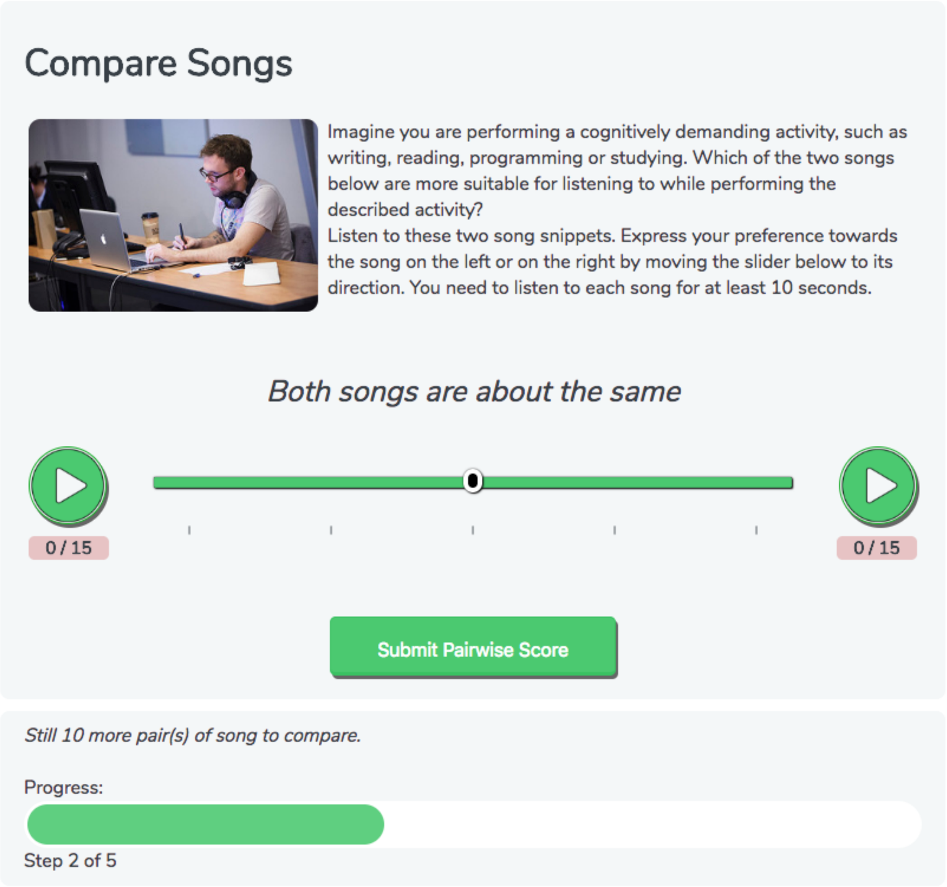
3.7.3 Personality Data

After submitting the pairwise scores, the user is redirected to a page where they have to answer 10 questions about their definitions of their own personalities from the TIPI questionnaire [36].The five personality factors for each user are calculated using the scoring scale available at

`https://gosling.psy.utexas.edu/scales-weve-developed/ten-item-personality-measure-tipi/`.

The list of the questions asked in this part of the interface are:

I express myself as...



Compare Songs

Imagine you are performing a cognitively demanding activity, such as writing, reading, programming or studying. Which of the two songs below are more suitable for listening to while performing the described activity?
Listen to these two song snippets. Express your preference towards the song on the left or on the right by moving the slider below to its direction. You need to listen to each song for at least 10 seconds.

Both songs are about the same

Still 10 more pair(s) of song to compare.

Progress:

Step 2 of 5

Figure 3.16: The section of the interface where the user is faced with a couple of songs each time, while their face is being analyzed

1. *Extroverted, enthusiastic.*
2. *Critical, quarrelsome*
3. *Dependable, self-disciplined*
4. *Anxious, easily upset.*
5. *Open to new experiences, complex.*
6. *Reserved, quiet.*
7. *Sympathetic, warm.*
8. *Disorganized, careless.*
9. *Calm, emotionally stable.*
10. *Conventional, uncreative.*

The possible answers to these questions were:

Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree

3.7.4 Music Sophistication

The last part of the interface asks the users to answer a selected set of questions from the Music Sophistication Index (MSI) [37]. We collected this data in order to identify potential correlations between more and less musically sophisticated people's scores. The list of the questions asked in this section are:

1. I spend a lot of my free time doing music-related activities.
2. I often read or search the Internet for things related to music.
3. Music is kind of an addiction for me I couldnt live without it.
4. I am able to judge whether someone is a good singer or not.
5. I find it difficult to spot mistakes in a performance of a song even if I know the tune.

6. When I sing, I have no idea whether Im in tune or not.”
7. I engaged in regular, daily practice of a musical instrument (including voice) for —– years.
8. I have had —– years of formal training on a musical instrument (including voice) during my lifetime.
9. I would not consider myself a musician.
10. I am able to hit the right notes when I sing along with a recording.
11. I dont like singing in public because Im afraid that I would sing wrong notes.
12. I only need to hear a new tune once and I can sing it back hours later.
13. I am able to talk about the emotions that a piece of music evokes for me.
14. I often pick certain music to motivate or excite me.
15. I sometimes choose music that can trigger shivers down my spine.
16. Music can evoke my memories of past people and places.
17. Pieces of music rarely evoke emotions for me.

And as with TIPI questionnaire, the possible answers to were:

Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree

Chapter 4

Modeling and Evaluation

In this chapter we describe the methods we used for inferring users' preferences and report and review the results of these methods. We also explain the final recommender system built using the pairwise preferences obtained from the online study.

A study was conducted with participants, using the interface described in Section 3.7. During the course of this study 75 people have provided data. Given that each individual was asked to compare 10 pairs of songs or more, we end up with 739 pairwise scores, of which 638 pairwise comparisons were also enriched by the user's responses to the Five Factor Model and Music Sophistication Index.

4.1 Primary Analysis

The song snippet labeled as *song 30* was the most repeated song in different pairwise comparisons, which is the result of the simulated long-tail effect as in Figure 3.12 and Figure 3.14. The participants were mostly young adults, as seen in Figure 4.1, on average 30 years old with standard deviation 10.4.

The educational level of the participants is plotted against their age in Figure 4.2, to spot the potential wrong answers or outliers. In fact, there are few people who have reported a PhD or Post-doc degree as their education status while being very young. There are also very few High School degree holders above 50 years old. However, the sample is diverse in terms of educational level, and no group is particularly populous with respect to others.

