AALBORG UNIVERSITY
COPENHAGEN

Semester:
4th semester

Title:
Recommendations by Emotions Detection through Facial Recognition

Project Period:
01-02-2017 to 08-06-2017

Semester Theme:
Master's Thesis

Supervisor(s):
Iwona Maria Windekilde
Per Lynggaard

Project group no.:
4GUE 4.30+

Members:
Alicia Esquivias Román

Pages:
147

Finished:
08-06-2017

Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV
Semester Coordinator: Henning Olesen

Secretary: Maiken Keller

Abstract: The research in Recommender Systems has evolved considerably over the past years; however, to date the investigation on how emotions could be used to complement such technologies is sparse. This Master's thesis took the initial steps in the research of the inclusion of affective feedback in a recommender system for entertaining videos. The investigation here presented consisted on two related aspects of affective recommender systems: i) finding the factors that could condition the adoption of this innovation; and ii) designing and implementing a data collection process in the form of a web service, due to the lack of data resources for developing this system which was detected, was, resulting in the creation of a dataset with emotional information detected during the visualization of entertaining videos.

After analysing the dataset, it was found that there is a correlation between the emotions and the opinions (provided explicitly by the watchers) on the videos. The study of the adoption, conducted with an adaption of a well-known adoption of innovations model, concluded that the perceived enjoyment of the use of the system and the social influence are the two factors conditioning the most the intention of adoption. Regarding the trust on such a system, the expectancy that the system would use the detected emotions only for creating recommendations is the most influencing factor. The most relevant contribution of the project is the dataset which could be used in future research on the topic, together with the developed methodology and web application, which can be considered as an embryo of a future affective recommender system.

When uploading this document to Digital Exam each group member confirms that all have participated equally in the project work and that they collectively are responsible for the content of the project report. Furthermore each group member is liable for that there is no plagiarism in the report.

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interfaces
ARS	Affective Recommender Systems
CB SEM	Covarianzed-based Structural Equation Model
CBF	Content Based Filtering
CF	Collaborative Filtering
CMV	Common Method Variance
EHR	Electronic health records
ICT	Information and Communication Technologies
JS	JavaScript
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
PLS	Partial Least Squares
PLS SEM	Partial Least Squares Structural Equation Model
RS	Recommender System
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SEM	Structural Equation Model
SEO	Search Engine Optimization
TAM	Technology Acceptance Model
TTF	Technology Task Fit
UML	Unifed Modelling Language
UTAUT	Unified Theory of Acceptance and Use of Technology
UTAUT2	Unified Theory of Acceptance and Use of Technology 2

Chapter 1

Introduction

When Internet was born, it was just used by a few with academic purposes. After some years, the academic world represents only a small part of what the Internet is used for. Now, there are millions of online businesses, it is used to communicate people all around the world, and to distribute content, among others. The possibilities that it offers for content distribution are enormous. A lot of this content has a clear purpose: entertaining the consumer. And a lot of this content comes in form of videos: home or professionally made, all available just a few clicks from the user, ready to be selected and make the watcher enjoy.

However, the huge amount of choices makes very difficult to find quickly the one that suits the watcher's taste. There are many methods used by online businesses to solve this problem, for instance, the recommendations that YouTube produces while watching a video. Those recommendations, however, do not tend to take into consideration the emotions the user experiences; and moreover, they require a lot of user interaction with the system. Emotions are an important factor in determining whether a user likes something or not. In addition, there are occasions where users prefer just to sit and watch, and not having to interact with their electronic devices for a while; it could be because they have this desire or because some physical impediment does not allow them to move or interact properly.

The present Master's Thesis tackles these two problems by investigating how a Recommender System (hereinafter referred to as RS) of entertaining videos could be designed, focusing on the first steps of the process. This system would rely mainly on users' emotions detected by tracking the facial expression for the generation of the recommendations. The proposed RS, after analysing the emotions, would be able to generate recommendations for the video to be watched right after the one that is being watched. The video would be automatically played, minimising the need for the user to interact with the device.

1.1 Background

Due to the appearance of Internet, the society is experiencing a huge amount of changes. Things are done in very different ways than before. It is impressive the fact that 90% of the data that exists currently has been generated in the last 2 years [3]; or the prediction that in

2020 every second nearly a million minutes of video will cross the IP networks, according to Cisco White Paper [4]. Many activities that were previously done in a different context are currently carried out using the Internet, such as buying, watching videos or movies, listening to music, selecting a holiday destination, etc. As a consequence, the online world is full of choices that can overwhelm the users and create them difficult decision-making situations.

This is one of the reasons why RSs come into play. As defined in [5], RSs are software tools or techniques that support the user in the decision-making process by suggesting possibilities that the system predicts the user would like. This is possible thanks to the digital trace that every user of Internet leaves, which allows the RS to *learn* about the user and from the information that it retrieves to generate personalised suggestions. Moreover, supporting the user in the decision-making process is not the only advantage of RS, they also help to increase cross-selling by suggesting products and they improve loyalty since they contribute to improving user experience [6].

RSs are widely used, but so far, there is not an optimal solution [7]. They are different for every context of use; therefore, there are different approaches on how to make or on what to base the recommendations. However, they all have in common the necessity of gathering users' information for creating a profile and selecting the most suitable options for each particular user.

Research on RSs started in the mid-1990s [8] and since then a lot of work has been done on the topic, not only in the research community but also in the enterprise world. There are many online businesses that rely on recommendations to provide an added value to their customers, such as *amazon.com*, using their recommendations to exploit the *long-tail* business model. In 2006, Netflix released a dataset with information about movies and anonymous ratings so experts in the matter could develop a recommender algorithm that improved the one they were using, giving a monetary prize to the most accurate one, showing the importance of recommendations for the company [6]. As the authors in [8] highlight, most of the research in RSs has been done under experimental setups and there is a need to conduct research where the RS is studied under a situation of real use.

The system needs information about the user to generate the predictions that allow it to recommend items. The more information the system has, the better recommendations it will produce. There are two different ways of collecting this information: explicitly or implicitly [9]. When the first one is used, the user provides the information to the system by consciously doing something, for example, rating a movie. In the implicit way, the user behaviour is tracked, not requiring any extra interaction of the user with the system, for example, using as input data the list of listened songs. Another example, which is the basis of the presented project and available thanks to the quick developments of technology, is automatically detecting the emotions the user shows while consuming the content.

The automatic detection of emotions is studied in a field of research named Affective Computing. The term Affective Computing was coined by Rosalind Picard in 1997 [10], who defined it as "computing that relates to, arises from, or deliberately influences emotions". Research on this topic investigates how affective factors influence the interaction between humans and computing and how techniques such as affect sensing could be used to provide information about human affect. Furthermore, another research topic of the field is the design and evaluation of systems that use affect at their core [11, p. 263].

The advances in Affective Computing over the past years are notable, especially in automatic detection techniques [12]. There are different methods for the detection of emotions, such as the analysis of facial expression, body language, voice, biometrics, etc. [11, p. 315].

In 2005, the research community accepted the importance of emotions in RSs [11, p. 312]. Emotions are important to RSs because, as indicated in [11], the consumption of a product is made with the intent of experiencing emotions. Emotions are fundamental to human experience and they impact in daily activities and decision making. For instance, the emotional state influences users' decision to consume the recommended item. However, RSs have largely ignored emotions as a source of feedback or context, because of the difficulties to measure them and that they are easily misunderstood [13]. So far, most of the research on RSs and Affective Computing has been done separately [12].

1.2 Motivation

The idea for this project came up thinking about kids that for some reason (as sickness or physical impediments) must stay for long periods of time sitting or lying, and one of the few things they can do to be entertained is watching TV or visual content. For these kids, it would be very positive if the content played was adapted to their taste. The idea can be extended to people of any age under the same circumstance.

After some consideration, the target group of this thesis was decided to be any kind of internet user, because the concept to be applied would be the same and it facilitated the data collection needed to create such a system. Moreover, once the primary idea is designed, it could be adapted to fulfil the first presented purpose.

On the other hand, the different areas this thesis requires to work on are considered of high interest: RSs are gaining importance as the amount of online choices grows. In both, the enterprise and the academic world, there is an increasing interest in this field. In addition, Affective Computing is a topic that requires a lot of research and development, according to many opinions, it is the direction towards the technology will drift [11, 14]. On top of that, the combination of both techniques is a rather new area which could be widely used in the future.

Finally, the range of applications of an RS using automatic emotion detection as feedback could be diverse. The fact of automatically detecting emotions and minimising the interaction of the user with the system would solve problems such as:

- To entertain people that for some reason (e.g. temporary or permanent sickness) have to stay for a long period of time in bed and cannot interact with a device. Here the system would just generate recommended videos this user would like by detecting the emotions expressed while watching the videos.

- For people that cannot verbally communicate, or cannot verbally communicate emotions, recommending content they would like.
- Or, as it is an emotion aware system, for calming users before a stressful situation, for example, a surgical intervention.

Generally speaking, the use of emotions in the online world has a big potential, and there is a research gap to study how users would react to systems using them and what would the implications of this systems be. These facts make the current study an interesting contribution for future RSs based on emotions.

1.3 Problem Formulation

The field of the problem is very broad, and it cannot be studied all at once because in RSs the type of item being recommended matters for generating the recommendations. The same methods cannot be used for recommending travel destinations as for recommending history books. Therefore, to focus this thesis the content has been restricted to entertaining videos. Nowadays, there are many different videos that are uploaded to the network with the named purpose, and people decide to watch them and like them for many different reasons.

This section presents the research question and the sub-questions that helped to narrow down the project and focused the research to obtain the outcome.

The research question is the following:

> What are the considerations in the design of a Recommender System for entertaining videos that uses automatic emotion detection through facial expression as feedback?

When talking about a system that uses emotions, users can be very reluctant to use it. Automatic emotion detection is not yet a usual practice and it may cause concerns and rejection. Understanding what factors influence the attitude of potential users towards a technology that uses automatic emotion detection can be very useful for designing and adapting these systems in a way that users would find more attractive. Two subquestions are raised to guide the study:

- What Information and Communication Technologies (ICT) adoption model is adequate for studying the adoption in this case?
- What are the factors that influence the adoption of such RS?

The kind of recommender approach presented here is a novel one. The traditional RS uses ratings or binary metrics as the input to the recommending engine. In this case, however, the feedback obtained from the user is emotions. For the traditional RSs, there are several datasets available online, with information about the ratings users gave to items and that can be used to try different recommending methods and algorithms; nevertheless, these kinds of datasets are not available for the recommender problem considered in this project. For the technical aspects of such a system the following subquestions are raised:

- How could a set of emotions be created with the purpose of using it in the study of the proposed affective RS?
- Could the affective information be used as implicit feedback for the generation of recommendations?
- In what ways could the emotional data be processed and used to generate recommendations?

1.4 Limitations

Due to time and resources limitations, the scope of the project had to be constrained.

First, the research will be constrained to recommendations for only one user at a time, i.e., it will not take into consideration that two or more users can watch videos together. Despite being aware that users very often watch entertaining videos in groups, it would imply a big added complexity to the project, so it has been chosen to leave it out for future work.

The automatic emotion detection has been done using an already deployed system, taking the process for detecting the emotions as a black box for the project, just considering the input and the outputs of this black box. The results are conditioned by the capacities of the emotion detection software; however, the work has been done pursuing the best possible results in spite of starting with this limitation.

Despite being an important aspect, the security aspects of this kind of systems had to be left out of the research. Furthermore, privacy was only studied as the *privacy concerns* users might have when using the system, but not from the point of view of how to handle user privacy.

Even though the topic of the research was the design of an affective RS for entertaining videos, it is limited to the initial steps to take in such design. The reader should not expect as the outcome of the project the designed and prototyped system, but the set of first moves that open a path to the future design and research of such a system.

Chapter 2

Methodology

This chapter describes the methodology that was followed during the realisation of the project.

As it was stated before, the purpose of the research is to investigate the initial stages of the design and development of an affective RS.

For answering the research questions, the following process was used:

In the first place, a literature review was conducted with the aim of identifying the status of the research in the affective recommender systems topic, from two points of view, the technical one and the sociological one.

From the literature review, it was found the lack of sources of data on the field. Therefore it was decided to focus the project in the collection of the necessary data for a posterior study.

Secondly, the proposed recommender problem was analysed to find the characteristics a dataset for this purpose should have. Once the analysis was done the data collection platform was designed.

It was decided to collect the data by means of a web service. The requirements for such system were created, in the basis of the needs of the emotional corpus. The system was also designed and implemented as a proof of concept, which was used subsequently to collect the emotions shown by people while watching online videos.

The software development life cycle (SDLC) chosen for the data collection platform was an iterative model [1, 15]. This model was chosen over other options, such as the waterfall model [16] or an incremental model [17] (e.g., the spiral model or the agile methodology). The reasons for choosing this iterative model is that it allows starting without a clear list of requirements which can be completed through the process. This was positive for the current project because it allowed to begin with the design and implementation in an early stage and to enhance it while findings of parallel literature review were obtained. An incremental model was not chosen because as there was only one developer it was counterproductive to divide the product into small parts, being more beneficial to work in the whole model and iteratively add features and improve it. In Fig. 2.1 a diagram of the iterative SDLC is shown.

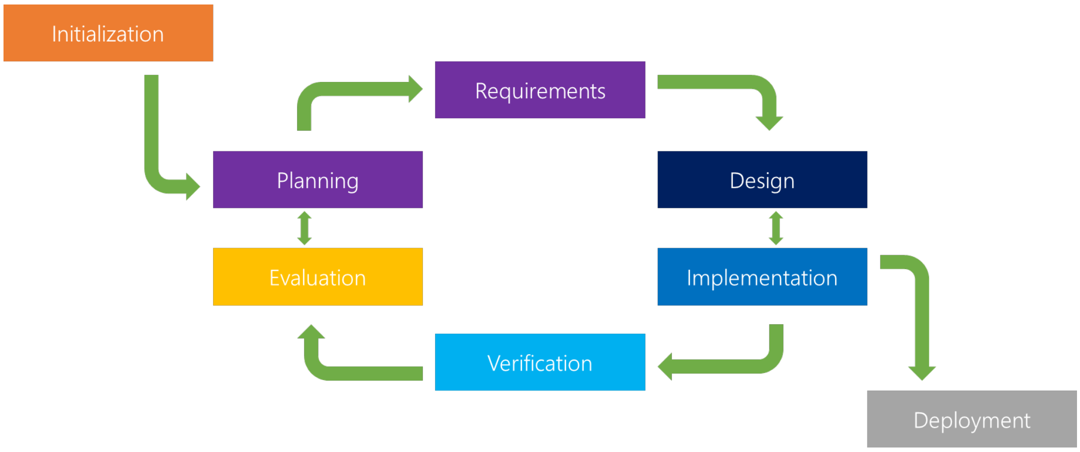

FIGURE 2.1: Iterative SDLC [1]

The data collection platform was thought as a preliminary prototype of the final recommender system. The final set of requirements of the platform was obtained from three different sources.

- The requirements that were noticed to be missing during the design and implementation of the data collection platform.
- A literature review on the State of The Art of RS and automatic emotion detection.

For the collection of the data using the implemented platform, the subjects were asked to watch three short videos. A video corpus was previously created and semi-automatic tagging was used to describe them. While the videos were watched, the software for emotion recognition was working, detecting the emotions and storing them[1]. After watching the videos, the subjects were asked to give some feedback about it.

It was decided to present the data collection platform in two languages: English, as the research was conducting using it; and Spanish, considering that my nationality is Spanish it increased the possibilities of reaching a higher number of respondents, easing the tasks for Spanish speakers.

Once the data were obtained, a first approach for analysing them was taken. The analysis was done by averaging the emotions shown while visualising the videos. A grouping of the feedback given by the subjects of the experiments was performed. The mean emotions were compared for the different groups. A set of suggestions on how to continue the research was drawn from the literature, providing an analysis of what the best option would be.

[1]more details of how the process was done in chapter 4

Affective Recommender System adoption

Following, the methodology used to study the adoption of the recommender system that automatically detects emotions is presented.

The aim of this part is to understand the attitudes of potential users towards an innovation. The purpose is to understand how users perceive the new system and which are the factors having the strongest influence in the adoption of the system.

The concept that is being studied has not been yet developed, is just a concept. This implies that the study of the adoption has to be done from the point of view of behavioural intention since the actual behaviour cannot be known; for instance, this approach is applied in [18].

The approach for this study was deductive; first, a theoretical model was selected and modified to fit the study. The hypotheses for the framework were adapted from the original model or generated from a literature review. The model was chosen also by a literature review of other studies that had common characteristics with the present one. Finally, the choice was to use the Unified Theory of Acceptance and Use of Technology 2 (UTAUT2) [19], with some modifications.

The strategy used to get the necessary data to prove the hypotheses was Survey Research. This strategy was chosen because it enables to collect data from a large number of respondents through questionnaires. As the aim is to explore the general attitude towards the innovation in the Internet users, this strategy was an adequate one. Moreover, it is the most common technique used in the literature about ICT adoption [20, 21], and particularly in the chosen model [18, 19]

The sampling group was selected following a strategy of convenience, which as its name suggests, the group is chosen under the criteria of what is convenient. After, some extra analysis of the data had to be done to assure its validity, because this sampling selection method lacks probabilistic [22].

The survey was distributed on-line. For the characteristics of the project, as it has already been described, a platform for data collection was created. Therefore, it was incorporated in this platform and set up as one of the steps the subject had to take when doing the experiment. The questionnaires were composed by closed-ended questions, with a seven-level Likert scale. The data collected were quantitative, as the aim is to explore the importance of the different factors for the adoption. The questions integrating the survey were, as the data collection platform, also translated to Spanish for the same reason. In this case, the translation was more relevant to assure that the understanding of the questions was the same by all the respondents.