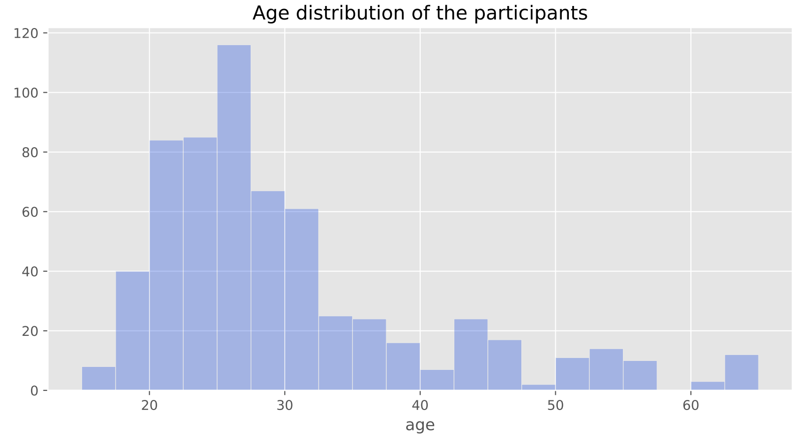


Figure 4.1: The age distribution of the participants

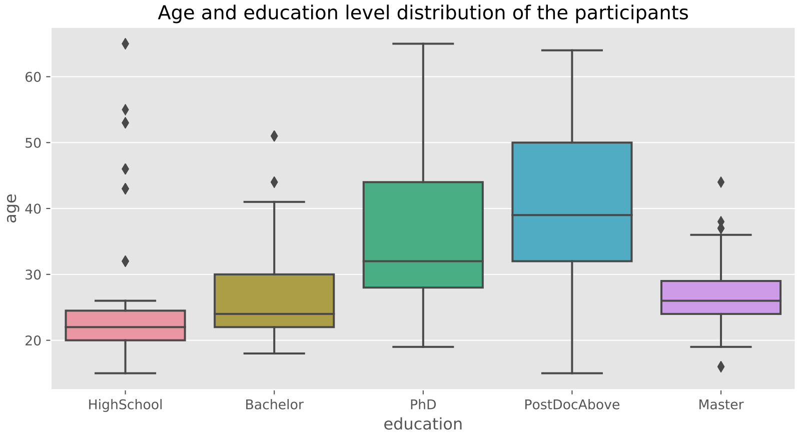


Figure 4.2: The Education Level and age distribution of the participants

There have been more males than females participating in the experiment, as it's evident in Figure 4.3, while half of the population has reported a very good knowledge of written English language. There are however around 60 people who have reported to be less confident in reading and understanding English text. The level of English is important to us because we want to make sure that the participants have fully understood the phrases and questions in the TIPI and MSI questionnaires, since they contain words and phrases that may be unknown to some people.

In Figure 4.5, the participants have generally reported a positive

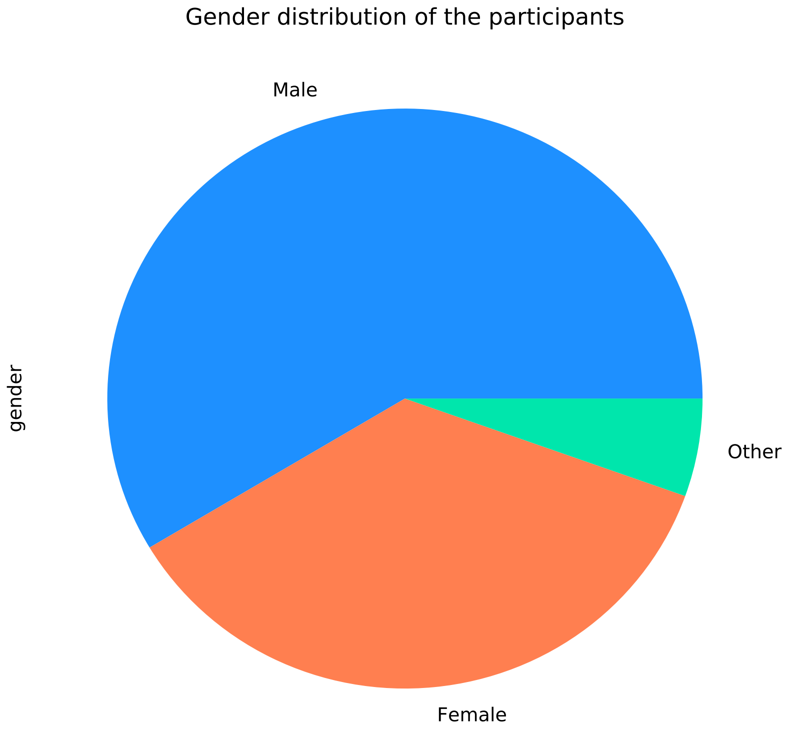


Figure 4.3: The gender distribution of the participants

valence status, while the arousal is rather scattered on the spectrum.

We convert the TIPI questionnaire answers into the values of the five personality factors. The distribution is shown in Figure 4.6.

It is worth noting that there were 13 people who refused to take part in the experiment when they realized that their video stream will be analyzed by the web camera even after we assured them that the video will remain in their computers and will be anonymized anyways.

Inspecting the answers to the Musical Sophistication Index questions, we realized that most of the answers had a normal distribution, apart from those reported in Figure 4.7. The questions are listed in Section 3.7.4.

The distribution of the answers to questions 16 and 17 indicates

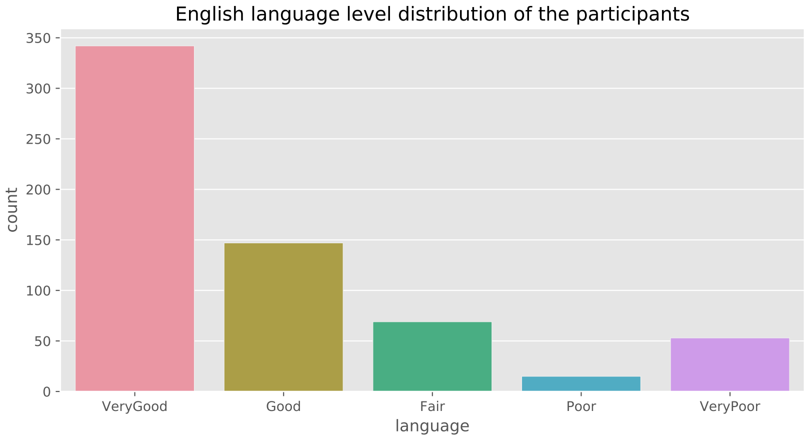


Figure 4.4: The English language proficiency distribution of the participants

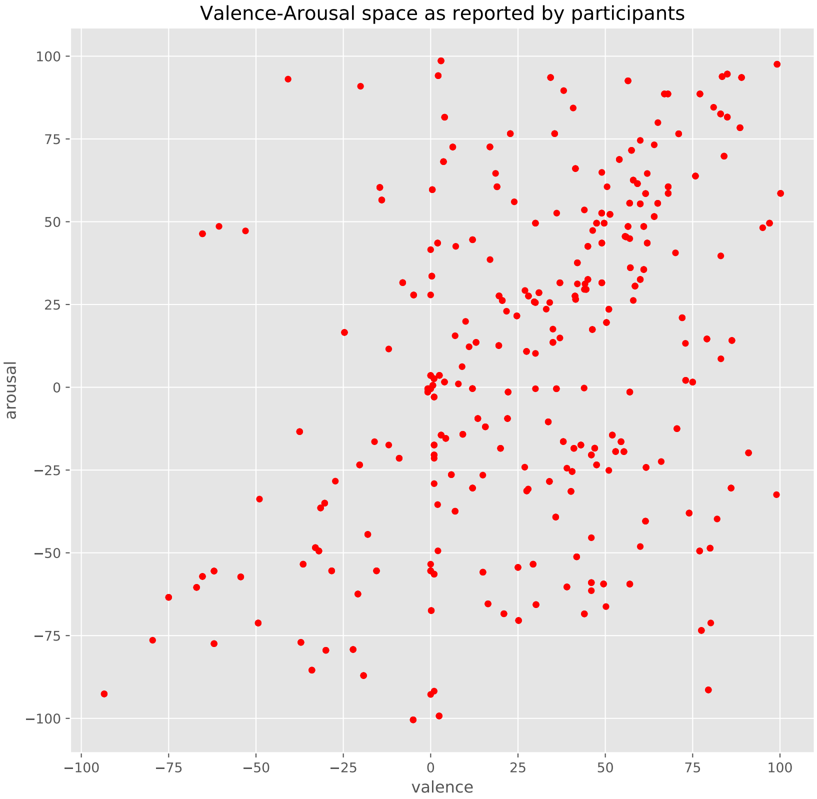


Figure 4.5: The Valence Arousal space of the participants

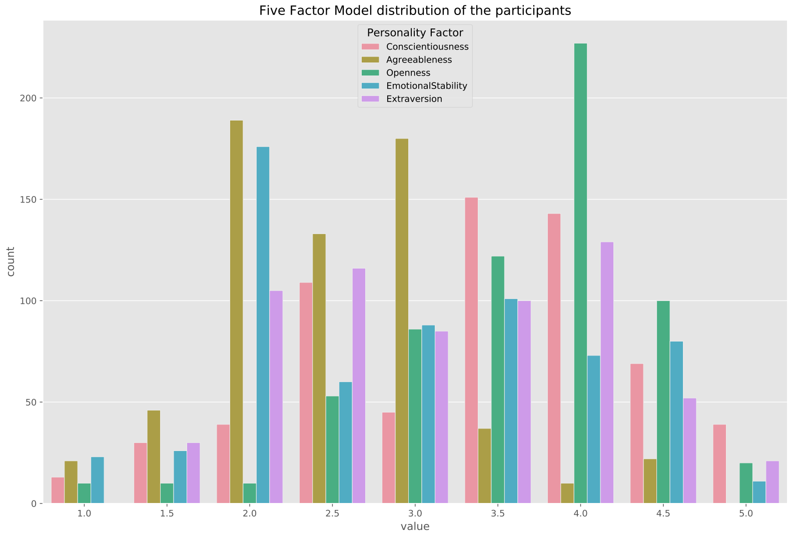


Figure 4.6: The Five Factor Model distribution of the participants

that more people do relate past memories and emotions with musical pieces. Moreover, many people have reported that they are able to judge whether someone is a good singer or not. A considerable proportion of the participants have also reported to use music as an instrument for triggering emotions in themselves.

4.2 Pairwise Score Prediction

The next step in answering our research questions is to investigate the possibility of accurately estimating the users' explicit pairwise preferences by means of their personal characteristics, their interactions with the interface while listening to songs, and more interestingly, their facial expressions.

We aimed at building a model that would predict the pairwise score $\hat{p}(u, k, l)$ a user u would give to a pair of songs k and l. After the customary data cleaning steps and the feature engineering process, the prediction of the pairwise score was modeled once as a classification problem, and then once as a regression problem.

The classifier would yield the predicted score $\hat{p}(u, k, l)$ and we would inspect the confusion matrix and f-measure against the true score $p(u, k, l)$,

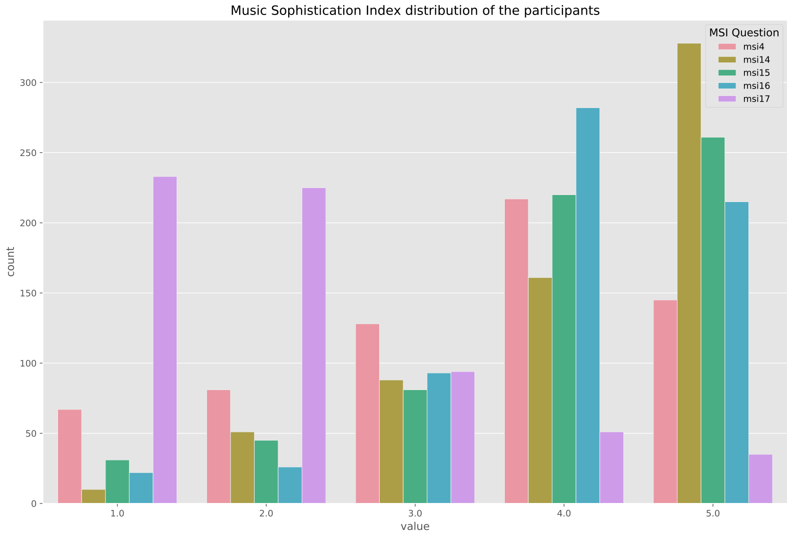


Figure 4.7: The distribution of responses to Music Sophistication Index questions by the participants

while the regressor would output $\hat{p}(u, k, l)$ on a continuous scale and then would be compared to $p(u, k, l)$ using Root-Mean-Square Error (RMSE) and Mean Absolute Error (MAE).

We argue that modeling the problem as regression and classification each have advantages and disadvantages. In classification, although a sharp $\hat{p}$ is predicted, the order of the scores might be neglected, depending on the type of algorithm used. Regression however, does take into account the intrinsic order of pairwise scores. In simpler words, the regressor acknowledges that, for instance, predicting a $p = -2$ (strong preference for the left song) as $\hat{p} = -1$ is preferable to $\hat{p} = 1$, since at least the general preference of the user has been correctly predicted, while classification would see them equally wrong. The same holds for the metrics utilized to measure the prediction quality. However, the downside of regression in this case might be that the users in the online experiment weren't able to express a non-integer pairwise score such as $p = 1.32$, while the regression model might, and will indeed, predict such values. A reasonable compromise in such a scenario would be to perform regression and then in a second step convert the continuous $\hat{p}$ to the closest integer. Such an approach is indeed implemented in some modern classification algorithms such as the Gradient Boosting (GB) [38].

4.3 The Dataset

Our dataset contains 739 pairwise comparisons, of which 638 are accompanied by FFM and MSI data, from 75 participants. Since the facial features of each user on average were sampled on average in 3139 points of time, after data cleaning, there were 235,000 instances in the dataset described by the following static features (by which we mean features that are song/time-independent for each user):

1. *user ID* $\in [1, 75]$, discrete, nominal
2. *age* $\in [15, 65]$, continuous
3. *gender* $\in \{Male, Female, Other\}$, discrete, nominal
4. *education* $\in \{HighSchool, Bachelor's, Master's, PhD, Post{-}doc\}$, discrete, ordinal
5. *English level* $\in [1, 5]$, discrete, ordinal
6. *valence and arousal* $\in [-100, 100]$, continuous
7. *FFM 1 to FFM 10* $\in [1, 5]$, discrete, ordinal
8. *MSI 1 to MSI 6, and MSI 9 to MSI 17* $\in [1, 5]$, discrete, ordinal
9. *MSI 7 and MSI 8*, text

   The following features which vary in each pairwise comparison and user triple $p(u, k, l)$:

10. *song 1 ID (on the left side of the interface)* $\in [1, 200]$, discrete, nominal
11. *song 2 ID (on the right side of the interface)* $\in [1, 200]$, discrete, nominal

12. *pairwise comparison* $\in \{-2, -1, 0, 1, 2\}$, discrete, ordinal

    And the following features which vary in each moment of the user's music listening. Apart from the first one, they estimate the user's emotions and their levels:

13. *current song*, (song 1, song 2), the song that the user was listening to at the moment of sampling.