For analysing the data, a structural analysis, using the Partial Least Square method for structural equation modelling (PLS-SEM) was executed. This analysis is composed of two parts, analysing the reliability and validity of the measurements and afterwards, the structure of the proposed model. Both steps were conducted, removing the variables which did not show reliability, validity or statistical significance. PLS-SEM was chosen and as the method for the analysis for being the common method in the literature were the UTAUT2 model is applied [18, 19].

Chapter 3

Theoretical Background and State of the Art

This chapter summarizes the knowledge necessary to understand the topics that this thesis deals with and it also gives an overview of what are the last advances in the field of affective RS, from the technical point of view and from the perspective of its adoption.

3.1 General aspects about Recommender Systems

An RS could be divided into three main components, according to their functionalities [23]:

- Data collection and processing
- Generation of the recommendations
- Presentation of the recommendations

Most of the research done in RS is focused on the generation of the recommendations. However, it should not be forgotten that an RS is an end-to-end system and that all the components are important to provide good user experience. It could happen that the generation of the recommendations is very good; however, if they are not presented to the user in a way that it is understood and facilitates the consumption of the item they would not be valued by the user.

An explanation about each of these parts relating them to the system proposed in the project is given below. These modules are strongly related and the taxonomy of each of them will condition the others (i.e., the recommendations creation model used will influence how the data is processed).

3.1.1 Data Collection and Processing

In RS the collected data are referred as feedback, they are used by the system to learn about the user preferences, inferring them from this feedback. There are many ways this feedback could be obtained. The data collection techniques can be classified in two groups according to how the user interacts with the system for providing the feedback:

- If the user has to consciously interact with the system for providing these data, in the form of a feedback, it is called **explicit feedback**. Some examples of this kind of feedback are asking for a rating of the consumed item or to evaluate it through comments. This method has been widely used in the commercial RSs, such as Netflix asking the users to rate a movie after the visualisation or Amazon that asks users to evaluate the bought products. However, it has some drawbacks, as stated in [24]; since the opinions the users provide are subjective to the context and to their own opinion of what they think an appropriate rating for the item would be and not their real preferences (i.e., users might consider a documentary deserves a better rating while they enjoyed more watching a comedy film), and the user's motivation to provide the feedback is not too high.
- The other way this feedback can be obtained is by tracking the users' activity when using the RS and extracting the information from that activity. This method, called **implicit feedback**, is more complicated and requires more data and, especially, more processing of the data, but at the same time it is more accurate for obtaining the users' opinion and the feedback is easier to be applied because it is done just with the users' interaction with the system. Therefore, there is a tendency in RS to use and migrate to implicit feedback [25]. Some examples of implicit feedback are tracking the links that are clicked, the songs the user listens to, as Spotify does, or the saved research papers, as Mendeley Suggest does.

In the case of this thesis, the feedback method is constrained by the research question "What are the initial considerations in the design of a Recommender System for entertaining videos that uses automatic emotion detection through facial expression as feedback?".In other words, the feedback is obtained by implicit means, by detecting the emotions shown by the user through facial expression. However, defining the kind of feedback that is going to be used is not enough to define this part of the RS. It is also necessary to establish how it will be obtained and how it will be stored and preprocessed, preparing the data for the recommender module to compute the recommendations.

There are several options for extracting the emotions from facial expression. It is an image processing task that has a lot of machine learning behind. There are some Application Programming Interfaces (APIs) available that do this processing and that have already aggregated data from many different people of diverse ages, genre, and nationalities. Some of these APIs are: Microsoft Cognitive Services Emotion API [26], Affectiva Emotion Recognition Software [27], Nvisio [28] or Google Vision API [29] In Chapter 4 an analysis of these APIs is presented and it will conclude with a choice of the API to use. This choice is relevant for considering the rest of the aspects of the data collection and processing module. Each of the APIs needs a different input and provides a different response, which will condition the way the face of the user is captured (a video, images every x seconds...) and the way the response is stored and processed by the recommender algorithm.

3.1.2 Recommender Engine

This component processes the information obtained from the data collection and generates recommendations of content, videos the case of the studied RS, personalised for the active user at the moment. It is accomplished in many ways which depend on the characteristics of the users, the items to be recommended and the available information. This aspect of RSs has been widely researched. In the literature, a lot of taxonomies describing the components of the recommender engine is found. In Fig. 3.1 a taxonomy of the different concepts, as presented in [2] is shown.

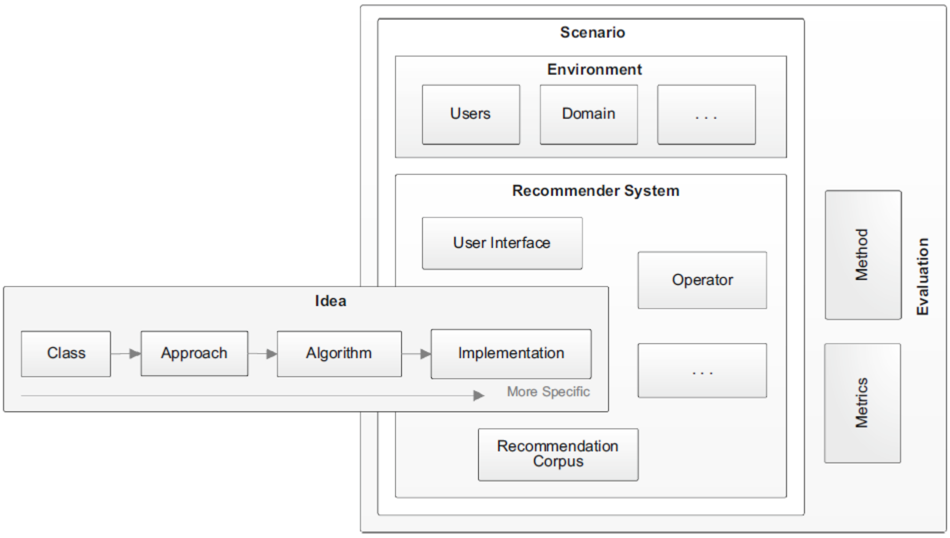

FIGURE 3.1: Recommender System concepts [2]

Mainly, the function of this element is predicting to what extent a user would like or would be interested in an item. Recommender approached are classified in the literature under different criteria, lacking consensus among researchers on this topic; as stated in [6], there is no common framework and, consequently, every researcher or expert on the topic follows its own approach [2, 5, 6, 9]. Nevertheless, there are three that are always present, which are described as follows:

- *Collaborative Filtering(CF)*: In this method, the characteristics of the set of items that are being recommended are *not* taken into account, which means -theoretically- that the taxonomy of the recommender item is not relevant. For instance, the algorithm would work in the same way for online lessons or social network friends. What is used for generating recommendations are the experiences provided by other users of the system, based on the assumption that "people who agreed in the past tend to agree in the future". The way they work is by predicting a measure of how much a user would like an item. This is done by taking the previous interaction of the user with the system and comparing it with the interaction of other users. Collaborative filtering has been researched widely and the recommending approaches can be classified in two different ways:

  – According to the main focus of the approach they can either be *item-to-item* or *user-to-user* systems. In the first case, the predictions are done focusing on the items, finding items that had a similar reaction by users. In the second case, the predictions are done for the users, by using how others' opinions on the items the active user is missing[5]. Even though these two approaches are similar they can create very different recommendations [30]
  – According to the way of computing the predictions there approaches can be either "memory" or "model"-based. The first type computes the predictions of ratings directly from the ratings that had been collected previously. In contrast, the second one uses the collected ratings to create a predictive model, and this is used to compute the recommendations.

- *Content-Based Filtering(CBF)*: This group of RS uses information of the past behaviour of the user that is going to get the recommendation combined with a set of characteristics of all the items that are in the recommender catalogue. By analysing the features of the items, the system finds other items that share, to some extent, the same characteristics. Once this is done, the system is able to find new items for the user by recommending those that share characteristics with the ones that have already been consumed and liked by the user[5, 6, 9].

- *Hybrid*: All the RS approaches have pros and cons. Another class of RS combines two or more recommending techniques to avoid the problems that could be encountered using only one approach. Typically the combinations use Collaborative Filtering and Content-Based Filtering methods. They are combined in several ways, such as assigning weights to the predicted rankings through the different methods, switching from one technique to another, mixing them or connecting them in cascade [31]

RSs present challenges that should be considered when designing a new one. Every Collaborative Filtering system faces the two following:

- *Cold Start problem*: as the CF uses previous interactions with the system to create the recommendations, there is always a moment in which the system does not have enough information to fulfil this task. The cold start problem can be of three different kinds. When the RS is starting there is the *new community* problem, typical ways of avoiding it are to use different techniques to encourage users to rate the items or to start using a CF approach when there are already enough ratings. The *new user* problem is the most critical one since users want to have a personalized experience from the first moment and the contrary might cause them to stop using the system; a way of dealing with this problem is to use more features apart from ratings, such as gender or age, to create the recommendations. The last Cold Start problem is the *new item*, that is possible to minimise by having users deliberately consuming and rating the new items, or by adding new means of discovering new items[9]

- *Sparsity*: This happens when a number of users and items is big but there are not enough ratings, implying then that there are users or items with no ratings in common. The sparsity problem can be solved using dimensionality reduction techniques[5, 9].

In Content-Bases systems the faced problems are different. One of their main challenges is the difficulty to get description about the items that can be used to calculate similarities. Another major problem is the *overspecialization*, which makes the system only presents items very similar to the ones that have already been consumed.

The kind of content to be recommended in the proposed RS is entertaining videos obtained from YouTube. Those videos might lack enough metadata they are of many different kinds; therefore, choosing a content-based approach would have an important drawback on how to characterise all the videos.

A CF approach would solve this problem. Moreover, it is proved that, in general, CF approach works better than CBF [23]. In a CF the kind of content being recommended would not matter, and instead, a model out of the behaviours of the users would be created. However, a CF approach would present the problems that have been stated earlier. This could be solved by using a **hybrid approach**, combining a content-based approach and collaborative filtering approach, obtaining this way the best out of the two models.

The best way of designing the system and finding the videos a user would like is to actually have data of what the users like. These data would help to understand even better the recommender problem and try different algorithms and combinations of approaches measuring their performance with off-line metrics, for afterwards choosing what would be the best design for the system.

3.1.3 Presentation of the Recommendations

Presenting the recommendations is a very important part of the recommender process. This is how users receive the recommendations and it will condition their reactions. There are many ways of presenting the recommendations:

- Top N: presenting a set of the most relevant items for the user. For example, YouTube presents these recommendations to a user watching a video.
- Related items: Approach that presents items with some association the user is consuming. One example is Amazon and its known "people who bought ... also bought ... "
- Categories: Presenting the items in groups according to some common feature. This is the approach Netflix uses, adding a classification of the items according to a common feature, such as the genre of the film [32].
- Single item recommendations: approach in which the top item predicted by the recommender engine is presented.

- Sequence of items: where a list of items is created to be presented after each other. In this case, the order in which the items are presented is important. It is the approach used by Spotify for generating playlists.

The intention of the proposed RS is to minimise the interaction of the user with the system, therefore the videos should automatically start playing every time one video ends. This leads only to two possible ways of doing it, by presenting the top item or by creating a sequence.

3.2 Affective Recommender Systems

In this section, the latest works done in affective RSs will be presented, giving insights of what has been done until now and where the research gaps are.

Tkalčič et al. in [12] present one of the few proposed frameworks for understanding better the role of emotions in an RS and for unifying the work that had been done until that moment (2009) on the topic. They define three stages in the consumption chain of an item: the entry stage, the consumption stage, and the exit stage. In each stage, the influence and the role of the emotions are different and could be used in distinct ways to generate recommendations. This thesis considers the three stages for the recommendations, including them to the research and data collection process.

This same group of researchers created the only database available for affective computing and recommender systems: the LDOS-PerAff-1 corpus [33] (at the date of writing and according to the findings in the literature review). It is a corpus of video clips of people affective response to a set of stimuli. The participants were asked to rate images as if they were choosing one for being their background and at the same time their emotions were being tracked. Even though the corpus was decided not to be used in this thesis because the images cannot be generalised to the recommendation of videos, a similar approach was being used for collecting the training data for the recommender system.

Continuing with the research mentioned in the paragraph above, the same group of researchers proved in [34] that the emotions detected through facial expresion improved the performance of an RS for images. The findings from the research establish a good perspective for the research conducted in this Master's Thesis

According to the findings, until the date, the only existing survey on Affective Recommender Systems is the one presented in [35]. In this work, the authors review the literature on the topic until 2016 and they present different classifications of the research that has been done, as well as what they consider that should be the research focuses. The work has a broader scope than the present revision of the state of the art, however, is a good source to get a general view in this field.

The most similar work to the area of this thesis that has been found is presented in [36]. The authors create a recommender system for movies that uses facial recognition with a hybrid approach, combining Collaborative Filtering with Content-Based Filtering. Their goal is also similar: create a system that does not disturb the user to provide information. However, even though they use facial recognition, they use it for different purposes. The first purpose is to detect the gender and the age of the user; the second one, to assign a genre to the movies from the emotions shown by the user while watching the trailer. Moreover, they give the option to rate the movie and write a comment, these comments are analysed using sentiment analysis techniques and combined with the rating the user gave, they create an average rating for the movie. These ratings and the genders detected while watching the trailers are used to generate the specific recommendations for each user. They have an actual implementation of the website that they use in live for evaluating the system. However, there are no metrics or data presented about the effectiveness of the recommending system.

The authors in [37] take a similar approach as well, using facial recognition in a product recommendation system. The facial recognition is used to determine the gender and the face of the user - so there is no log in required - and to detect emotions. Similarly to our idea, these emotions are used as the feedback of the system, using information about if the user is happy or not to generate the recommendations. For that, they use a collaborative approach that separates females from males and k-nearest neighbour for finding similar users and the Fisher algorithm for the classification of happy or not.

In [38] the authors analyse the semantics of tags that a set of users had assigned to music to classify them in positive or negative emotional tags or factual tags. They use a user-collaborative filtering approach and they use precision metric for its evaluation.

Creating recommendations for YouTube videos, as our RS does, is something that has been done in several works. For example, in [39] they use YouTube video's comments for generating recommendations of educational videos. The YouTube API is used to retrieve the comments of a set of videos. They conduct a sentiment analysis on the comments, tagging each video with one or more emotions. As user feedback, they collect the emotions felt by the users during the interaction explicitly, by making them select one out of the seven presented emoticons representing emotions and if they liked or disliked feeling that emotion.

Another work using YouTube is [13]. The authors try to find out the role of emotions in short films available on YouTube, and if it is possible to annotate the emotions present in the short film by analysing the comments, as an alternative to human explicit annotations. They use for describing each film a set of eight emotions (four pairs of contrary emotions, e.g., joy and sadness). They apply two approaches for annotating the movies, the first one explicit annotations obtained by crowdsourcing and the second approach by automatically extracting them from the comments. They study the role of emotions in short films. In their results, they present which emotions are more present when the subjects have to choose movies with a determined purpose. They find out that the recommendations generated by the human annotated videos are very similar to the automatically retrieved annotations. They use a collaborative approach with a learning-to-rank algorithm.

The work presented in [40] is on information retrieval, but there are things that can be used in RSs. They present that, when the user is looking for entertainment is because there is a need to enhance or retain positive states and to lessen or steer of negative ones. They conduct an experiment to detect what emotions users experience while conducting different video search tasks. They use facial expression recognition and biometrical signals to detect emotions and they make the participants rate the watched videos. They found out that facial expression is the best method for the detection of the emotions.

An improvement for a web-based RS is proposed in [41] by the use of emotions. They use explicit emotion detection, by making the user to choose three colours. Then, the detected emotions are combined with a hybrid approach (collaborative and content-based filtering).

In [42] a Context-Aware RS is designed, using the emotions as the contextual information. They use the LDOS-CoMoDa[43] data set to compare two different approaches to context-aware recommender systems with a non-context aware RS. They do the comparison using the RMSE metric. They conclude that when the emotions were included as context information the system performed better than when they were not, which is a good starting point to support the idea of this research.

[44] is the only paper found describing an architecture for an RS involving emotions. It describes the architecture of an adaptive learning environment, which adjusts to the state of the user and to some contextual information. It uses a cloud architecture based on different layers and using APIs for the communications

A recent work in the field is [45], that uses emotions as the implicit feedback for creating recommendations for educational videos. They acquire the emotions from three different measurements. Two of them is through facial recognition, by using the FaceReader software, that detects emotions and classifies them into six categories: happy, sad, angry, surprised, scared, disgusted and neutral. This software can also detect the Arousal and the Valence of the emotions. The third way of detecting the emotions is through the psychological responses. If two of the three methods show negative emotions the system suggests stopping watching the clip and move on to the next one. For selecting the videos they analyse the emotions present in the texts related to the videos and choose the videos according to the level of the emotions present there.

In the European project "Emotion-based analysis and recommendation of lectures" [46] the emotions presented in Ted Talks are automatically detected and then used to generate recommendations. Another European project named "Mixed Emotions" [47] is creating an emotion aware RS for Apple TV by analysing the emotions present in the content and the social media for the topics covered by the video.

The enterprise world, Skyscanner Russia [48] is one of the first companies to use an RS that reads emotions through facial expression recognition and then generates travelling recommendations. Affectiva [27], which is one of the pioneer firms working with affective computing, has a patent from 2011 for an RS using facial expression as feedback[49].

3.3 Adoption of Innovations and Recommender Systems

The adoption of innovations refers to the process in which an individual decides to use a new technology in an environment or in an organisation. The adoption is closely related with the term diffusion of innovation, however, it is important to understand the difference among the two: adoption focuses on the individual acceptance of the innovation, while diffusion looks at the aggregation of the adoption in a group [21].

Most of the work done on the topic of ICT adoption of innovations has been focused in company environments, and that is why most of the models for studying adoption consider the new technology as a new way of doing a determinate job or task. However, in the last years, changes became radical not only in the working places, but also in our everyday life, and the motivation for individuals for accepting these changes has become an important subject of study in both the academic and the enterprise worlds [18]. Consequently, the models for studying the adoption have been extrapolated and used also for studying how and why individuals accept innovations outside a working environment.

The study of the adoption of innovation in the academic world has different purposes. It can be studied for understanding why things have happened in a certain way, such as what they do in [50], where the authors examine the adoption of ICT for understanding what psychological factors are causing the digital divide. It can also be studied to compare different implementations of the same concept to understand what are the factors that condition the choice of one over the other [51, 52] or simply what are the factors that condition the adoption of a technology [53]. There are also other works that study adoption as a feedback to improve the design and implementation of a system [53]. Other researchers focus on developing new models and proving that these models are valid for its purpose [54]. Adoption can be studied as something that already happened, by studying past behaviours, or, as done in [53] or citeUTAUT2RS1, it can also be studied as the behavioural intention.

Different models have been developed for the study of the adoption of innovations. An example is The Technology Acceptance Model (TAM) [55], that has been widely used for the study of the adoption of ICT. After this model appeared, modifications of it were proposed, for later being aggregated in the Unified Theory of Acceptance and Use of Technology (UTAUT) [56]. These two models focus on the new technologies as supporters for job task. There are other models for studying ICT adoption, such as the Task-Technology Fit (TTF) [53] or the attribute-perception-intention model, used in [57]. Some other researchers choose not to use any already existing framework, and they just propose one that suits their research [50]. As for research methodologies, there are works that use an empirical method, and others use non-empirical. The type of data used has been both quantitative and qualitative, is the former one more usual. The methods used include surveys, interviews, studies, as well as literature reviews [55].

Some of the previous works on adoption of RS or similar systems are presented in the following paragraphs.

In [53] the researchers develop a model combining the diffusion theory of innovation and the Technology Task Fit model for finding out the factors that have influenced the adoption of Personal Information Systems - such as smartphones. Their research, as it uses the TTF, studies how the innovation characteristics support the performance of a specific task. With 5 hypotheses, the tasks are linked with the characteristics presented in the Diffusion of Innovations theory as the important ones for an innovation to be adopted. This paper is relevant to the project because it studies what are the attributes that condition the adoption of an innovation through studying the intentional behaviour, not the actual behaviour.

An example of a research conducted using TAM is presented in [58], where the authors add some modifications to the model in order to study what are the factors that influence the Perceived Ease of Use and the Perceived Usefulness of an e-learning Recommender System. The paper only proposes the framework and explains how it would be analysed, however at the time the paper was written the experiment for testing it had not been yet conducted yet.

In [59] the authors use TAM to conduct the verification of an RS proving that in such a system there are more factors than just the accuracy of the recommendations that condition the users' choices of using or not using the system. This affirmation is one of the bases of our research since we want to find these factors for our RS and use them for designing it in the best possible way.

The model UTAUT is used in [51] to compare the behavioural intention to use an RS implemented in two different ways, one using content-based filtering, and the other using collaborative based filtering. They also study if the kind of product being recommended influences the acceptance of the system. Trust is added to the UTAUT model since they consider it as an important factor conditioning the adoption of an RS. Also, the *facilitating conditions* construct is eliminated from the model because they are interested in behavioural intentions.

The authors of [19] present a modification to UTAUT, called UTAUT2, which modifies UTAUT to fit in a customer use context. In [18] UTAUT2 is used in the context of a Social Recommender Systems, as the TAM and the UTAUT are more focused on studying innovations in an organisation, and UTAUT2 is tailored for the context of consumers acceptance. As in [53], the adoption is studied as the behavioural intention. The authors propose three factors that might or might not influence the adoption of a social recommender system, and they state that the results of the study can be used as the basis for the design and implementation of such Recommender Systems. Another example of the use of UTUAT2 can be found in [60]. In this case, it is not a Recommender System's adoption the one that is being evaluated, but an augmented reality application for a smartphone. By empirical methods, they evaluate the relation between functional system properties, emotions, and adoption behaviour. They used UTAUT together with the pleasure, arousal and dominance model for representing emotions. For combining these two models together they use the Stimulus-Organism-Response model. For the experiment, they test the app with tourists and then ask them their intention to use it again. They also presented two versions of the app to two groups, one that offered personalization and the other that did not. They found out that personalization was not a factor conditioning adoption, and that only pleasure and arousal were significant for understanding user behaviour.

Chapter 4

Data Collection: Analysis

This chapter discusses the findings from the literature review to apply them to the current research, defining the path of the project on basis of these findings.

4.1 Motivation

From the literature review it was found the low number of works that conduct a similar research to the one being presented in this Master's thesis. Furthermore, no dataset that could be applied to the recommender problem being researched was found. As stated in [61], one of the problems of the RS research is the little availability of datasets than can be used to develop and evaluate a system. In a field as the one being studied the problem is aggravated since the input data to the system are not the common ratings, but some data completely different and considerably less explored by the research community, as the emotions are.

On the word of the researchers in [33], a proper dataset is an essential element in the affective RS investigation. Even though there were some similar researches found, these do not provide with the resources used for their investigation [36, 37].

On basis of these findings, it was decided to build a dataset of the emotions induced by video visualisations that could open the path for future research. In this chapter, the data collection procedure is presented. The purpose of the data collection phase was, consequently:

- Get emotive data of users for the generation of a dataset that could help to understand the relation between the emotions and the opinion about a video; try and evaluate different recommending approaches and generate a model for creating the recommendations.
- Get answers to a questionnaire about the acceptance of an affective RS, in order to infer which are the main factors influencing the users' attitude towards it.

To this end, a Web Application was built. The reasons to choose a Web Application as the platform for collecting the data are listed below:

- Reaching as many respondents as possible: The quality of the results will depend upon the number of subjects that take part in the data collection phase. On both the emotional data and the questionnaire data is necessary to have a lot of responses. The first one for being the input data to a machine learning problem and the second one for being statistically accurate.
- Avoid the *laboratory-effect* by emulating the real situation: the subjects of any experiments tend to be biased for knowing that they are under study [62]. Being able to choose when and where to do the experiment, instead of doing it under a laboratory set up, reduces the effect.
- Making it as similar as possible to how the real experience would be: in this way, more accurate results would be obtained. Regarding the response of the questionnaire, the subjects would get an idea on how the system being studied would be.

The choice of collecting the data through a web application was backed-up by the actions taken by the researchers in [63], who use an online application to collect ratings on movies with contextual information.

During the rest of the section, an analysis of how the studied RS would be and how the collection of the data should be done is provided, resulting in a set of requirements for the implementation and design of the data collection platform.

4.2 The Recommender Problem

In the initial phase, the recommender problem should be properly defined and understood. As a reminder, the research question is repeated here, since it is the question defining the recommender problem:

> What are the considerations in the design of a Recommender System for entertaining videos that uses automatic emotion detection through facial expression as feedback?

Therefore, from the research question it can be extracted the following recommender problem, described using the taxonomy proposed in [32]

- **Domain** - in other words, the kind of content that the RS is going to recommend. As the research question states, the content for this RS is entertaining videos. It is important to know what is being recommended and the main characteristics of it because they will condition most of the design choices. There is a lot of videos online, of different characteristics because of the ease of uploading content that the Internet brings. Most of these videos are hosted by YouTube, the biggest video sharing platform; therefore, this project will use YouTube as the source of the videos to recommend. As the purpose is to recommend entertaining videos, out of all the available content in the platform, only the one that was uploaded for this purpose will be taken into account (videos for other purposes, such as teaching, advertising or informing, for instance, are not considered). Some characteristics of the YouTube videos are listed in the following:

- There are videos of very different sources - ranging from professional (e.g. TV shows) to homemade videos.
- The duration of the videos is also very variable, from a few seconds video to some hours.
- The metadata available with each video is not always the same. The uploaders are the ones in charge of completing this information, therefore it is not consistent. Some videos have a lot of information whereas others are only provided with a title.

From this characteristic, the first requirement is drawn:

- *the platform must contain a catalogue of YouTube videos selected under a certain criterion.*

A second requierement can also be extracted:

- *The system must embed YouTube videos (to make the visualizations possible)*

- **Purpose** - this aspect can be looked from two points of view, the users' and the providers' of the service. In the case of the proposed RS, the purpose from the user's point of view is to facilitate the automatic consumption of the content, minimizing the amount of interaction between the user and the system (of course the purpose is to help the user discover new content he likes, but that is inherent to any RS). The purpose for the service provider in this system would be to increase the content consumers loyalty.
- **Context** - or the situation the user is in when he receives the recommendation. This RS would be used for entertainment purposes, which means that, even though the context can vary a lot, it will be in a place where the user is laid-back and comfortable and under suitable environmental conditions.
- **Personalization level** - the recommendations in this RS should be personalised to the user using the system.
- **Whose Opinions** - this aspect refers to based on what the recommendations are produced. For example, they could be created by considering the opinions of a set of experts on the topic, or as in this recommending problem, by taking into account the opinions of the mass.
- **Privacy and Trustworthiness** - in this recommender problem the privacy is an important aspect since it will be dealing with emotions, which are very sensitive, and

with personal information. The Trustworthiness in the system is not as relevant since the consequence of giving a bad recommendation would only be that the user would watch a video that he does not enjoy. This has no major effect since the visualisation of the video has no cost to the user (YouTube videos are available for free) and he is not forced to finish the visualisation if he does not like the content.

From the former paragraph, another requirement is built:

– *The users' personal information and information about the visualisation should be secured. The collected information should be anonymous.*

- **Interface** the designed recommender system should minimise the interaction between the user and the system, therefore the interface should be designed to fulfil this premise.

  In order to make the data collection platform as similar to the future product as possible, two new requirements are drawn:

  – *The design must be consistent over all the web pag*e
  – *95% of the subject should not find difficulty navigating the page*

- **Algorithms** the algorithm to use, even though is a big part defining the RS, is one of the components to be determined after conducting research and concluding what option performs better. Therefore little affirmation about them can be done at this point.

4.3 Scenarios and Use Case

The Use Case of the system is presented in this section. In order to get a better understatement of the software system that is being studied, an affective RS for entertaining videos, and to use it in following steps of the design and as a support for the establishment of requirements. For that, a Use Case diagram is used, following the specifications of the Unified Modelling Language [64].

Before the Use Case is presented, different scenarios of the use of the system are given:

The possible uses of the RS are differentiated mainly by two aspects. The first aspect is the number of people watching the videos and the second one is how the user starts playing the first video. For describing the scenarios the name *Recomotions* will be used for referring to the affective RS for entertaining videos [1]

- Scenario 1: A user decides to watch entertaining videos making use of *Recomotions*. He wants to watch a sequence of videos that he would like, and he does not have anything concrete in mind. He logs into the system with his username and password and he presses the button "start watching" getting immediately a video that he likes. Once the video is over a new video starts playing right the way. This video is adapted to the emotions shown during the visualisation of the previous video. This continues happening until the user decides to stop watching videos.

[1] the same name was posteriorly used as the domain name for the web service collecting the data

- Scenario 2: A user decides to watch entertaining videos making use of *Recomotions*. This time he has some specific video in mind that he watched some time ago and wants to watch again. He logs in with his username and password. He looks for it on the search bar and press play. Once the video is over a new video starts playing, adapted to the video the user just watched, his previous visualisation history and the emotions shown in the visualisation moment. The sequences of videos continue until the user decides to stop watching.

- Scenario 3: A user decides to watch entertaining videos making use of *Recomotions*. In this occasion, the user does not have something specific in mind but wants to choose from a list of options. He logs in with his username and password. He browses the videos available for the different topics and selects one. Once the video is over a new video starts playing, adapted to the video the user just watched, his previous visualisation history and the emotions shown in the visualisation moment. The sequences of videos continue until the user decides to stop watching.

- Scenario 4: A group of users decides to watch together entertaining videos using *Recomotions*. One of them logs in and uses the option of watching with more people and lets other users add the information about username and password. Once this is done they press the option "start playing" or browse/search a video they want to watch. The sequence of videos shown to them will be adapted to the emotions shown by all of the watchers.

It is important for the reader to be aware that the scenario 4 is presented here because it is an important case to have into account since it would be naive to assume that the users would never watch videos accompanied. However, this scenario, as stated in section 1.4 limitations, is left out of the study because of the complexity it would add. Therefore this case will be omitted in the rest of the report

In Fig. 4.1, the use case diagram is shown.

From this section, the following requirements are created:

- *The users are able to register, login and logout from the system*
- *The users are able to browse for videos in the catalogue*
- *The users are able to search other videos not present in the catalogue and the system should add them after the visualisation*

4.4 Adoption Theoretical Framework

This section presents the framework that was used to study the potential adoption of the proposed Recommender System. As it has been said before, the purpose of studying the adoption is to find out what factors are the most relevant for a user to choose to use the system, allowing the possibility of using these factors as design principles of the RS.

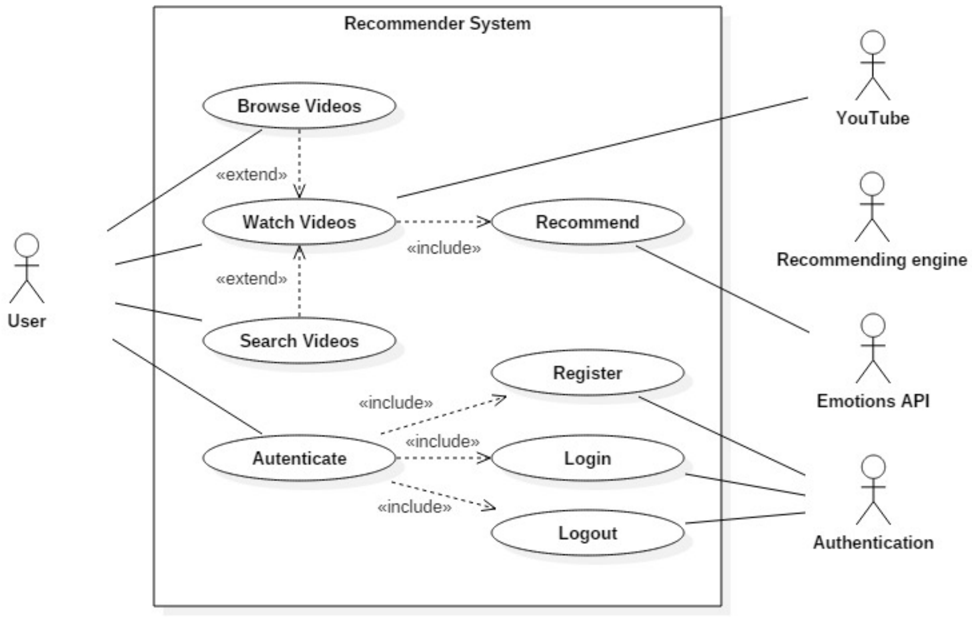

FIGURE 4.1: Use Case Diagram

For studying the adoption the Unified Theory of Acceptance and Use of Technology 2 [54] was used. This framework, presented in [54] is a modification of the Unified Theory of Acceptance and Use of Technology (UTAUT). The UTAUT is a framework presented by [56] in 2003 as a combination of different frameworks that had been used until the moment for studying ICT adoption. The UTAUT presents four constructs that directly affect the adoption intention of ICT. These constructs are performance expectancy, effort expectancy, social influence and facilitating conditions. It also presents four other constructs that act as moderators in the adoption process: age, gender, experience and voluntariness of use.

The named extension of UTAUT, UTAUT2, is presented in [54] with three additional constructs that are interesting for the current study of affective RS. The new constructs added by UTAUT2 are hedonic motivation, price value, and habit. The model is specially targeted to study users in their daily life. In the research where this framework is presented for the first time, it is tested by studying the adoption of mobile Internet. In [65] the authors also use the framework, this time to study the acceptance of phablets [2], and in [67] to study trust in Electronic Health Records system by the medical community. This framework has been applied to the field of RSs in [18] and [68].

In the former one ([18]), the authors use it to study the intention of adoption of a social RS. For this purpose, some constructs are added (related to the social aspect of an RS) that influence directly to the performance expectancy of the RS. As the proposed system does not exist, it is just an idea, the authors study the behavioural intention instead of the use behaviour. For the measurements, single-item tests are used, as they claim that there are

[2]A phablet is a computing device with a screen size between four-and-a-half and seven inches, measured diagonally. As the name implies, the device is essentially a tablet that also functions as a phone. However, the smaller size makes it easier for users to carry them around in pockets or small bags, as is customary with smartphones but less common with tablets [66]

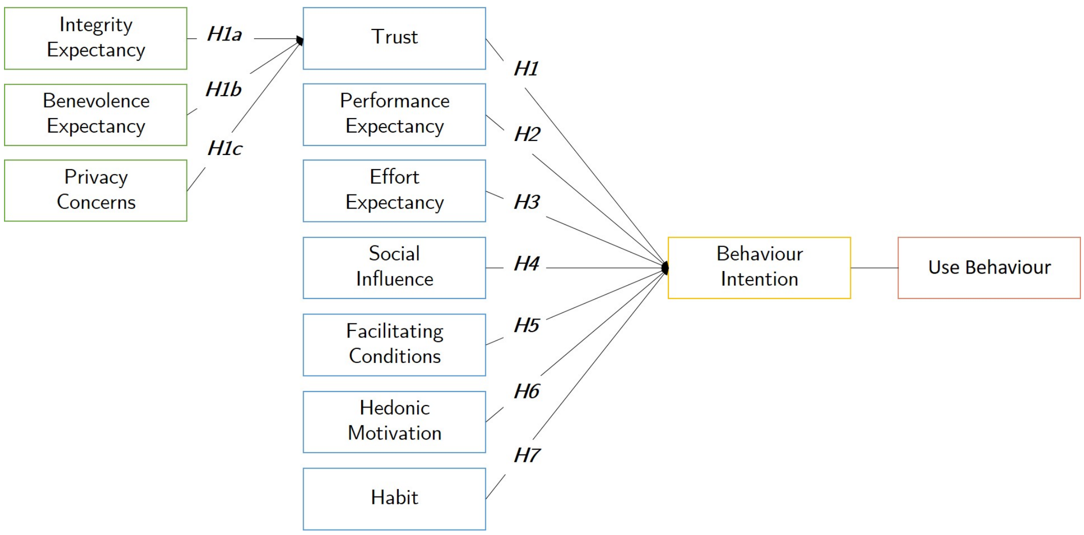

FIGURE 4.2: Framework for studying adoption

several authors that have demonstrated that there is no difference in the results between using single or multiple item methods slightly adapted from the original UTAUT [56] to fit the context. Some new measures were designed to evaluate the new constructs. As a part of their study, they present the validity of the model for social RSs, which let us extrapolate and assume that the model will also work for an Affective Recommender System.