14. *joy* $\in [0, 100]$, continuous

15. *sadness* $\in [0, 100]$, continuous

16. *disgust* $\in [0, 100]$, continuous

17. *contempt* $\in [0, 100]$, continuous

18. *anger* $\in [0, 100]$, continuous

19. *fear* $\in [0, 100]$, continuous

20. *surprise* $\in [0, 100]$, continuous

21. *valence* $\in [0, 100]$, continuous

22. *engagement* $\in [0, 100]$, continuous

The reader might notice that although the Affectiva SDK also returns some lower level features of the user's face as shown previously, we haven't included them in the list of features. The reason is that all of the 28 facial features of the user (low level features and emotions) should be divided on the basis of the *current song* feature, and they would be doubled. This would yield 56 facial features, and performing the feature engineering steps on them (which will be described in the next section), would generate 696 facial features, which is even greater than the number of pairwise comparisons. Therefore, we decided to work with only the emotions returned by Affectiva.

4.4 Feature Engineering

We used the song/time-independent features as they were in the modeling process. While we generated new features on top of the rest.

We speculate that the more attracted a user is to an online content, the more time he or she will spend interacting with it. Therefore, we described the user's interactions by listening time. For each $p(u, k, l)$ triple we measured the time a user has listened to k and the same for l, to produce the features T_k, T_l and the difference between them as ΔT.

Furthermore, we inspected the levels of each emotion for each $p(u, k, l)$ as time series, for instance, the series of a user's *joy* levels while she listened to *song 1* (joy_k). After manually inspecting a few of such time series for the left and right songs, we realized that the time series trends and shape can be indicative of user preference. We conjecture that, for instance, if a user doesn't like the left song much, there can be a spike in his $disgust_k$.

To summarize these time series we generated different features. First we measured the monotonicity of each of the time series by calculating the Spearman's rank correlation $\rho_{\psi k}$ between the time series ψk and an strictly increasing function, where ψ is one of the user emotions. For example, where $\psi = joy$, $\rho_{\psi k} = 1$ if the user's joy levels have strictly increased during the time she listened to the left song, and $\rho_{\psi k} = -1$ if her joy levels have strictly decreased. Then, we calculated the same for the right song.

Moreover, we fitted second degree polynomials on each emotion time series with the following formula:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

where β_0 is the offset, and β_1 and β_2 are the coefficients. We used β_0, β_1, and β_2 of the curve fitted to each time series as features. This approach was applied to the time series of each emotion for each user while listening to the left and right songs, to yield the features $\beta^0_{\psi l}$ and $\beta^0_{\psi k}$, $\beta^1_{\psi l}$ and $\beta^1_{\psi k}$, and $\beta^2_{\psi l}$ and $\beta^2_{\psi k}$.

Then, we calculated the difference $\Delta\rho_\psi$ between each $\rho_{\psi k}$ and $\rho_{\psi l}$, to understand how different was the user's emotion trend while listening

Table 4.1: Features engineered on top of user's facial expressions, where ψ is an emotion in joy, sadness, disgust, contempt, anger, fear, surprise, valence, and engagement

Feature Name	Description	Range
$\rho_{\psi l}$	Spearman's rank correlation between ψ_l and an strictly increasing trend	-1, 1
$\beta^1_{\psi l}$	First coefficient of a second degree polynomial fitted on ψ_l	$\mathbb{R}$
$\beta^2_{\psi l}$	Second coefficient of a second degree polynomial fitted on ψ_l	$\mathbb{R}$
$\beta^0_{\psi l}$	The offset of the second degree polynomial fitted on ψ_l	$\mathbb{R}$
$\rho_{\psi k}$	Spearman's rank correlation between ψ_k and an strictly increasing trend	-1, 1
$\beta^1_{\psi k}$	First coefficient of a second degree polynomial fitted on ψ_k	$\mathbb{R}$
$\beta^2_{\psi k}$	Second coefficient of a second degree polynomial fitted on ψ_k	$\mathbb{R}$
$\beta^0_{\psi k}$	The offset of the second degree polynomial fitted on ψ_k	$\mathbb{R}$
$\Delta\rho_\psi$	$\rho_{\psi l} - \rho_{\psi k}$	$\mathbb{R}$
$\Delta\beta^1_\psi$	$\beta^1_{\psi l} - \beta^1_{\psi k}$	$\mathbb{R}$
$\Delta\beta^2_\psi$	$\beta^2_{\psi l} - \beta^2_{\psi k}$	$\mathbb{R}$
$\Delta\beta^0_\psi$	$\beta^0_{\psi l} - \beta^0_{\psi k}$	$\mathbb{R}$

to the left song with respect to the right song. We repeated the same for $\beta^0_{\psi l}$ and $\beta^0_{\psi k}$, $\beta^1_{\psi l}$ and $\beta^1_{\psi k}$, and $\beta^2_{\psi l}$ and $\beta^2_{\psi k}$ to measure how different were the curves of the user's emotions while listening to each song. The features generated are summarized in Table 4.1.

Then, we calculated the correlation between the features and the pairwise scores and found several significant correlations. The features with the strongest correlations were features related to the differences of contempt, sadness, joy and valence, as seen in Table 4.2.

4.5 Predictive Modeling

There were two steps for performing predictive modeling: (i) prediction of pairwise scores using machine learning techniques (classification and regression) and (ii) building a recommender system using the pair-

Table 4.2: Features with the highest absolute correaltions with pairwise preferences

Feature Name	Absolute Correlation
$\beta^2_{contempt^l}$	0.242
$\rho_{sadness^l}$	0.202
$\beta^2_{joy^l}$	0.196
$\beta^2_{valence^l}$	0.189
$\rho_{valence^l}$	0.167
$\beta^2_{valence^k}$	0.162
$\beta^2_{contempt^k}$	0.155
$\beta^1_{joy^l}$	0.147
$\beta^2_{joy^k}$	0.142
$\rho_{valence^k}$	0.136
$\beta^2_{joy^k}$	0.134
$\beta^2_{joy^l}$	0.131
ρ_{joy^k}	0.128
ρ_{joy^l}	0.123
$\rho_{sadness^k}$	0.117

wise preferences. We defined an evaluation scheme for pairwise score prediction and building the final recommender system.