It is difficult to find works about adoption in affective computing. The most relevant are [69], in which it is emphasised that the biggest issue to adopt affective technologies is the lack of trust in the new technologies. For this purpose, they use a trust based acceptance model and they study several aspects, such as what are the factors that condition the trust on affective computing and what does trust in affective computing mean.

As the RS that is being studied here has an important component of affective computing, some attention should be paid to it. Therefore, the framework used to study the acceptance will be the UTAUT2 with a modification: trust was included as a new construct that directly affects the behavioural intention, and three other constructs that are hypothesised to affect the trust were added. Another modification is that the Price Value, a construct present in the UTAUT2 model will be dropped, as done in [70]. The reason for this decision is that most of the recommender services available on-line tend to be for free, so studying the effect of the price in this kind of technology is not relevant.

The framework for studying the adoption is shown in figure 4.2

As it can be seen in figure 4.2 the Use Behaviour (UB) is studied through the direct relation with the Behavioural Intention (BI), since the proposed RS is just a concept and not a final idea, which means that there is no way to measure the actual use of it. As it was done in [18] the behavioural intention to use is used as a reliable predictor of the actual use behaviour.

In the framework presented in figure 4.2 there are seven constructs affecting directly to the Behavioural Intention. The first one, trust, is a supplementary construct, added to the model after the literature review on adoption and affective computing. Three extra constructs have been added to investigate what factors model the trust on an affective RS.

The four constructs below trust are from the original UTAUT and the last two are from its extension UTAUT2 and the last one is added particularly in this study. An explanation for these constructs can be found in the following paragraphs. As explained in section 5.1, the constructs were measured through a questionnaire, leading to the settlement of three new requirements for the data collection platform:

- *The data collection platform provides the participants in the experiment with a questionnaire*
- *The system stores all the answers of the questionnaire*
- *A subject should only be able to respond the questionnaire once*
- *The system assures that all the questions given in the questionnaire are responded before submission*

Trust:

Trust is defined as the confidence that a user presents when using a system.

The main idea of including trust was obtained from [69]. In this work, the authors state that, due to the Affective computing characteristics, trust is the main construct affecting its adoption. In [71] the authors also contribute to the idea that trust - and in this case, they talk about privacy - is one of the biggest factors conditioning the adoption of affective computing. Moreover, in [72], trust is considered one of the biggest challenges of affective computing.

In [69] the following definition for trust is proposed: "psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviour of another". The authors state that trust in Affective Computing depends on three factors:

1. the willingness to depend on proper functioning of the technology
2. the willingness to depend on proper technology behaviour on basis of a functioning emotion recognition.
3. the willingness to depend on proper data handling and privacy protection.

In this paper, they also defend that the trust on Affective Computer depends on a new construct that they call "self-reflexivity", which refers to the level a user is aware of his emotions. Such handling is important for confirming that the emotions detected by the affective system are right and therefore building the trust. However, this factor will be left out of this study because its aim is to use the findings as design principles and the self-reflexivity is something inherent to the user, not possible to be modified by the system.

There are also works that present the research of Trust in RSs (or similar technologies such as web personalization) [51, 57, 67, 73, 74]. In [73] and [51] Trust is incorporated as a construct of the UTAUT model to study behavioural intention for adopting an RS, and in both works, the authors conclude that Trust has an important role in the adoption of RS.

In [74] a framework for studying adoption in mobile banking is proposed; Trust is considered as a construct directly affecting the behavioural intention of adoption. The authors use three factors that condition the initial trust on the system: structural assurance, personal propensity to trust, and firm reputation. These constructs could be included into the proposed modification of UTAUT2 model as Trust modifiers, however, as it has been formerly said, the focus is to find design principles, so these constructs will not be considered.

In [57] the authors create an acceptance model based on Trust for studying the effect of personalization in mHealth acceptance. They present some factors regarding Trust that are similar to those presented in [69] as factors on which the Trust on affective computing depends. This is why the framework used to this study presents three constructs that affect the overall trust on the system:

- Integrity expectancy, defined as the degree the user thinks the system is capable of detecting the correct emotions and generating reliable recommendations
- Benevolence expectancy, defined as the degree the user thinks that the system cares about him and acts for his interests
- Privacy Concerns, defined as the worries related to the correct use of his data the user has when using the system

The hypotheses for the Trust construct are the following:

> *H1: Trusting the system has a positive impact on the intention of adoption.*
>
> *H1a: A positive integrity expectancy has a positive impact on the trust of the system.*
>
> *H1b: A positive benevolence expectancy has a positive impact on the trust of the system.*
>
> *H1c: Privacy concerns have a negative impact in trusting the system.*

Performance Expectancy:

This construct refers to the perceived improvement the user will get by adopting the innovation. In [75], the authors compare two music Recommender Systems. Is proved that the system providing better quality recommendations is the preferred one by the users. Consequently, considering this information and the UTAUT2 framework it can be hypothesized that:

> *H2: The performance expectancy has a positive effect on the intention of adoption of an Affective Recommender System*

Effort Expectancy:

Effort Expectancy is defined as the subjective difficulty the user expects in the process of adopting an innovation. The authors in [75] concluded that the lower the expectancy of the initial effort of using an RS, the more prone is a user to decide to adopt it. Because of this, and following the UTAUT2 framework the following hypothesis is generated:

> *H3: A low effort expectancy has a positive influence on the intention of adoption an Affective Recommender System.*

Social Influence:

This construct refers to the degree in which the individuals think that their important circle (family, friends, colleges...) is in favour of the use of the new technology. In concordance with the UTAUT2 framework it can be hypothesis:

> *H4: The social influence will have a positive impact on the intention to adopt the system.*

Facilitating Conditions:

The Facilitating Conditions refer to the perceived available resources and support to help the individual with the new product. For example, customer service or tutorials. The UTAUT2 framework exposes that the availability of facilitating conditions has a positive influence on the intention of an adoption of the technology. Therefore, it is hypothesised:

> *H5: The facilitating conditions will have a positive impact on the intention to adopt.*

Hedonic Motivation:

Hedonic Motivation talks about the personal enjoyment and pleasure obtained from using the technology. This is one of the most important constructs in a study([54]), in which it was found that Hedonic Motivation is more significant than Performance Expectancy. Since the studied system in this research has entertaining purposes, it is expected that this factor will be of high relevance:

> *H6: A positive hedonic motivation has a high impact on the adoption intention of an Affective Recommender system.*

Habit:

Habit is defined as the degree the consumer perceives that the completion of the task is automatic. In the UTAUT2 model, Habit is studied as having a direct effect on Behavioural Intention and on actual behaviour. In this case, the actual behaviour cannot be studied, therefore:

> *H7: Habit will have a positive impact on the adoption intention.*

4.5 Flow of the Data Collection Process

In this section, the description of the data collection process, focusing in the functionalities the platform should do is described textually. Following, the reasons for the presented experiment setup are described.

> When a subject enters into the website for the first time the *home* page is shown. This page shows a welcome message with a short explanation of the experiment and the resources needed to complete it (laptop with camera and Chrome or Firefox). Two buttons, one for registering and the other one for logging in, are shown. The first time the subject must press the *Register* button. This page allows the option of changing the language (the possible languages are English and Spanish).
>
> When the subject clicks *Register* he is redirected to a form that asks for a username and two passwords (original and confirmation). Once this information is given the subject clicks the *Sign Up* button and is redirected to a new form that asks for age, gender and country.
>
> After this information fulfilled the subject finds an explanation on what he will be doing and reminded that a laptop with camera and Chrome or Firefox is needed. The subject is also warned here that he will be recorded by the laptop camera and that the video will be sent to the emotions detection service to be analysed and to extract the emotions. Once this information is read the subject presses *Okay* and is redirected to a page explaining the first part of the experiment.
>
> The first part of the experiment has three steps that are repeated three times. After the subject reads about these steps and clicks *Okay* the first step is loaded. The subject is asked to check if the camera is working properly, by showing on the screen what the camera is capturing. Once the check is done he presses the confirmation button and is redirected to the second step. This step consists of watching a video that is selected randomly from the video catalogue. If the video does not suit the watcher at all, he can ask for a new one pressing the *New Video* button. When the video is over a new page is shown, with a short form asking for the subject's opinion about the video. When the form is submitted, it redirects to either the first step or, if it's the third video, to the second part of the experiment.
>
> In the second part, the subject answers the adoption questionnaire. Before the questionnaire, an explanation is given for a better understanding of the whole idea. The questions are presented in a list, and the answers are a selectable range scale from one to seven. All the questions have to be answered for being able to submit it.
>
> Once the questionnaire is fulfilled and sent, the subject has finished the experiment. The last page is shown thanking him for taking part and offering two options: *Log Out* or *Continue Watching videos*.
>
> If the option *continue watching videos* is selected - or if a subject that already took part in the experiment logs in the system through the *home* page, he is

redirected to the first step of the first part of the experiment: the visualisation part. He will be in the loop check camera - watch video - video feedback until he decides to stop.

An extension of the UML activity diagram has been used for detailing the user interface navigation. A detailed description of the model can be found in [76]. The diagram is shown in Fig. 4.3

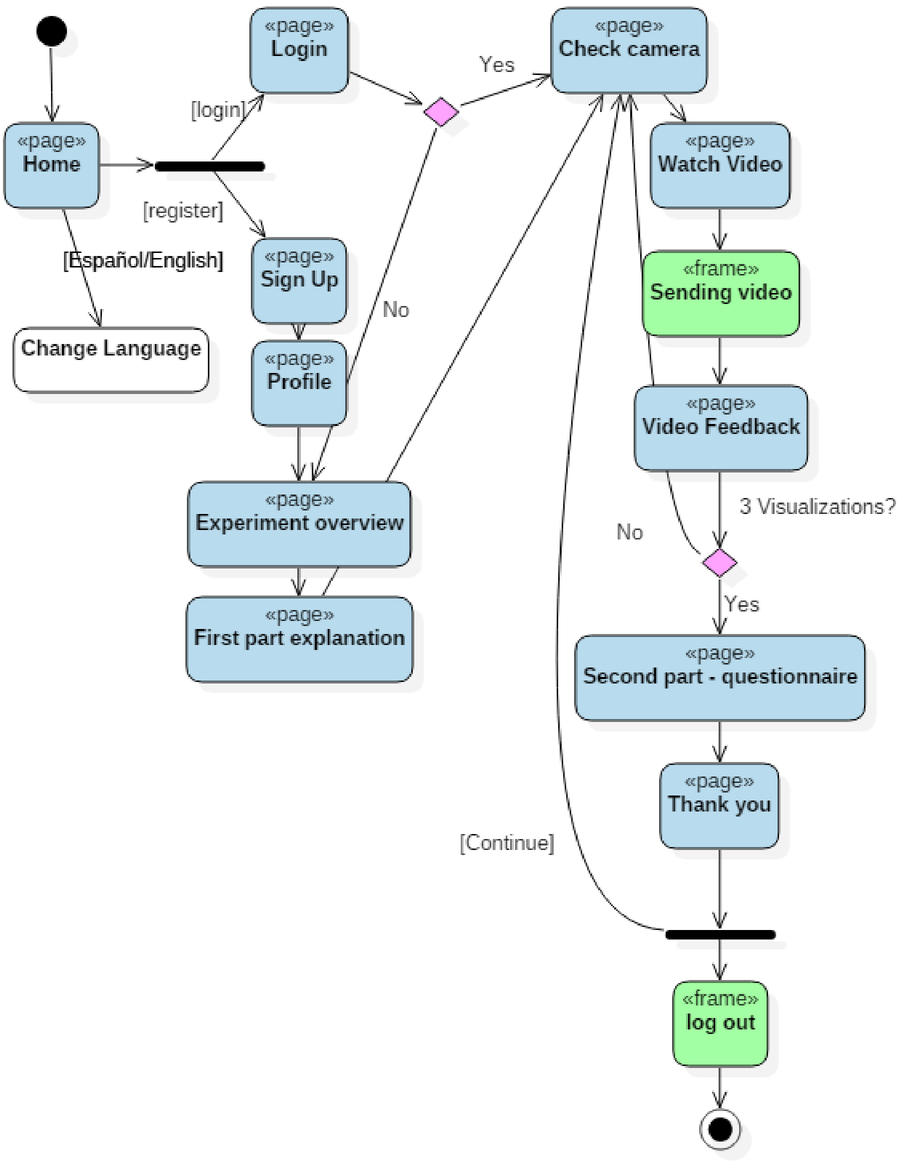

FIGURE 4.3: User Interface Activity Diagram

The reason to ask the subjects to **register** is that is necessary to be able to associate that all the actions a subject does during the experiment with the same person. For making this association is necessary that he is registered and logged in throughout the process. The identity of the user is not relevant, but his responses should be aggregated together.

Some personal information about the subject is asked for: **age, gender, and nationality**. These three characteristics are some of the typical ones in demographic Recommender Systems; it was chosen not to ask for more for not being too obtrusive [9]. As at the purpose of collecting the data is to generate a dataset for posterior analysis such information was considered could be used as extra input to a recommender engine.

From these affirmations, two new requirements can be defined:

- *Each user has one profile (and only one)*
- *The profile includes the following information: age, gender and country*
- *The subject are logged in conducting the experiment*
- *The profile information is completed by all users*

The **camera check step** was is necessary since, as explained later, is the tool used to capture the emotions. It was placed on a different page than the one showing the video with the aim of avoiding the distraction of the subject by seeing his image on the screen. This step is repeated before each video as a safety check, for the case that, for example, the subject moved and he is placed out of the viewing angle of the camera. A new requirement is established:

- *The system should ask for confirmation of the user of the proper functioning of the camera*

The videos the user watches are selected randomly from the catalogue to ensure that every category of videos gets visualisations. If the catalogue of videos is big and the user is required to watch only three videos, the chances of getting the same video twice are low, however, it could be checked by the system. The reason to allow the user to skip the videos is to also obtain information about that situation, in order to be able to identify the users' reaction when they do not want to watch the video. It can be stated:

- The videos are randomly selected for presenting them to the subject
- The same subject do not get the same video more than once
- The system allows the user to skip the assigned video

The questions asked as a feedback on the watched video is intended to be used posteriorly for trying recommender options and proving the feasibility of creating recommendations through the detection of emotions.

The first question is the typical binary question in which the users judge by "yes" or "no" if they liked the consumed multimedia. There is a tendency in the community of RS to increase the use of binary feedback because it is less affected by contextual subjectivity.

The second question has two aims. The first one is to confirm that if a user liked a video, then he would like the whole category where the video is contained or the opposite, that if he rated a video as he did not like it, but he normally likes videos of this kind. The second purpose is to avoid the bias effect caused by what the user thinks is an appropriate rating for the watched content, and what his actual rate would be. There is a tendency in Internet users to rate according to what is thought to be socially correct: a video about world issues could possibly be rated higher than a video about TV comedies or funny moments because the former is considered more *correct*. The question "Would you watch something similar again?" tries to diminish this effect by indirectly repeating the first question.

The users are asked to watch three videos to keep the experiment not too long. The maximum length of each video is 4 minutes, so the maximum time spent watching videos would be 12 minutes. Adding the time of understanding the tasks, waiting for the videos to be sent and answering the questionnaire, the time of carrying out the experiment is approximately 20 minutes.

In order to assure that the length of the experiment is as designed, the following requirements are settled:

- The 80% of the subjects are able to understand what has to be done by reading once the explanations
- The waiting time between one page and other should not be greater than 5 seconds under normal conditions

There are two reasons for having the **questionnaire after the visualization**. The first one is that the subjects get an idea of how would the system be, in the sense of the camera recording the emotions. The second reason is that the questionnaire is the tedious part of the experiment, and then the subject would be more prone to finish it if he has already invested some time in doing the first part; if he had received the questionnaire in the first place it would be more probable that he would decide not to complete it. Therefore, it is established that:

- *The system assures that after 3 complete visualisations the user receives questionnaire*

As the intention is to collect the emotional data, it is necessary that the system stores information about the visualisation, storing the what video the user watched, the emotions detected and the feedback provided. It is stabilised:

- *The system stores each visualisation, relating the video watched with the subject that watched*
- *The system obtains and stores the emotions shown while the subject is watching the video*

Moreover, in order to allow the possibility to apply the framework described in [12] the recorded video should be longer than the visualisation time. As stated in 3.2, the framework defines three stages in the consumption chain of an item: entry, consumption and exit stage. The interaction of the user with the system before and after watching the video should also be recorded to allow working with the three stages.

- *The system records the user before, during and after the visualisation of the video.*
- *The system stores information about the start and end of the recording and of the playback.*

It was chosen to have the website available in two languages for reaching more people, and for avoiding that the lack of English knowledge was a barrier to Spanish speaking people to take part in the experiment. A non-functional requirement is defined as:

- *The site is in English and in Spanish*

Other requirements can be added to assure the possibility of the realisation of the experiment:

- The site has contact information
- The site has an about page
- The admin interface should only be accessed by users with special permissions
- The site should be accessible from any device with a camera and from the relevant browsers
- The platform must be available 99% of the time

4.6 Emotion Detection API Selection

This section presents different options for acquiring the emotions through facial recognition. As it was explained earlier in this thesis, this aspect of the project will rely on the technology available on the market for this purpose. With the technological advances, the amount of companies doing this kind of software is increasing. Seven options were found to be interesting for the project; they were compared and finally the most appropriate was chosen. The seven options were:

- **Nvisio** [28]: This solution, developed by the company with the same name, runs on IBM SmartCloud Enterprise or on IBM System x Servers. It uses a deep learning proprietary solution for detecting the emotions. According to the website, the technology can track emotions in real time by tracking hundreds of muscle movements using a webcam. It can recognise emotions of one person or more. However, the available information on the web page is limited and for getting more information is necessary to send a contact request. This was done, but no answer was received, and therefore this technology was discarded.

- **Affectiva Emotion Recognition Software** [27]: Affectiva was the first company to develop this kind of software that analyses facial expression to detect emotions. It has a database of pictures showing emotions with people from 75 different countries that has been collected for more than seven years [77]. It was born as a spin-off from the Affective Computing group in the MIT. It detects emotions from an image or from a video. It detects seven emotions: anger, contempt, disgust, fear, joy, sadness and surprise. It can also measure engagement and valence (how positive or negative the expression is). It provides as well metrics about physical appearance such as age, gender and ethnicity. It can be used as a Restful API service or by integrating it with the applications using the provided software development kit (SDK).

- **EmoVu** [78]: This software, by Eyeris, has also a deep learning algorithm trained with a big data set with annotated data from people of different ethnicities, age groups, genders etc. It is developed with a target in embedded systems, so the software is light weighted. It also allows high customization and it is improving constantly as it gets more data to train the system. It works for a person or for a group of people. When the API is used, the response it a JSON file with information about the position of the face in the image, the age group of the person, the gender and the emotions in the form of anger, disgust, fear, joy, neutral, sadness and surprise. It also provides some other metrics such as engagement and degree of attention or valence.

- **Kairos** [79]: This software allows to use it through an API and it provides a lot of information about the photo or the video: six emotions, age, gender, attention and sentiment. It is a paying service but it allows a freemium subscription, where you can use a certain amount of API calls and hours of video analysis a month. This software provides a lot of information about the emotions shown in the uploaded media: emotions, age and gender, ethnicity, emotional depth; it also allows face recognition and face detection. The service was tested resulting that the processing time was too long to do any real time application.

- **CrowdEmotions** [80]: This is a Beta service for analyzing the emotions in videos. It is interesting for three reasons, it integrates the software for capturing the video, that is browser based, using HTML5 and it has examples using the API in different programming languages available open source in GitHub and it is a free service. However, at the moment of the investigation, it was a *beta* service, and the documentation on the API was difficult to understand

- **Google Vision API**[29]: the service provided by Google was also considered, but it was soon discarded for not allowing the analysis of videos.

- **Microsoft Cognitive Services Emotion API** [26]: This service is provided as a RestFull API and returns eight emotions: anger, contempt, disgust, fear, happiness, neutral, sadness and surprise. It allows analyzing video and photos with one or more people (together with the emotions it returns the position of the face showing those emotions).

As there was a high availability of RESTful APIs for the emotion detection it was decided to use the service in this form to make the implementation of data collection platform simpler and decrease the necessary computing capacity. This decision was taken being aware that the latency of the network would increase the necessary time to analyse the video and obtain the emotions, a cost that was determined as acceptable for the proof of concept.

It was determined that the API service had to be able to detect the emotions from a video. The emotions over the visualisation of the content shown by the to-be-designed RS vary along its playback. It was an option to use an API that could only analyse images by taking pictures of the user while watching a video instead of recording him, however, this option was rejected with means of obtaining as much emotional data as possible.

From the presented APIs it was chosen to use the Microsoft Cognitive Services API. This is a Restful API service that allows detecting emotions from a video. It returns a JSON file with information only about the emotions and about the position of the faces, making the received data simpler. Out of the presented solutions, the Microsoft one was the simplest one in terms of the input parameters and the response. The response provided satisfied the necessity of the data collection. In addition, it could be complemented by other APIs from the cognitive services group, which allow other functionality such as face recognition, or age or gender detection, in the case of necessity in future steps. Another reason for choosing it was how simple it was to use it and that the documentation provided with examples and an API reference which allowed to try it interactively, modifying different parameters and receiving real responses. Moreover, it allowed the option of *real time* retrieval by returning the emotions detected per frame of video.

4.7 Requirements

From the use case, a set of functional and non-funtional requirements for the data collection platform can be inferred. These requirements are presented below.

The Functional Requirements describe an action the system should perform, whereas non-functional requirements express a verifiable property the system should fulfil [81, p. 69].

TABLE 4.1: Functional Requirements

Functional Requirements				
Requirement Code	Category	Description	Rationale	Priority
FR01	Profile	The users are able to register, login and logout from the system	4.3	MUST
FR02	Profile	The profile includes the following information: age, gender and country	4.5	MUST
FR03	Videos	The platform contains a catalogue of YouTube videos selected under a certain criterion	4.2	MUST
FR04	Videos	The system embeds YouTube videos	4.2	MUST
FR05	Videos	The videos are randomly selected for presenting them to the subject	4.5	MUST
FR06	Videos	The same subject do not get the same video more than once	4.5	COULD
FR07	Videos	The users are be able to browse for videos in the catalogue	4.3	COULD
FR08	Videos	The users are able to search other videos not present in the catalogue and the system should add them after the visualisation	4.3	COULD
FR09	Visualization	The system stores each visualization, relating the video watched with the subject that watched	4.5	MUST
FR10	Visualization	The system obtains and stores the emotions shown while the subject is watching the video	4.5	MUST
FR11	Visualization	The system records the user before, during and after the visualization of the video.	4.5	MUST

TABLE 4.2: Functional Requirements (Cont.)

Functional Requirements (Cont.)				
Requirement Code	Category	Description	Rationale	Priority
FR12	Visualization	The system stores information about the start and end of the recording and of the playback.	4.5	MUST
FR13	Visualization	The system allows the user to skip the assigned video	4.5	MUST
FR14	Questionnaire	The data collection platform provides the participants in the experiment with a questionnaire	4.4	MUST
FR15	Questionnaire	The system stores all the answers of the questionnaire	4.4	MUST
FR16	Questionnaire	A subject is able to respond the questionnaire only once	4.4	MUST
FR17	Questionnaire	The system should show the questionnaire questions in random order	5.1	MUST
FR18	Experiment Flow	The system should ask for confirmation of the user of the proper functioning of the camera	4.3	MUST
FR19	Experiment Flow	The system assures that after 3 complete visualizations the user receives questionnaire	4.3	MUST
FR20	Experiment Flow	The site has contact information	4.5	MUST
FR21	Experiment Flow	The site has an about page	4.5	COULD
FR22	Experiment Flow	The system detects the emotions in real time	4.5	COULD

Table 4.3: Non Functional Requirements

Non Functional Requirements				
Requirement Code	Category	Description	Rationale	Priority
NFR01	Usability	The 80\% of the subjects are able to understand what has to be done by reading once the explanations	4.5	MUST
NFR02	Usability	The design is consistent over all the web page	4.2	MUST
NFR03	Usability	95\% of the subject do not find difficulty navigating the page	4.2	MUST
NFR04	Accesibility	The site is in English and in Spanish	4.5	MUST
NFR05	Accesibility	The site should be accessible from any device with a camera and from the relevant browser	4.5	MUST
NFR06	Performance	The waiting time between one page and other should not be greater than 5 seconds under normal conditions	4.5	MUST
NFR07	Availability	The platform must be available 99\% of the time	4.5	MUST
NFR08	Security	The users' personal information and information about the visualization should be secured	4.2	MUST
NFR09	Security	The admin interface should only be accessed by users with special permissions	4.5	MUST
NFR10	Security	The http connection is secured with SSL	5.4	MUST
NFR11	Profile	Each user has one profile (and only one)	4.5	MUST
NFR12	Experiment Flow	The subject are logged in conducting the experiment	4.3	MUST
NFR13	Experiment Flow	The profile information is completed by all users	4.3	MUST
NFR14	Experiment Flow	The system assures that all the questions given in the questionnaire are responded before submission	4.4	MUST

Chapter 5

Data Collection: Design Considerations

This chapter presents the design and implementation of the data collection platform. In a first instance, the design of the questionnaire which is part of the data collection is presented. Afterwards, the design of the platform, based on the requirements concluded in chapter 4, is described using the Unified Modelling Language (UML) [64] diagrams that are relevant to the case. A description of how the video catalogue was selected is given, and finally, the implementation of the system is explained. The implemented proof of concept can be accessed in *wwww.recomotions.me*

5.1 Questionnaire Design

The questionnaire was given to the subjects once they had already completed the first part of the experiment. Therefore, it was important to make them understand that their previous interaction with the platform was part of the data collection, not the actual system that is being studied. A short explanation was given before the questions to make this concept clear and to have all the respondents under the same level of knowledge about the system. The text is presented below:

> In my Master's Thesis, I am studying Affective Recommender Systems (ARS). These are systems which can recommend an activity based on the emotions of the user that are automatically acquired.
>
> The idea is to use the ARS in an online application (which I named Recomotions) in order to recommend entertaining videos. While the user watches a video, the application detects the emotions from the facial expression using the camera, and when the video is over it starts playing a new one selected by the system. These recommendations are personalised for each user and improve over time, each time the user watches a video, the system will obtain more information about him or her and eventually it will create better recommendations. Below

> you will find a questionnaire that will help me to evaluate the acceptance of this application. When answering, keep in mind that what you did in the first part of the experiment was similar but not the same as what my application Recomotions would do. In Recomotions the videos would be personalised and played without pause, that is to say, it wouldn't ask for your opinion after the visualisation.

It was chosen to use an informal style for the explanations because the aim was that any subject of the experiment could understand the concept explained. As the survey was targeted to any Internet user it was expected to have participants from different age groups and educational backgrounds, leading to the use of simple language and not technical terms.

After the explanation, the questionnaire was presented. The questions appeared in random order for each user to reduce the effect of common method variance(CMV)[70]. A seven-point Likert scale was used for the measurements. The values were 1 = "strongly agree" and 7 = "strongly disagree".

For designing the questionnaire, the model presented in 4.4 was used as the base. All the measures used were adapted from other studies, benefiting that they had already been validated by other researchers.

Each construct was evaluated with a multi-measure approach presented below. Most of them were adapted from the ones presented in [54], [68] was also used to refine the questions, since its authors study a product that is not yet commercialise. The questions in [82] were also used as inspiration when modifying the questions from the original UTAUT2. The behavioural intention questions were determined by adapting some of the questions of the ResQue framework [83]. The questions regarding trust were set by slightly modifying the questions used in [57] to fit this context.

A preliminary set composed of 33 questions was created. It can be found the Appendix A. The questions in the set were tested with three potential respondents of the survey and had to be modified because they were not clear enough. The modifications were done for expressing the same question in a less technical language. The number of questions was reduced to avoid excessive repetition and to reduce the responding time of the questionnaire.

Both the questionnaire and the introductory explanation were translated to Spanish. The respondents could choose to answer in any of the two languages. As said in the methodology chapter (2), the choice to translate the questions to Spanish was to reach a higher number of responses. In order to assure that the questions had the same meaning in both languages, they were translated and afterwards a professional translator was consulted. Small modifications were done after her feedback. The Spanish version of the introductory text and questionnaire can be found in the Appendix A

The English version of the questions used is presented in table 5.1

TABLE 5.1: Questionnaire

PE	**Performance Expectancy**
PE1	I think that using the Recomotions application would increase my chances of discovering videos that entertain me
PE2	I think that using the Recomotions application would help me to discover videos that entertain me faster
EE	**Effort Expectancy**
EE1	I think I would easily learn how to use the Recomotions application
EE2	I think that the Recomotions application would be easy to use
SI	**Social influence**
SI1	I think that people who are important to me would recommend me to use the Recomotions application (e.g.: my family, my friends. . .)
SI2	I think that people who influence my behaviour would recommend me to use the Recomotions application (e.g.: famous people, people I admire. . .)
SI3	I think that my friends would find the Recomotions application attractive
FC	**Facilitating conditions**
FC1	I have the resources needed to use the Recomotions application
FC2	I have the knowledge needed to use the Recomotions application
FC3	I think that I could get help from others when using the Recomotions application if I needed to
HM	**Hedonic Motivation**
HM1	I think using the Recomotions application would be fun
HM2	I think using the Recomotions application would be enjoyable
HM3	I think using the Recomotions application would be entertaining
H	**Habit**
H1	I think that using the Recomotions application could become a habit to me
H2	I think that using the Recomotions application could become natural to me
T	**Trust**
T1	I think that I would trust the Recomotions application
IE	**Integrity Expectancy**
IE1	I think that the Recomotions application could properly detect my emotions
IE2	I think that the Recomotions application would be able to recommend videos I like from the detected emotions
BE	**Benevolence Expectancy**
BE1	I think that the Recomotions application would use my emotions in my best interest
BE2	I think that the Recomotions application would use my emotions only for generating recommendations
PC	**Privacy Concerns**
PC1	I think that using the Recomotions application would lead me to lose control over the privacy of my emotions
PC2	I think that others might take control of my personal information if I used the Recomotions application
UI	**Use Intention**
UI1	If the Recomotions application existed I would use it to discover entertaining videos
UI2	If the Recomotions application existed I would try to use it in my daily life
UI3	If the Recomotions application existed I would recommend it to others

5.2 Design of the Platform

This section presents the design of the data collection platform. It will be done by representing different concepts of the platform through diagrams.

As it was stated in chapter 4, the data collection was done through a web application - the reasons for this are also presented in the same section. Figure 5.1 shows a high-level architecture diagram that will help to get an overall view of how is the system that is being designed.

As it can be seen in the figure there are five main elements involved in the data collection process:

1. The subject participating in the data collection
2. The Web Server
3. The Database
4. Microsoft Cognitive Services Emotion API
5. YouTube

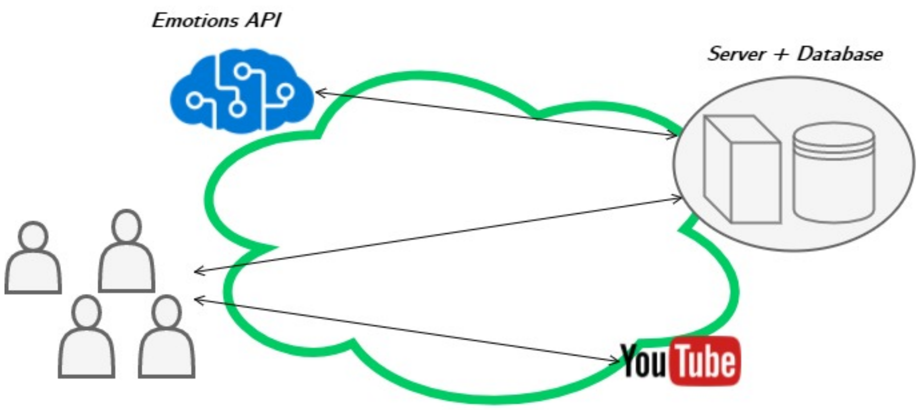

Figure 5.1: High Level Architecture

These elements are able to interact with each other - if needed - through Internet, except for the Web Server and the Database that are hosted on the same hardware for making the development of the platform simpler and because due to the simplicity of the platform and that it is not a service targeted for a big mass, there is no need for a different deployment option.

The users taking part in the experiment accesses it through the browser of their devices, establishing a connection between the device and the server. As it is a web application, most of the functionality are performed on the server. The exception is recording the video, which is done by the user's device to be sent to the server. The selection of the video to watch also takes place on the server; however, the call to the YouTube platform for retrieving the video

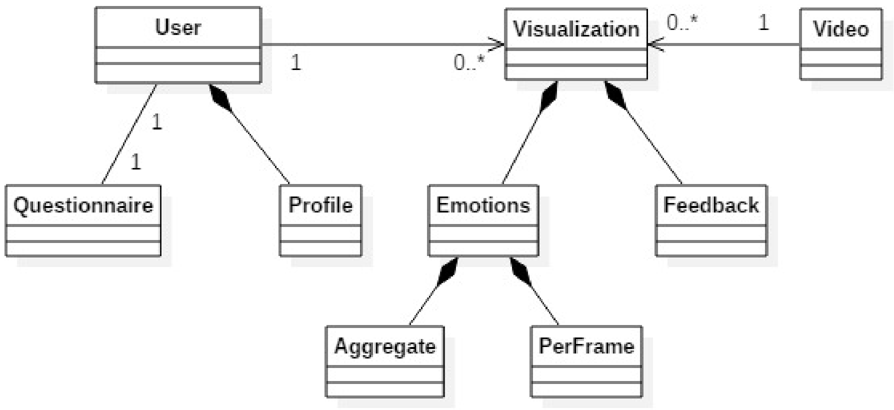

FIGURE 5.2: Logical Class Diagram

is done by the user side. To properly keep a record of a user's interaction it is necessary to set up sessions for each of the connections.

The Emotions API is a Restful like API which is accessed through HTTP requests, as explained in section 4.6. This element interacts with the server, which sends the requests to analyse videos. The videos to be analysed are sent to the server from the web browser.

Figure 5.2 represents the logical classes of the system. As it can be seen, the users are extended by the profile and by a counter of the number of videos that the user has watched. The profile allows storing the information related to the user described in requirements RF01 and RF02. The counter contributes to fulfilling requirement FR19 by keeping track of a number of videos that a user has watched, that is used to calculate when the questionnaire should be presented. The Questionnaire answers represent the set of 25 answers to each one of the questions of the questionnaire. The relation in the three cases is one to one. One user can only be associated to a single profile, video count and set of answers for the questionnaire.