Performing the feature engineering steps, resulted in a dataset of 638 instances and the mentioned features, which we call the *TransformedDataset*. We divided the *TransformedDataset* into the following 5 subsets:

1. *PairwisePreferencesTrain (PTR)*: 60% of *TransformedDataset* for training and validating a classifier or regressor for the prediction of pairwise scores in PTT.

2. *PairwisePrefrencesTest (PTT)*: 40% of *TransformedDataset* for testing the classifier or regressor for the prediction of pairwise scores.

3. *RecSysTrain (RSTR)*: 80% of PTT for training and validating a recommender system that provides pairwise ranked recommendations.

4. *RecSysTest (RSTT)*: 20% of PTT for testing the recommender system that provides pairwise ranked recommendations.

5. *RecSysTrainPredicted (RSTRP)*: The same 80% of PTT as of $RSTR$, but with pairwise scores predicted by the classifier or regressor instead of the ground truth pairwise scores, for training and validating an alternative recommender system.

For building the subsets mentioned above, the sampling from the *TransformedDataset* was stratified on the class variable (pairwise score). The stratification was done to ensure that all of the subsets have a similar distribution of pairwise scores to that of the *TransformedDataset*.

For prediction of pairwise scores, in both classification and regression modeling scenarios we used the Random Forest and the Gradient Boosting algorithms. For each experimental setup (i.e. algorithm type, user groups, which we will investigate in the final section of this chapter) we repeated the entire procedure 5 times and averaged the results.

As baseline for pairwise score prediction, we used (i) a classifier that always predicts the majority class in train (in the classification scenario), or a regressor that always predicts the mean of the pairwise scores in train set, and (ii) the Random Forest and the Gradient Boosting regressors using features extracted from the listening time T_k, T_l and the difference between them ΔT, together with the static user features.

The main classification and regression models were trained using the static user features, and the engineered features summarized in Table 4.1. The training and cross-validation was done on PTR, and the quality of models was judged by predicting the unseen PTT.

4.5.1 Classification

In classification modeling, the class variable (pairwise score) of $\{-2, -1, 0, 1, 2\}$ was reduced to a score of $\{-1, 0, 1\}$ by mapping -2 to -1 and 2 to 1.

The results of the classification predictions are reported in Table 4.3 and also illustrated in Figure 4.8. Except for the case of precision score, the Gradient Boosting using the engineered facial features outperforms all the other models.

Classifier	Features	Accuracy	Precision	Recall	F-Measure
Majority Classifier	None	0.476	0.227	0.476	0.307
Random Forest	Facial Features	0.642	**0.617**	0.642	0.610
Gradient Boosting	Facial Features	**0.646**	0.616	**0.646**	**0.617**
Random Forest	Listening Time	0.593	0.545	0.593	0.557
Gradient Boosting	Listening Time	0.593	0.545	0.593	0.557

Table 4.3: Accuracy, precision, reacll, and f-measure of classifiers using facial features (the proposed method) and listening time features (baseline).

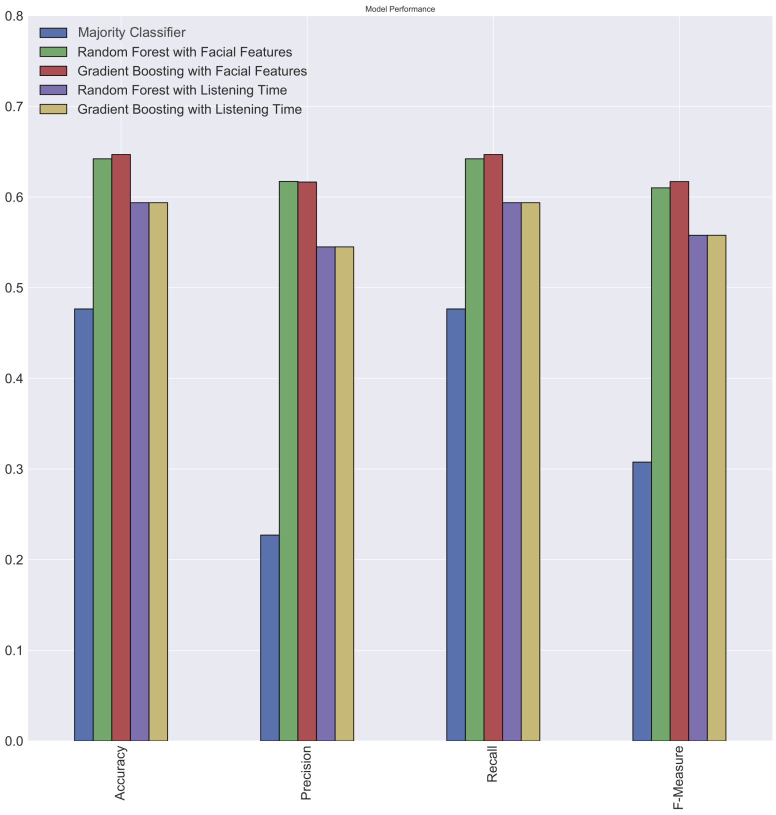


Figure 4.8: Performance metrics of predictions using the classification models. The results are averaged over five runs.

4.5.2 Regression

For the regression scenario we trained models that would predict a continuous value between -2 and 2. Like in the case of classification, the results of the regression predictions are reported in Table 4.4 and also shown in Figure 4.9. The RMSE of the regressor with the proposed facial features are lower than the RMSE of the baseline method which uses listening time features.

Reggressor	Features	RMSE	MAE
Mean Regressor	None	1.26707	0.959344
Random Forest	Facial Features	1.1061	0.839
Gradient Boosting	Facial Features	**1.08113**	**0.832439**
Random Forest	Listening Time	1.26724	0.980468
Gradient Boosting	Listening Time	1.25485	0.960107

Table 4.4: RMSE of regressors using facial features (the proposed method) and listening time features (baseline).

4.6 Predictability of Groups

Besides the global prediction model, we explored whether some groups of people are easier to predict than others. To achieve this we split the users in two groups, trained two separate models (one for each group) and compared the RMSE using the t-test. We perform the splitting in two groups several times, each time along a user characteristic. We used median splitting on the five personality factors.

We found that there were significant differences in the mean RMSE of two groups when the splitting was done on agreeableness, conscientiousness, and openness. The results are summarized in Table 4.5 and shown in Figure 4.10 through Figure 4.12. We speculate that users, who score low on agreeableness, high on conscientiousness and/or low on openness, tend to either show less emotions through their facial expressions or have generally lower variance in their preferences.