The video class is the representation of the elements present in the video catalogue, characterised with the ID used by YouTube to identify all the videos present in the platform.

As it is shown in Fig. 5.2 a user can watch many videos and a video can be watched by many users. However, each visualisation element has only one associated video user, as well as information about when the visualisation happened. The visualisations are extended by the emotions class and by the feedback. The emotions can be of two classes: aggregate or per frame (more detail in section 5.4.1.3). The feedback contains information about whether the users liked the video and if they would watch something similar.

The sequence diagram of the experiment is presented in Fig. 5.3.

The sequence diagram shows the different elements of the system and the actions happening when a user takes part of an experiment. There is an authentication section in the server managing the registration of the users and the storage of their information in the database, as well the acquisition of the necessary user's information for completing the profile.

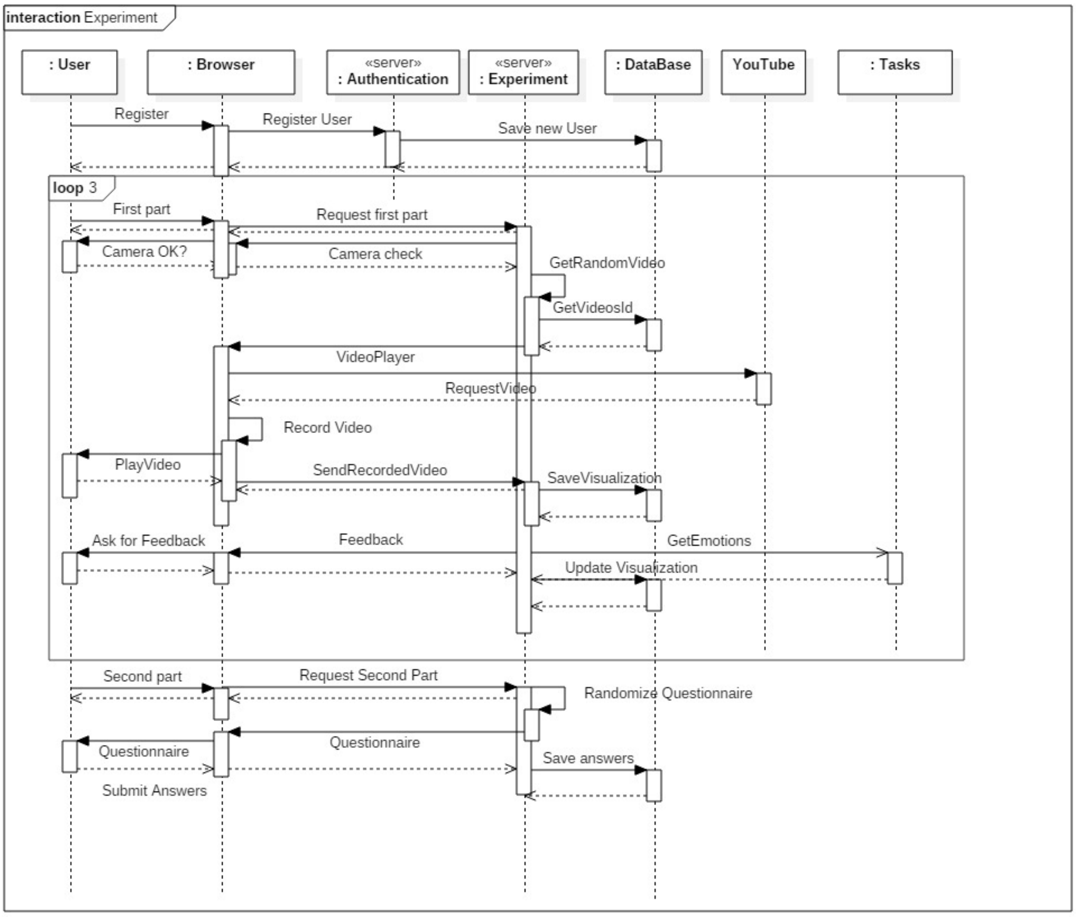

FIGURE 5.3: Sequence Diagram of the Experiment

There is another part in the server handling the experiment, which takes over once the registration has been completed successfully. This instance in the server takes care of everything except the recording of the video that, as said earlier, is managed on the client side. Once a video is recorded and sent to the server it must be sent to the emotions API so it can be analysed. It can be seen in the sequence diagram that this is taken care of in an entity named *:Tasks*. The message sent from the experiment to the tasks is asynchronous, in order to make the waiting time of the user as low as possible, avoiding the delay caused by calling the API and waiting for the emotions to be analysed. A more detailed description of the process for calling the API can be found in 5.4.1.3.

5.2.1 User Interface Design

The user interface must be consistent throughout all the page, as stated in requirement NFR02. For that the distribution of the browser window shown in Fig. 5.4 is used. This way all the pages in the test have the same components and the specific content will be in the container. The header presents the option to change languages and a link to the about section, fulfilling requirements NFR04 and NF21. It has also a link to the home page, following the normal practice of most of the websites.

It was chosen to have a background picture to make it more attractive for the participant in the experiment. A picture with warm and soft colours was selected, to give calm to the user and not distract him from the realisation of the experiment.

The container should be the focus of the user, as there the relevant content is shown, therefore it is white and outstanding from the rest of the site. The footer displays the contact information that the subjects could use if needed and the copyright information of the used template.

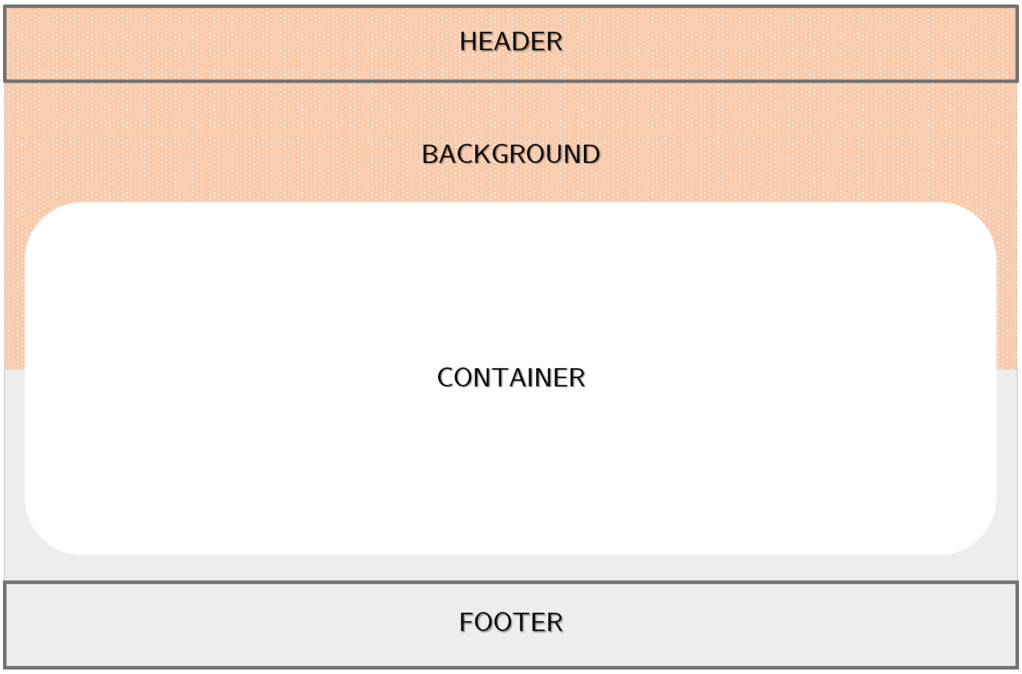

FIGURE 5.4: Interface Design

5.3 Video Catalogue

This subsection explains how the videos played during the experiment were selected. The reasons to create a specific catalogue for the project are listed below:

- There is a huge variety of videos in YouTube. The purpose of the studied RS is to recommend entertaining videos, however, the videos available range in categories such as *education*, *science & technology*, *news & politics* or *how-to & style* [84] - these, and more, are categories given by YouTube[1]. From all the variety, only the videos which intention is to entertain the watcher are relevant to the project.
- According to *Statista* [85] in July 2015 there were 400 hours of video uploaded every minute to the platform. This means that the number of videos there, even only those with entertaining purposes, is extremely big. Trying to work with all the available

[1] YouTube sets categories that assigns automatically to the channels, the video uploaders can choose to what category the video belongs to

> videos would turn the studied recommender problem into something different from the original intention: a recommending problem where the number of items is very big and continuously growing. Nevertheless, the purpose of the thesis is to create recommendations by using the emotive response of the users while watching a video so using a small set of videos would serve the purpose and make the task feasible.

As in the early stage of collecting the data the way to generate the recommendations was still unknown and it was decided to select the videos under some manual classification. The reason to do this was to be able to use this classification, if needed, as a simple way to implement a content-based recommending algorithm.

For creating these categories two different approaches were used. This was necessary because the first one failed, resulting in a selection of videos that were not good for the purpose. The two approaches are explained below.

First Video Selection Approach:

As mentioned, the intention was to create categories for later using them for a possible content-based recommending part. In this first approach, the creation of these categories was done in two different ways.

The first one was to distribute a very short questionnaire to family and friends asking how they would categorise the *making you laugh* videos they watch online. It was an open question, allowing the respondent to answer with as many categories as wished. Twenty-one answers were obtained. The obtained results can be seen in the appendix B. Some of the suggested categories were general while others could be used and subcategories of the general ones.

The second way to get the categories was following a similar approach as in [86] for obtaining a list of the most searched queries in YouTube. The approach consisted in writing the term *funny* in the search bar provided by YouTube and use the suggestions made by the platform for completing the query as a category, and the second level suggestions as a subcategory. The search queries obtained by this method can be found in the appendix B.

Once the two sets of categories were obtained some of them were merged - as they were obtained through different methods some of them were too similar for being different categories. The categories obtained from the survey were classified in *high relevance*, *medium relevance* and *low relevance* - depending on the amount of times they had been given as an answer. The relevance tag conditioned the number of videos retrieved for each of them (twenty, ten or five, respectively). The categories obtained by the second approach were all tagged with medium relevance. Once this was done, the YouTube Data API (v3)[87] was used to do a search of each category (using the category as query set).

The YouTube Data API is a Restful API. The method used for retrieving the videos was the *search:list* that allows sending a query set, with some other parameters, and returns the first x results for this search. The options used to filter the retrieved videos were:

- *part: snippet* - this parameter allows to choose the amount of metadata about the video the response provides. Out of the two options, the chosen ones return more information.
- *maxResults* - Parameter controlling the number of results the API returned. The parameter was varied between 20, 10 and 5 depending on the the classification as a relevant category, normal category or subcategory, respectively.
- *order: relevance* - defines the ordering criteria for the retrieve videos. Out of the three possibilities (relevance, number of views and number of likes) the default one -relevance- was found to be the most suitable to the task.
- *regionCode: ES* - this parameter makes the response to show the results for a specific country. Spain was chosen as this country because it was expected that most of the respondents to the experiment were from this country, as explained in the methodology chapter 2.
- *safeSearch:moderate* - this parameter makes the API return results that are filtered regarding the appropriateness of their content. At the beginning, this parameter was left to the default value (not filtering). However, inappropriate results were obtained, so it was modified to moderate.
- *type:video* - for making the search return only videos, and no playlists or channels.
- *videoCategoryId:24* - as said before, YouTube sets some categories for the content. It was decided to use the videos that were under the category entertainment, which was the id 24.
- *videoDuration:short* - short videos in YouTube are under four minutes. It was chosen to use these videos to assure that is was feasible for the subjects to take part in the experiment.
- *VideoEmbeddable:true* - for assuring that the videos could be embedded to the web page where the experiment was held.

After the API was called, those categories that had less than 1000000 total results were dropped out. However, once the final categories were used to retrieve the corresponding videos it was found that this method was not appropriate. Many video uploaders use *Search Engine Optimisation(SEO)* to position their videos in the top positions in YouTube search, and YouTube presents them as the most relevant for the given queries. But, after manual revision by watching a selection of them it was determined that the content was not entertaining and that the videos were of low quality (content wise), moreover, the metrics available in YouTube to determine the general opinion about the videos - View count, Likes and Dislikes - were very different in the resulting videos than in the videos that YouTube classifies as popular.

If the idea of the RS is to find videos the watchers would like, it is a bad practice to select a catalogue with videos that most of the people would not like, it would make almost impossible the task of recommending content the users like. Hence, it was decided to follow a different approach for selecting the videos.

Second Video Selection Approach:

For taking the second approach a more in-depth research on the dynamics of the platform was done.

YouTube has three different kinds of elements: videos, playlists and channels. Each playlist and each channel belongs to a user and are aggregations of videos (the playlists) or of videos and playlists (the channels). It is the normal practice that the channels have a specific topic, they are used by video uploaders to share their content. A channel allows other users to subscribe to it, making them get notifications when a new video is uploaded.

This new approach was focused on getting popular videos, using the metrics provided by the platform to determine the popularity of the videos. The incentive to use the popularity approach was the following quote by Xavier Amatriain in a master class in RS "If you don't know what to recommend, go for the popular items, it's very difficult to beat the popularity" [23].

For getting popular videos at the same time as videos for different categories it was decided to find popular channels that uploaded content about the chosen categories. It was done this way because it was easy to find channels of different kinds and because the measure of subscribers to a channel is more relevant than the number of views or likes to a video: the user must consciously decide to subscribe, and this happens habitually after visualizing some videos and deciding that he likes them, whereas a view just counts how many people clicked on the video, not if they watched it.

The channels were found by searching by topic in the YouTube search or by *best YouTube channels about –topic–* in a Google Search. The choice of the channels had some restrictions; they had to have over one million subscribers, playlists aggregating the videos and are of a specific subject or have playlists aggregating videos of a specific subject.

The specific channels were complemented with the auto-generated YouTube channel *Popular in YouTube* [88]. This channel has the most popular videos on YouTube at the moment in different playlists, aggregated by topic. Some of the playlists were used as categories. The final categories are presented in table 5.2. In appendix B the channels and playlists per categories are summarized in tables B.3 and B.4

The choice of the number of categories was done by some simplified probabilistic calculations. As the reason to do categories is to possibly use them as a way of implementing the recommendations it was important to assure that there was a reasonable number of users that had watched a category in common (since achieving this for the videos was an unrealistic purpose). The normal practice in the literature is to discard the items or users that have less than ten ratings in common [5, p. 818], [89].

Table 5.2: Categories

Adventures
Animals
Animations
Babies
Best of TV shows
Catch Up on Late Night
Comedy
Everyday life jokes
Fails
Game Overload
Interesting facts
Kids
Leading Women
Learn Something New
Look Good & Feel Good
Magic
Parodies
Pranks
Shared and Liked
Sports Highlights and Great Moments
Stand-up comedy
Stories That'll Restore Your Faith in Humanity
That's a Good Question

If a subject watches three videos, the probability of watching a video of category *c* in one of the three watched videos is equal to:

$$P_c = 1 - P_{noc1} \cap P_{noc2} \cap P_{noc3} = 1 - (\frac{n_{categories} - 1}{n_{categories}})^3$$

Substituting the number of categories by 23 the following probability is obtained:

$$P_c = 1 - (\frac{22}{23})^3 \approx 0,125$$

This means that by having around 70 video watchers, that watch at least the three mandatory videos, each category will receive a visualisation of 12,5 % of the users, which is approximately 9 viewers per category. A number considered to be good enough for this work. However, no support in the literature was found to confirm that the chosen number of 23 videos was an appropriate one.

The reason to limit the chosen channels to those that had playlists was to be able to use the YouTube Data API (v3)[87]. With the method Playlist:list returns a response containing the videos of a playlist (it returns maximum 50). The intention was to obtain approximately four hundred videos in total, therefore the best approach was to automatize the retrieval of the videos and the upload of the necessary data to the database.

With the purpose of the automation, a python script was written. This script can be found in the attached files. The script takes a dictionary containing the category names as *keys* and a list of the playlists as values.

For each playlist of each category, it calls two methods of the YouTube Data API: playlist:list and video:list. The first one returns 50 videos of the given playlist, the second one gets metadata for those videos. This information is saved in a dictionary object that contains the category name and a list of the videos belonging to this category with their metadata (also in dictionary form).

Afterwards, the list of videos was filtered using two criteria. For that, the retrieved metadata is used. The first criterion is that the videos should be four minutes long or shorter. The second one is that they should have at least 300.000 visualisations. After this filtering, the list of videos is ordered according to a dislikes/likes rating. The videos with the smallest ratio go in the beginning of the list. Different options for filtering and ordering were considered (e.g. ratio likes/views). As the dynamics of YouTube are very difficult to understand, for example, a video can have a low rating of likes over visualisations because users only rate a video once, but they can visualise it many times; very simple metrics were chosen. The number of visualisations was set to a low limit, but it would filter out videos that had not receive any views [2]. The ratio *likes/dislikes* was chosen because both actions can only be done once by a user, the comparison of the two parameters is more accurate for all of the videos than other criteria using the number of visualisations.

[2] It could be that they had not been viewed because they had just been uploaded, but for creating the catalogue this fact is not relevant

Finally, the top 30 videos for each category were chosen and the *MySQLdb* library was used to connect Python with the database and insert the videos to the database. The database, as it was presented before, stores the id, the title, the channel and the category of each video.

After the process, the number of selected videos was **435**.

5.4 Implementation

This section describes the implementation of the prototype for the data collection.

5.4.1 Used Technologies

To begin with, an overview of the selected technologies that have been used to implement the proof of concept is presented.

5.4.1.1 Programming Languages and Frameworks

Back-end

The chosen programming language for the back-end was **python** [90]. There were several reasons for this choice. Python is a general purpose and high-level programming language. The fact that it is general allows that the backend is entirely written in the same language although it must execute different tasks: it should provide the usual functionalities of a web server as well as generating recommendations from the retrieved data, which is rather a machine learning problem. Moreover, there are several Python libraries, such as Python-Recsys [91] or CRAB [92] for generating recommendations.