However, the differences are not only in the RMSE of the predicted scores but also in the RMSE of the mean baseline. This indicates that the group pairs reported in Table 4.5 differ in the variance of the pairwise scores that the users gave to music pairs. Users who scored

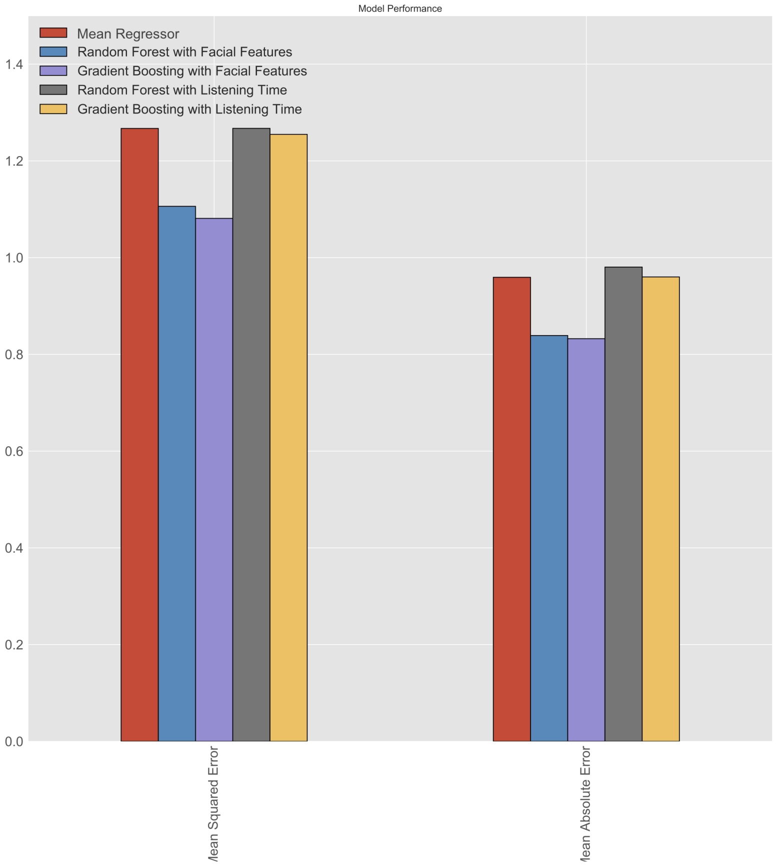


Figure 4.9: RMSE and MAE of predictions using the regression model. The results are averaged over five runs.

higher on openness tend to give less diverse pairwise scores ($\sigma = 0.99$) than users who scored low on openness ($\sigma = 1.39$). Similarly, users with high conscientiousness have less variability in their scores than users with low conscientiousness, and users with high agreeableness have less variable scores than users with low agreeableness. We speculate that users who are open, conscientious and/or agreeable tend to be more hesitant to prefer clearly one option to another and prefer to opt for a neutral pairwise score.

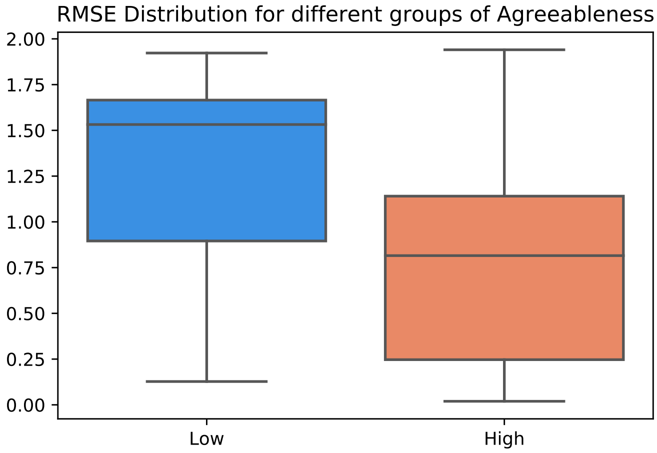


Figure 4.10: RMSE distribution for users with high and low agreeableness.

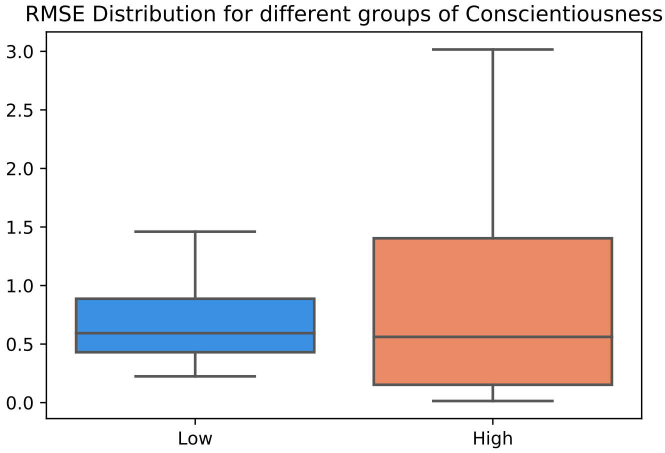


Figure 4.11: RMSE distribution for users with high and low conscientiousness.

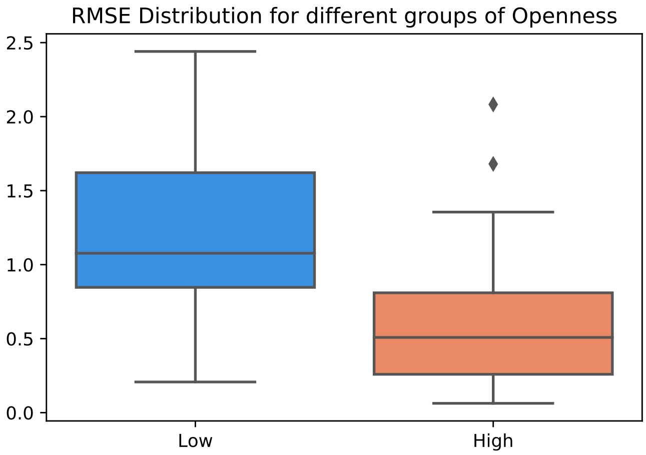


Figure 4.12: RMSE distribution for users with high and low openness.

	FaceFeatures GradBoost	FaceFeatures RandForset	Baseline
High Openness	**0.91**	0.99	0.99
Low Openness	**1.33**	1.39	1.58
High Conscientiousness	**0.97**	0.99	1.05
Low Conscientiousness	1.43	**1.37**	1.51
High Agreeableness	0.96	**0.94**	0.98
Low Agreeableness	**1.39**	1.40	1.68
Global Model	**1.05**	1.07	1.26

Table 4.5: RMSE of the prediction of the pairwise score on the scale from -2 to 2 for different groups of users.

4.7 Recommender System

After performing the pairwise score prediction, we built a recommender system using the pairwise scores in the dataset. we chose the algorithm Pairwise Matrix Factorization (MFP) introduced in [9].

The MFP is a matrix factorization algorithm that takes a set of $p(u, k, l, r) \in P$ as input, where u is a user, k and l are a pair of items, and r is the pairwise score. Then, for each user u present in the set, MFP provides a ranked prediction vector u_ν, containing the combinations of all values k and l in P. Then, u_ν^1 is the highest ranked number for user u by the recommender, and where $i < j$, item u_ν^i is considered more preferred by u than u_ν^j.

The quality of u_ν can be measured with a metric such as RankHit (RH). RankHit measures the ranking error between a set of pairwise scores present in the test set and a ranked list, and is calculated as:

$$RankHit = \sum_{r_{uij}} \in RH(r_{uij})/|T|$$

where $RH(r_{uij})$ is 1 if the RS has ranked for u the item i above item j and the user u does prefer the item i over item j and 0 otherwise.

Using the MFP algorithm, we trained the recommender system in five different setups and measured the RankHit:

1. *Normal MFP Recommender*: Using RSTR as the train set and testing on the RSTT.

2. *Swapping Predictions with Ground Truth*: Iteratively training on RSTRP and testing on RSTT, while at each step replacing one predicted pairwise score in RSTRP with its equivalent ground truth pairwise score in RSTR.

3. *Cold Start*: Iteratively training on n ground truth pairwise scores from RSTR and testing on RSTT, while $n \in [1, |RSTR|]$.

4. *Cold Start + 60 Predictions*: Iteratively training on a train set consisting of n ground truth pairwise scores from RSTR and 60 predicted pairwise scores from RSTRP and testing on RSTT, while $n \in [1, |RSTR| - 60]$.

5. *Cold Start + 90 Predictions*: Iteratively training on a train set consisting of n ground truth pairwise scores from RSTR and 90 predicted pairwise scores from RSTRP and testing on RSTT, while $n \in [1, |RSTR| - 90]$.

4.8 Results

The RankHit scores of the recommender in the five setups mentioned above can be seen in Figure 4.13.

The quality of the RS in the setup *Swapping Predictions with Ground Truth* is lower than the quality of the *Normal MFP Recommender* in the first iterations, because of the inevitable prediction error in classifiers and regressors. However, its RankHit increases gradually and after observing 220 ground truth pairwise scores from RSTR, it reaches the RankHit score of the *Normal MFP Recommender*.

Moreover, the *Cold Start* setup, in which the recommender is trained by simulating a cold start scenario and gradually adding ground truth pairwise scores from RSTR, doesn't perform well in the first iterations. However, when the recommender is assisted with predicted pairwise preferences from RSTRP in the setups *Cold Start + 60 Predictions* and *Cold Start + 90 Predictions*, its RankHit is higher even in the initial iterations.

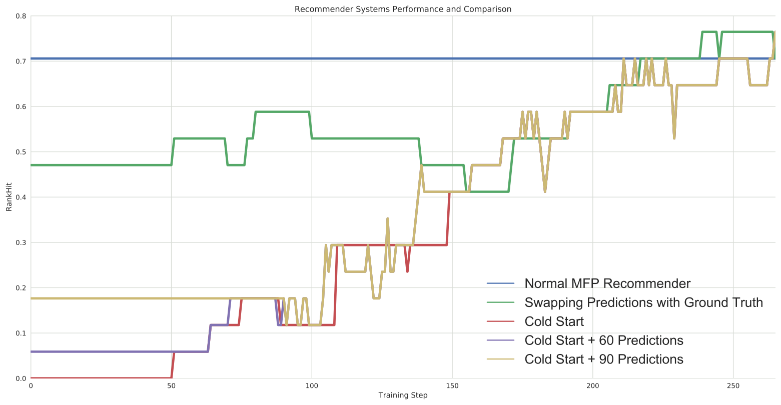


Figure 4.13: Recommender Systems performance and comparison

Chapter 5

Summary, Conclusion and Discussion

In the beginning of this research work we posed the following three **research questions**:

1. **Can we predict users' preferences about items only by analyzing their facial expressions and by means of a supervised machine learning technique?**
2. **Can a recommender system make high quality recommendations using *only* the user preferences predicted from facial expressions?**
3. **How much more implicitly acquired pairwise scores, as compared to explicitly acquired, are needed for achieving an equal quality of recommendations?**

We have indeed found positive answers to all three of them. We have predicted the users' preferences on songs by means of both classification and regression with high accuracy. We have built a recommender system in different setups, which provides ranked recommendations to users, and measured its quality in each setup using RankHit.
Firstly, we trained a recommender system RS_b using the ground truth pairwise scores collected from the users. Secondly, we did the trained another RS_a but using only the predicted pairwise scores. The RankHit in the second setup RS_a was lower than RS_b, but comparable to it.

Then, we trained the recommender system iteratively using the predicted pairwise scores and by replacing the predicted scores with the ground truth equivalents in each iteration. The RankHit of the RS in this setup increased gradually until it reached the quality of RS_b setup after observing 60% of the ground truth pairwise scores.

In the last two setups, we simulated a cold start scenario, where the RS is trained iteratively, each time using one more ground truth pairwise score than the last iteration. The RS in this setup and in the initial iterations struggled to perform. However, when the train set of the cold start setup was enriched with a number of predicted pairwise scores (namely 60 and 90), the RankHit was higher.

We proposed a new approach for using implicit signals to infer pairwise preferences. It uses features based on facial expressions captured during the music-listening sessions. Compared to the baseline method, which uses listening time to predict the pairwise preference, our method performs better in terms of RMSE, and also all other metrics in case of classification. We made a trade-off between (i) intrusiveness/high recommendation accuracy and (ii) unobtrusiveness/lower recommendation accuracy.

We have shown that there are several features that correlate well with the pairwise preferences. Furthermore, we have observed that personality factors account for differences in the accuracy of prediction.

It is worth noting that personality can be inferred automatically from online behavior as it has been shown by many works, such as Kosinski et al. [39] and Skowron et al. [40].

5.1 Future Work

Privacy is an important aspect of our approach. We can easily assume that there will be users who will not be comfortable with sharing the stream from their web cameras. Moreover, data protection regulations can limit the abilities of companies to utilize data extracted from the camera. In our experiment, we did not store any video or images from the cameras. The video was analyzed by Affectiva SDK, which computes the facial features and does not store the video itself. However, further research in terms of privacy concerns is needed before such technologies can become widespread. In fact, one important research direction can be replicating the same experiments by replacing the "sensor" of the user's emotions (web camera and Affectiva) with one that constraints the user less in terms of movement. Using wearable technologies such as heartbeat trackers can be a good alternative to the camera, since an experiment such as ours can be conducted while the user is actively in the context (studying, working, jogging, etc.) instead of having to imagine themselves in such a scenario. A better sensor might also gather more precise and viable raw data, which can in turn increase the quality of preference prediction and recommendation.

Another important experiment to conduct is the generation of pairwise preferences for unseen item pairs. For instance, if a user u has already expressed his preference for item i over j, and j over k, one can conjecture that he would prefer i over k. Such a rule can be evaluated when the ground truth $p(u, i, k)$ is available, and then used to generate pairs that u hasn't even observed before. This technique might uncover another potential advantage of pairwise preferences over absolute ratings.

Replicating the same work using more low level facial features might also be an interesting trajectory, as well as investigating the stability of the results in larger populations and/or different cultural environments and countries.

Bibliography

[1] F. Ricci, L. Rokach, and B. Shapira, eds., *Recommender Systems Handbook*. Boston, MA: Springer US, 2015.

[2] S. Kalloori, F. Ricci, and M. Tkalcic, “Pairwise Preferences Based Matrix Factorization and Nearest Neighbor Recommendation Techniques,” in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys ’16*, (New York, New York, USA), pp. 143–146, ACM Press, 2016.

[3] A. K. Dey and G. D. Abowd, “Towards a better understanding of context and context-awareness,” in *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, vol. 4, pp. 1–6, Citeseer, 2000.