Python is a cross-platform programming language, a convenient fact since the development device is a Windows 10 computer while the server runs an Ubuntu operating system.

Django framework [93] was the technology used to implement the server functionality. This framework is an open source Python-based framework that is being developed since 2005 and that is widely used for server development. Some of the advantages of Django is that it is a very well documented framework, that it allows dividing any project into different and independent applications and that it has some typical back-end functionalities, such as a registration and authentication module, already implemented.

Django uses the Create, Read, Update and Delete (CRUD) philosophy [94], allowing the management of the databases indirectly, from the python code, which makes it very convenient for a database driven project, as the one being designed.

One of the main principles of the framework is the *Loosely Coupled* principle, making it very simple to rewrite and change parts of the implemented code, leaving others untouched.

This fact, the simplicity of starting new projects and the integrated functionalities it provides makes it a good option for developing prototypes.

The design pattern used by Django is a slight modification of the generally known Model-View-Controller (MVC) pattern [95]. In Django's case, the controller is the framework itself, using a design pattern that has been called by many Model-Views-Templates.

The Model is implemented in Django by a file called *model.py* and is the description of the database.

The Views is the part of the framework describing the business logic of the web service. It is the module that describes what happens in each moment. It is described in Django by a file named *views.py*.

The newly introduced concept into Django is the use of Templates. The templates describe how the application should look. Usually, they are implemented with HTML although a different mark-up language can be used. Django allows inserting some python code in the templates, which is executed before rendering the document. It is also possible to use templates inheritance, being able to create a base template that serves as the skeleton which is used on every page of the site and loaded only once.

Another advantage of Django is that it allows clean URLs. These URLs are the only part of the *controller* that can be accessed in this framework. They allow to define what method defined in the *views.py* file is used when an URL is accessed from a browser.

Frontend

For the implementation of the frontend the following technologies were chosen: **HTML5 [96], Bootstrap [97] and JavaScript (JS) [98]**.

HTML5 is the latest version of HTML and it is widely used for deploying web pages. Its main principle is that the browsers incorporate all the functionalities needed for the web pages, being the necessity of installing plug-ins for giving certain services. It was possible to record the video using the features provided by this framework, making it possible for the participants to take part in the experiment using the browser – a more convenient way than making them download any specific plugin for the experiment, which might have been a reason for deciding not to complete the experiment.

Bootstrap is an open source framework for making responsive and esthetical websites, that combines HTML, CSS and JavaScript. It was chosen for its simplicity and for the availability of open source templates that matched the designed interface and could be used minimising the time spent in the implementation of the aesthetical aspect of the site. The template used was obtained from *Creative Tim* [99]

JavaScript allows implementing the front-end dynamic functionalities. It is the most widely used language for this purpose. A library was found for doing the video recording using JavaScript. It is presented in section 5.4.1.4.

5.4.1.2 Database Selection

The most common practice is to use relational databases for RS using collaborative or content-based algorithms. Moreover, these databases are the most deployed and, as it is claimed in the blog post [100], a clear structure of the project is needed for deploying a data-driven database. To avoid extra difficulties it was chosen to use a relational database.

MySQL was chosen as the Database for the prototype. This Oracle relational database solution is one of the most mature of its kind as well as one of the fastest relational database available. The RS being designed will have to deal with plenty of numbers and unique reference in a consistent way, which can be achieved thanks to the indentation, relational and transactional features of MySQL [101]. Moreover, there are different RSs research works that use this database for the implementation of the prototype [20, 41, 44, 102, 103].

Even though the logical class diagram of the data collection platform, is presented in Fig. 5.2, for the implementation in the database some classes were combined in the same table. For a clearer understanding of the implementation part 5.4.2 a class diagram representing the tables and their attributes is presented in Fig. 5.5

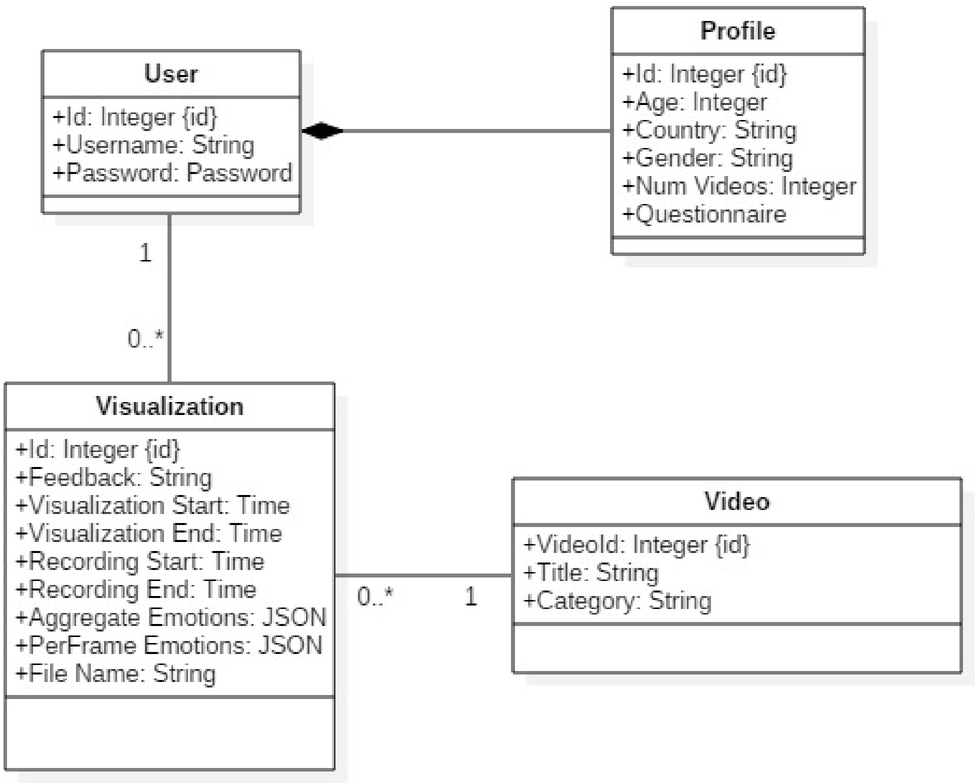

FIGURE 5.5: Database tables class diagram

5.4.1.3 Emotions API

This subsection details the process of obtaining the emotions from the recorded video. It will focus on the description of the sequence diagram of this process, leaving the implementation details for the following section.

As said in section 4.6, Microsoft Cognitive Services API is a Restful API. It can analyse either pictures or video and it returns a set of emotions. If the media sent is an image it returns a JavaScript Object Notation (JSON) file with the emotions shown in each of the faces present in the image with the position of each face in the image. On the other hand, if the media sent is a video the answers can be of two kinds:

- **Aggregate Emotions:** It returns a JSON file with the emotions detected in a video smoothed over time for getting a better accuracy of the results.
- **Per Frame Emotions:** This option returns the set of emotions detected in each frame of the video for each one of the faces.

As in the moment of deciding which one of the two to use for the data collection, it was difficult to guess which would be better for the generation of the recommendations it was decided to retrieve both results, making it possible to analyse them in following steps of the project.

Whereas for detecting emotions in an image the method used is a POST request - with the URL of the image specified in the body - and the HTTP response contains the detected emotions, for analysing a video is different. The analysis of the emotions of in a video requires some processing time, therefore there are two calls to the API. At a first instance, a POST request is sent with information of the URL where the video to be analysed can be accessed. To the named request, the API replies with a response containing the status of the request (if it was accepted or there was any other problem) and, in the case it was accepted, an operation ID The operation ID is used as the identifier for indicating the operation from which the emotions are trying to be retrieved in the following request, this time being a GET request. To the GET request, the service provides a response with the status of the operation (not started, uploading, running, failed or succeeded). If the status of the requested operation is succeeded, the answer also contains the emotions extracted from the video, in the form of <emotion, value>, for each frame or aggregation of frames. It also contains information about the time scale and the start of the section of video to which those emotions correspond.

The Microsoft Cognitive Services Emotion API also allows a method for analysing emotions in the video real time. However, although this fact was a determinant in choosing what emotion recognition service to use, it was left out of the data collection prototype due to time limitations.

In Fig. 5.6 the sequence diagram of how the API is called can be seen. As it is shown in the picture, three elements are present in the API call, the task manages, the database and the Emotions API.

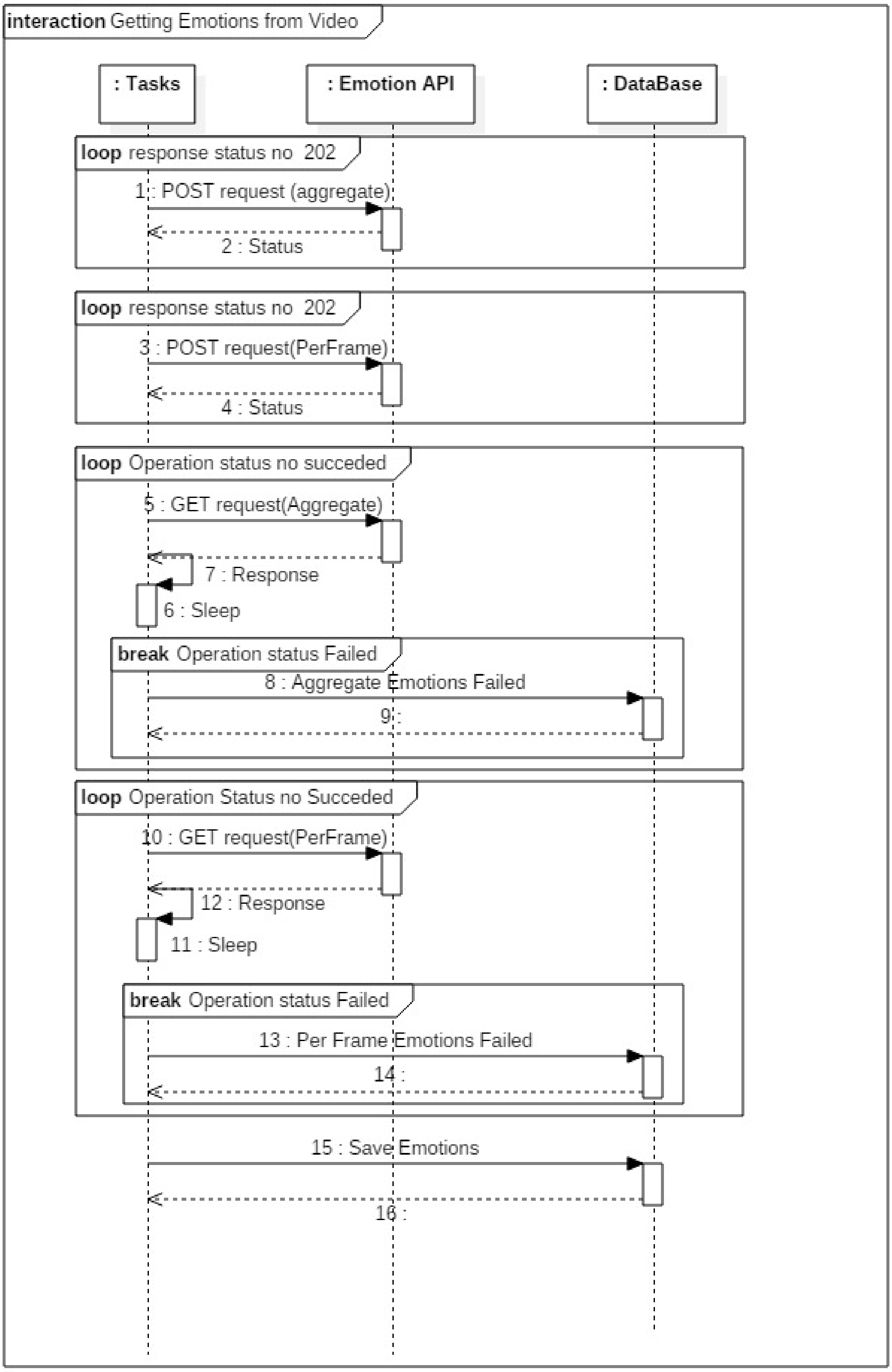

FIGURE 5.6: Sequence Diagram of the analysis of the emotions

In order to manage the API calls a task scheduler was needed: as stated in requirement NFR06 the waiting time of the user while doing the experiment should not be more than five seconds between pages. Having to wait for the request to be sent to the API, the video processed and the response obtained made the waiting time way higher than it was supposed. Therefore, the request had to be sent asynchronously, allowing to continue the processes on the client side when the requests are being processed

A list of the available schedulers for Django can be found in [104]. Out of them *Django-Celery* was chosen for being a stable version and written in Python (compatible with Python 3). No further research was done by comparing it with others.

Celery allows synchronous and asynchronous tasks and to set different workers to independently run tasks. The asynchronous tasks were used and only one worker was run, in daemon mode, because running more workers - i.e. calling the Emotions API at the same time with two different requests - caused the *429 - too many calls* error. The celery configuration was done following the steps in the tutorial provided by the celery official site [105]

A message broker was needed to work with celery and manage the queue of tasks. Celery allows multi-broker support, however, the official site recommends the use of RabbitMQ [106] and was the chosen one. For using it is was enough to configure it following the steps in [107]

5.4.1.4 Recording Video

For obvious reasons the recording of the video to be sent to the Emotions API had to be performed on the client side. Such task could have been done in two different ways: using HTML5 [96] or using plugins, such as Flash [108]. Conforming to the opinions of several experts on the web, out of the two, HTML5 is the technology outstripping [109–111]. HTML5 presents advantages over the plugins, as it is cross-device compatible, more secure and more stable. Moreover, it is a W3 specification and not a proprietary solution. The last reason to choose HTML5 was to simplify as much as possible the task of the respondents, avoiding that they had to install any plugin if they did not have it.

HTML5 allows to access the user media using the API *navigator.getusermedia()* [112]. This API allows the use of the user's microphone and camera, handling the necessary permissions. However, this API does not provide the functionalities for recording the videos or audios. The option for recording video offered by HTML5 is the MediaRecorder API [113]. This API can only be used with from secure origins only (HTTPS or localhost) [114], therefore this choice implies including in the implementation the obtainment of SSL certificates and the handling of HTTPS connections.

It was chosen to use a JavaScript library, named RecordRTC [115]. This library supports both of the APIs needed for the record of the video, is very well documented and offers a set of *demos* which complement the information in the documentation [116]. It is compatible (for the recording of video) with Firefox, Google Chrome, Opera and Android. According to W3Counter [117], the market share of the three mention browsers adds up to 70%, which indicates that most of the interested people will be able to take part in the experiment.

The chosen library was the only found choice for this functionality. It was decided to use it because as it combines the two APIs it facilitated the development of the front end. The availability of well-described documentation and demos were also a decisive factor.

5.4.2 Implementation

To continue, details about the implementation of the platform are provided.

5.4.2.1 Development and Deployment Technical Details

This section gives an overview of how the prototype for the data collection platform was implemented.

For the implementation process, two devices were used, a development device and a deployment device. The development one was a Windows 10 computer with Python, Django and MySQL installed. PyCharm [118] was used as the debugging software. The deployment machine run under Ubuntu server 16.04.2x32, it was hosted by *digitalocean.com* [119] and it had the following technical characteristics:

- Memory: 2GB with 2 CPUs
- Storage: 40GB SSD disk
- Transfer: 30 TB

At the beginning, more moderate technical characteristics were chosen; however, the chosen CPU (512 MB with one CPU) was not enough for the tasks to be conducted and had to be upgraded. The deployment server was accessed using a Secure Socket Shell (SSH) connection using *PuTTY* [120]. To be able to access to a Graphical User Interface in the server, xfce4 was installed - chosen for being a light interface, together with a VNC server. Setting the SSH connection to allow tunnelling the VNC server was accessed from the development device using VNC viewer [121].

The deployment server runs an Ubuntu 16.04 server operating system. Apache2 was installed on top of it to manage the HTTP connections. Apache2 was connected to Django by using the WSGI interface, provided by the Django package *mod_wsgi*.

A domain name was obtained from *www.namecheap.com* [122]. The acquired domain was: *recomotions.me*. The connection was secured using the Secure Sockets Layer protocol (SSL) obtaining the keys from the domain provider and configuring them as detailed in [123].

The Apache2 server was configured to accept any connection with the given domain name or IP address and sends it internally to the Django project, to allow it to handle the request. The following configuration file shows how all the connections to the server were redirected to make them secure, fulfilling the requirement NFR10.

5.4.2.2 Django Project Structure

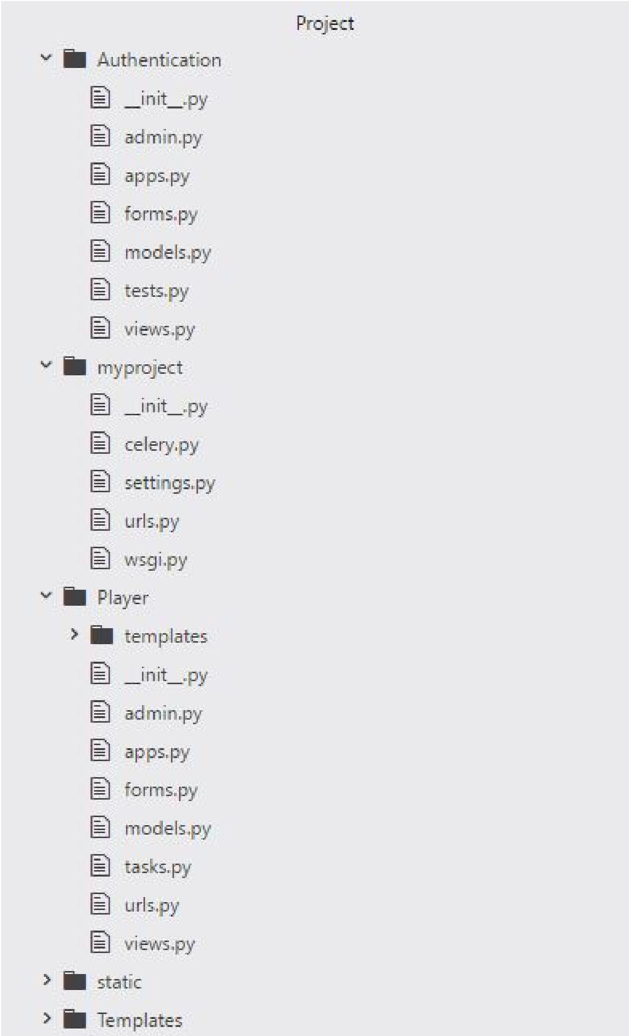

Figure 5.7: Django Project Structure

In Fig. 5.7 the structure of the Django project, with the different folders and files, is shown.