[4] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems*, pp. 217–253. Boston, MA: Springer US, 2011.

[5] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications,” in *1994 First Workshop on Mobile Computing Systems and Applications*, pp. 85–90, dec 1994.

[6] G. Chen and D. Kotz, “A Survey of Context-Aware Mobile Computing Research,” pp. 1–16, 2000.

[7] Y. Zheng, “A Revisit to The Identification of Contexts in Recommender Systems,” in *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, IUI Companion ’15, (New York, NY, USA), pp. 133–136, ACM, 2015.

[8] S. Rendle and C. Freudenthaler, “Improving Pairwise Learning for Item Recommendation from Implicit Feedback,” *Proceedings of the*

7th ACM international conference on Web search and data mining, pp. 273–282, 2014.

[9] S. Kalloori, F. Ricci, and M. Tkalcic, “Pairwise Preferences Based Matrix Factorization and Nearest Neighbor Recommendation Techniques,” *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys ’16*, pp. 143–146, 2016.

[10] U. Böckenholt, “Thresholds and Intransitivities in Pairwise Judgments: A Multilevel Analysis,” *Journal of Educational and Behavioral Statistics*, vol. 26, no. 3, pp. 269–282, 2001.

[11] D. Parra and X. Amatriain, “Walk the Talk,” in *UMAP 2011* (J. A. Konstan, R. Conejo, J. L. Marzo, and N. Oliver, eds.), vol. 6787 of *Lecture Notes in Computer Science*, pp. 255–268, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[12] M. Schedl, A. Flexer, and J. Urbano, “The neglected user in music information retrieval research,” *Journal of Intelligent Information Systems*, vol. 41, pp. 523–539, jul 2013.

[13] M. Zentner, D. Grandjean, and K. R. Scherer, “Emotions evoked by the sound of music: characterization, classification, and measurement.,” *Emotion (Washington, D.C.)*, vol. 8, no. 4, pp. 494–521, 2008.

[14] M. V. Thoma, S. Ryf, C. Mohiyeddini, U. Ehlert, and U. M. Nater, “Emotion regulation through listening to music in everyday situations.,” *Cognition & emotion*, vol. 26, pp. 550–560, jan 2012.

[15] M. Schedl, E. Gomez, E. Trent, M. Tkalcic, H. Eghbal-Zadeh, and A. Martorell, “On the Interrelation between Listener Characteristics and the Perception of Emotions in Classical Orchestra Music,” *IEEE Transactions on Affective Computing*, p. 1, 2017.

[16] P. N. Juslin, L. Harmat, and T. Eerola, “What makes music emotionally significant? Exploring the underlying mechanisms,” *Psychology of Music*, vol. 42, pp. 599–623, jul 2014.

[17] B. W. Schuller, "Acquisition of Affect," in *Emotions and Personality in Personalized Services: Models, Evaluation and Applications* (M. Tkalčič, B. De Carolis, M. de Gemmis, A. Odić, and A. Košir, eds.), pp. 57–80, Cham: Springer International Publishing, 2016.

[18] R. Valenti, N. Sebe, and T. Gevers, "Facial Expression Recognition: A Fully Integrated Approach," in *14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007)*, pp. 125–130, sep 2007.

[19] M. J. Black and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion," *International Journal of Computer Vision*, vol. 25, pp. 23–48, oct 1997.

[20] F. Bourel, C. C. Chibelushi, and A. A. Low, "Robust facial expression recognition using a state-based model of spatially-localised facial dynamics," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pp. 106–111, may 2002.

[21] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan, "Automatic Recognition of Facial Actions in Spontaneous Expressions," *Journal of Multimedia*, vol. 1, pp. 22–35, sep 2006.

[22] T. Baltrušaitis, D. McDuff, N. Banda, M. Mahmoud, R. e. Kaliouby, P. Robinson, and R. Picard, "Real-time inference of mental states from facial expressions and upper body gestures," in *Face and Gesture 2011*, pp. 909–914, mar 2011.

[23] J. F. Cohn, K. Schmidt, R. Gross, and P. Ekman, "Individual differences in facial expression: Stability over time, relation to self-reported emotion, and ability to inform person identification," *Proceedings - 4th IEEE International Conference on Multimodal Interfaces, ICMI 2002*, pp. 491–496, 2002.

[24] M. Tkalčič, A. Odić, and A. Košir, "The impact of weak ground truth and facial expressiveness on affect detection accuracy from

time-continuous videos of facial expressions," *Information Sciences*, vol. 249, pp. 13–23, nov 2013.

[25] J.-I. Biel, L. Teijeiro-Mosquera, and D. Gatica-Perez, "FaceTube: Predicting Personality from Facial Expressions of Emotion in Online Conversational Video," *Icmi'12*, pp. 1–4, 2012.

[26] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, *Semantics-Aware Content-Based Recommender Systems*, pp. 119–159. Boston, MA: Springer US, 2015.

[27] Y. Koren and R. Bell, *Advances in Collaborative Filtering*, pp. 77–118. Boston, MA: Springer US, 2015.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, (New York, NY, USA), pp. 285–295, ACM, 2001.

[29] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4 PART 2, pp. 2065–2073, 2014.

[30] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 734–749, jun 2005.

[31] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, pp. 30–37, aug 2009.

[32] P. J. Rentfrow and S. D. Gosling, "The do re mi's of everyday life: The structure and personality correlates of music preferences.," *Journal of Personality and Social Psychology*, vol. 84, no. 6, pp. 1236–1256, 2003.

[33] M. Soleymani, S. Asghari Esfeden, Y. Fu, and M. Pantic, "Analysis of EEG signals and facial expressions for continuous emotion detection," *IEEE Transactions on Affective Computing*, vol. 3045, no. c, p. 1, 2015.

[34] M. Pesek, P. Godec, M. Poredoš, G. Strle, J. Guna, E. Stojmenova, M. Pogačnik, and M. Marolt, "Introducing a dataset of emotional and color responses to music," *15th International Society for Music Information Retrieval Conference, 2014*, no. Ismir, pp. 355–360, 2014.

[35] M. Tkalcic, A. Odic, A. Kosir, and J. Tasic, "Affective Labeling in a Content-Based Recommender System for Images," *IEEE Transactions on Multimedia*, vol. 15, pp. 391–400, feb 2013.

[36] S. D. Gosling, P. J. Rentfrow, and W. B. Swann, "A very brief measure of the Big-Five personality domains," *Journal of Research in Personality*, vol. 37, pp. 504–528, dec 2003.

[37] D. Müllensiefen, B. Gingras, L. Stewart, and J. Ji, "Goldsmiths Musical Sophistication Index (Gold-MSI) v1.0: Technical Report and Documentation Revision 0.3," tech. rep., University of London Goldsmiths, 2013.

[38] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics, Vol. 29, No. 5*, pp. 1189–1232, 2001.

[39] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, pp. 5802–5805, mar 2013.

[40] M. Skowron, M. Tkalčič, B. Ferwerda, and M. Schedl, "Fusing Social Media Cues," in *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, (New York, New York, USA), pp. 107–108, ACM Press, 2016.