As explained in section 5.4.1, a Django project has the advantage to be divisible in applications that are independent of each other. For the data collection platform it was decided to use two applications:

- **Authentication:** application that uses the already provided by Django authentication facilities as a base for managing the registration and login and logout of users to the system. It also manages the creation of the user profile.
- **Player:** application managing the realization of the experiment, taking care of showing random videos to the user, and showing them the questionnaire when they have watched the three required videos.

As seen in Fig. 5.7 the Django project is comprised of five folders. The folders named *Authentication* and *Player* contain the python files describing the two applications mentioned in the previous paragraph. The folder *myproject* hosts the main configuration python files of the project. These three folders have the *__init__.py* file which indicates to the python processer that the folder contains python files to be used in the execution of the project.

The directories *static* and *templates* contain the static files (such as CSS, images or JavaScripts) and the django html templates, respectively.

myproject:

This folder contains the settings for the project.*celery.py* and *wsgi.py* set the two applications to use the configuration set in the *settings.py*. This file configures all the project options and there is only one per project.

One of the options to configure is whether Django should work in Debug mode or not, if debug mode is activated an error page will be shown when an URL is accessed and the framework detects some unignorable error. This feature is very useful; however it should be set to false for production. Among the settings configured in the named file it is set the *allowed hosts* for the project:

'ALLOWED_HOSTS = ["138.68.158.47", 'recomotions.me']'

The applications that are used in the project are also indicated in this file:

s_apps.py

```python
INSTALLED_APPS = [
    'player.apps.PlayerConfig',
    'authentication',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.staticfiles',
    'django_mysql',
    'django_celery_results'
]
```

As it was expected the authentication and the player apps appear in this configuration variable [3], together with several applications that are provided by the framework and that have been used to achieve the required functionality of the project. These applications are explained below:

The applications that are defined as *django.contrib.appname* are the ones provided by the Django framework. **Admin** allows to create an administrator interface for controlling the database from the browser, with no need of knowledge about database management. It is useful in those cases that users with no knowledge about databases need to manage the content and to check the models structure [95] and as it was the case of this project, in

[3] eventhough the code used for including them is different it provides the sabe functionality

which data are being obtained and stored properly. **Auth** facilitates the management of users, providing means to register and authenticate them as well as handling the log in and log out. **contenttypes** tracks all the models installed in the project, providing a high level and generic interface for working with them. **Sessions** is the Django tool for handling sessions in HTTP connections. As known HTTP is a stateless protocol and for being able to maintain parameters during the interaction of one user with the server is necessary to use a complementary technology. Django sessions handle the sessions using cookies [124].

The last two applications present in the configuration file are **mysql** which connects the framework with the database and **celery** for running background and parallel tasks.

The *settings.py* file also configures the middleware that the application uses, it can be found in the attached files. The most relevant middleware is the CSRF that contributes to the security of the application avoiding injections from POST requests by assigning a code to every POST [125].

The database to use in the project is configured as follows:

s_database.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'myproject',
        'OPTIONS' :{
            'charset' : 'utf8mb4',
        },
        'TEST' : {
            'charset' : 'utf8mb4',
            'COLLATION' : 'utf8mb4_unicode_ci',
        },
        'USER': 'myprojectuser',
        'PASSWORD': 'password',
        'HOST': 'localhost',
        'PORT': '',
    'TEST_NAME': 'test_myproject'
    }
}
```

The configuration used for the celery application is the following:

s_celery.py

```
#celery
BROKER_URL = 'amqp://'
CELERY_ACCEPT_CONTENT = ['json']
CELERY_TASK_SERIALIZER = 'json'
CELERY_RESULT_SERIALIZER = 'json'
```

```
CELERY_TIMEZONE = 'Europe/Madrid'
CELERY_ENABLE_UTC = False
```

Celery is configured to use rabbitmq as the queue manager.

As explained earlier in this section, the *url.py* file manages what to do when a connection is received. Below the content of the file is presented:

urls.py

```
urlpatterns = [
    url(r'^$', TemplateView.as_view(template_name='home.html'), name='home'),
    url(r'^about/$', TemplateView.as_view(template_name='about.html'), name='about'),

    url(r'^signup/$', authentication_views.signup, name='signup'),
    url(r'^profile/$', authentication_views.profile, name='profile'),
    url(r'^login/$', authentication_views.user_login, name='login'),
    url(r'^esp/$', authentication_views.switch_to_spanish, name='esp'),
    url(r'^en/$', authentication_views.switch_to_english, name='en'),

    url(r'^logout/$', auth_views.logout, name='logout'),

    url(r'^video/', include('player.urls')),
    url(r'^admin/', admin.site.urls),
]
```

Django uses regular expressions for defining and checking the URLs, this is why every URL start with *r*. The ând the $ delimit the url to what is exactly written in between, if any of the two was missing it would mean that it would match it with URLs containing the expression, not requiring them to be exactly the same.

The first two ULRs control when to show the home page or the about page. The home page is shown when only the base URL is used, the about will be shown after the base direction */about* is added. For both of them the method *TemplateView.as_view()* from django.views.generic.base has been used. This allows to directly render the template avoiding the call to a method defined in a views.py file.

The following five URLs call the corresponding method defined in the *views.py* file from the authentication application. The function of these methods will be explained below, however, their functionality can be inferred by their names: the first three manage the registration of a user, the creation of the user profile and the log in; the last two handle the change of language of the website.

The URL that manages the log out differs from the previous five because it uses directly the method provided by the Django auth application.

The second to last line of the urls patern uses a feature provided by Django, used for a better management of the URLs in a project. When the project grows the urls.py file can become difficult to set up. To solve the inconvenience, Django allows to define URLs in independent *urls.py* files within each application and include them in the main one. As the project is configurated, the framework knows that when the requested URL begins by */video/* it has to look for the definition in *player/urls.py*.

The last line is similar; in this case it indicates the framework to use the URLs provided by the admin application.

Authentication:

This application, as said earlier, uses the *auth* application provided by Django. It is also responsible of executing the change of language of the site when is necessary.

The database tables belonging to this application are defined as follows:

authentication_models.py

```
class Profile(models.Model):
    GENDER_CHOICES = (
        ('M', 'Masculin'),
        ('F', 'Femenin'),
    )
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    country = models.CharField(max_length=30, default="Spain")
    gender = models.CharField(max_length=1, choices=GENDER_CHOICES,
    ↪   default="F")
    age = models.PositiveIntegerField(default=0)
    num_videos = models.PositiveIntegerField(default=0)
    questionnaire = JSONField()
```

The above python code creates a database table with six fields. The number of videos and the questionnaire answers are included as user profile information to simplify the tables structure. The field user is defined as a OnetoOneField, which indicates the one to one relationship between user and profiles. One user can have only one profile and vice versa.

As shown in Fig. 5.7, the application authentication has a file named *forms.py*. It is used in Django to configure forms for asking inputs to the users. For the authentication application is simple since the forms are defined using the database models as the base. The code can be seen in the attached files.

The file *views.py* defines the business logic of the application. The code can be found in the attached files. The methods defined in this file are explained below:

The methods *user_login(), signup() and profile()* are very similar. The three take a parameter, as most of the methods defined in Django, that is the *request* used to keep the session during the connection. When the method is called it checks whether the request is a POST request or not. If it is not, it returns a rendered html template (corresponding with the purpose of the method) in which the also corresponding form is passed as a variable. Given the case that the request is from the type POST, it gets the data contained in it, corresponding to the answers of the form, and does the pertinent actions with it. If it is the *login*, it authenticates the user; if it is the *sign up*, the user is saved to the database and logged in; and if it is *the creation of the profile*, it is created. If there are no errors in the given information and the tasks are successfully completed the method returns a *redirect* to the following step of the experiment – after the login it shows the home page, after the registration it shows the profile creation page and after the profile creation page it shows the page with the explanation of the experiment.

The language of the site is handle by adding a parameter in the request session named lang. If this parameter is set to "en" the web site will be shown in English, and when it is "esp" the webpage will be in Spanish. The methods *switch_to_english()* and *switch_to_spanish()* change the value of the language variable when they are invoked.

Player: The Player application is the one managing most of the experiment functionalities. The *models.py*, defining the database, is as follows:

player_models.py

```python
class Videos(models.Model):
    video_id = models.CharField(primary_key=True, max_length=200)
    tittle = models.CharField(max_length=200)
    channel = models.CharField(default='', max_length=200)
    duration = models.CharField(default='', max_length=20)
    tag = models.CharField(default='', max_length=500)


class Visualization(models.Model):
    LIKED_CHOICES = {
        ('U', 'Unknown'),
        ('Y', 'Yes'),
        ('N', 'No'),
    }
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    video = models.ForeignKey(Videos, on_delete=models.CASCADE)
    file = models.CharField(max_length=20)
    visualization_time = models.DateTimeField(auto_now_add=True)
    emotionsAggregate = JSONField()
    emotionsPerFrame = JSONField()
    feedback = models.CharField(max_length=10, default='U, U')
    startRecordingTime = models.TimeField()
    stopRecordingTime = models.TimeField()
    startVideoTime = models.TimeField()
    stopVideoTime = models.TimeField()
```

It defines two tables, one that saves the videos and one that stores the visualizations. The table with the videos stores the video id (used as key), the title of the channel where the video is hosted, its duration and the tag (or category name) that is assigned to it. The table with the visualizations stores the user that watched the video and the watched video with the relevant information about that visualization.

The *urls.py* file is the part of the framework that is closer to act as the controller of the common Model-Controller-View design pattern. The method to call after a requested URL is defined in this file. The URLs for the *Player* application can be seen below:

player_urls.py

```
urlpatterns = [
      url(r'^1', views.video_intro, name='intro'),
      url(r'^2', views.video_check, name='check'),
      url(r'^3', views.video_player, name='player'),
      url(r'^4', views.video_feedback, name='feedback'),
      url(r'^5', views.survey, name='survey'),
      url(r'^6', views.end, name='end'),
      url(r'^7', views.first_part, name='first_part'),
]
```

Every URL pattern calls a method defined in the *views.py* file of the same application. The following paragraphs describe the function of each one of these methods.

When the first ULR is used, the method *video_intro()* (it can be found in the attached files) is called. This method checks for the user that is logged in (obtaining it from the attribute session in the request that is given as a parameter to the method) the amount of videos the user has watched. If the watched videos is one or more, following with the design specifications shown in the sequence diagram in Fig. 5.3, the user is directly redirected to the camera check page to start visualising videos. If the user has not watched videos yet, the method returns the rendered html template with the explanation about the experiment.

When the last URL of the list is requested (this will happen when the users reads the information about the experiment and presses the "Okay" hmtl button in the shown template), the method *first_part()* is called. This method is the simplest possible: it directly returns the corresponding html template rendered. The next ULR that is requested, after the user presses the confirmation button, is the one with the number 2. It calls the method *video_check()* that has the same functionality as the previous one, returning a different html template.

The next URL, requested once again when the user presses the corresponding html button calls the *video_player()* method. When this method is called there are three things that can have happened:

1. A user gets to the player site and needs a random video, that he has not watched before

2. A user has started watching a video, decided that he did not want to continue watching it and asked for a new video. In this case, a video of the user expressions has been recorded and sent to the server

3. A user has finished watching a video and the video of his emotions has been recorded and sent to the server. The user needs to be redirected to the feedback form for giving his opinion about the video.

video_player.py

```python
@login_required(login_url='/login')
video_player(request):
    if request.method == "POST" and len(request.FILES) != 0:
        if request.POST['finished'] == "no":
            file_name = random_filename()
            video_emo = request.FILES['video-blob']
            random_video = request.session['video_r']
            start_recording_time = request.POST['startRecordingTime']
            stop_recording_time = request.POST['stopRecordingTime']
            start_video_time = request.POST['startVideoTime']
            stop_video_time = request.POST['stopVideoTime']
            save_video(video_emo, file_name)
            u = request.user
            instance = Visualization.objects.create(user=u,
                video=Videos.objects.get(video_id=random_video), file=file_name,
                emotionsAggregate={'value': 'no emotions'},
                emotionsPerFrame={'value': 'no emotions'},
                startRecordingTime=start_recording_time,
                stopRecordingTime=stop_recording_time,
                startVideoTime=start_video_time, stopVideoTime=stop_video_time)
            user = Profile.objects.get(user_id=u.id)
            user.save()
            tasks.emotions_api.delay(file_name, instance.id)
            request.session['visualization_id'] = instance.id
            return redirect('video:player')

        else:
            file_name = random_filename()
            video_emo = request.FILES['video-blob']
            random_video = request.session['video_r']
            start_recording_time = request.POST['startRecordingTime']
            stop_recording_time = request.POST['stopRecordingTime']
            start_video_time = request.POST['startVideoTime']
            stop_video_time = request.POST['stopVideoTime']
            save_video(video_emo, file_name)
```

```
            u = request.user
            instance = Visualization.objects.create(user=u,
                video=Videos.objects.get(video_id=random_video), file=file_name,
                emotionsAggregate={'value': 'no emotions'},
                emotionsPerFrame={'value': 'no emotions'},
                startRecordingTime=start_recording_time,
                stopRecordingTime=stop_recording_time,
                startVideoTime=start_video_time, stopVideoTime=stop_video_time)
            user = Profile.objects.get(user_id=u.id)
            user.num_videos += 1
            user.save()
            tasks.emotions_api.delay(file_name, instance.id)
            request.session['visualization_id'] = instance.id
            return redirect('video:feedback')
    else:
        random_idx = random.randint(0, Videos.objects.count() - 1)
        random_video = Videos.objects.all()[random_idx]
        request.session['video_r'] = str(random_video)
        return render(request, 'player/video_player.html', {
            'video_id': random_video
        })
```

The code for this method is shown above. The first *if* discriminates between the first option and options two and three. If the situation that happened before calling the method is number one the evaluation of the *if statement* would result *False* and the code under the else would be executed.

This part of the code generates a random number between zero and the number of videos the database [4] and uses this number as the index to obtain the key -and YouTube video ID – from the list returned by the Django method *Videos.objects.all()*. Afterward the template is rendered, passing the video ID as parameter.

If the options two or three are the ones that happened the first *if* expression is evaluated as true. It will be explained later that the front end send the video in the File field of an HTTP POST request. Afterward there is another comprobation, the POST request contains a variable named "finished" that is initialized to "no" and changes to "yes" when the user finishes watching the video. If the user finished watching the video a random file name is created for it, and after all data contained in the POST message are obtained (the video in a blob format, and the time of the record and the playback start and end, fulfilling requirements FR11 and FR12). Later, the file is saved and a new instance is created in the visualizations table, the number of videos watched is increased in the user profile and the *emotions_api()*

[4]The reason for checking the database each time was to possibly allow users to look for a video in YouTube and add it to the database, functionality that was finally not implemented

is called with the delay() option to execute it when the celery worker is available. Finally the method returns a redirect statement to the URL in charge of the video feedback.

When the option described as two happens, the executed code is a combination of the two that have already been explained. A new random video ID is returned to the user and a new visualization instance is saved; however, the number of videos watched by the subject is not increased.

For creating the random file name and for saving the video the methods *random_filename()* and *save_video()* are called. The methods can be found in the attached files. It can be noticed that these two methods, as an exception, do not take a request as a parameter.

After watching a video the user is redirected to the URL managing the video feedback. When this happens the method named *video_feedback()* is called. If the request given as parameter is not a POST message, the method returns the template for the feedback site rendered with the form as parameter (the feedback form is defined in the *forms.py* file that can be found in the attached files. In the case the request is a POST the code will update the corresponding visualization entry with the information of the user's feedback. The visualization is obtained because it had been saved as parameter in the request session field in the method *video_player()*. Once this is completed it check the amount of videos the user has watched. If the number is different than three the method returns a redirect to the page for checking the camera, if the amount of videos is exactly three it returns the questionnaire site.

When the URL with the number five is requested the method *survey()* is called. This method has a local variable "questions" that includes the questions in dictionary format:(qn, question) being n the number of the question. As first instance, the language variable of the request is check and the texts are adapted to this language. If the request is not a POST, request the *random_suffle()* method is used to randomize the order of the questionnaire and it is passed as a variable for rendering it with the template for the questionnaire. If the request is a POST the answers of the questionnaire are saved to the database in a JSON form, using as keys the same ones as in the "questions" variable and as values the number selected by the user for the matching question.

Once the answers are saved the method returns a redirect to the URL that calls the *end()* method. This method, is again very simple, just returning the rendered html template.

The last file to describe from the Player application is *tasks.py*. Here the tasks that the celery worker will execute are defined. The purpose of having the tasks asynchronous is to fulfil the requirement NFR06 which states that the user should not wait during the realization of the experiment.

As explained in section 5.4.1.3, the Microsoft Cognitive Services Emotion API allows to get the emotions in two different ways: aggregate or per frame. Both are retrieved. However, this must be done in two different API calls; with the same request both answers cannot be obtained. The code for executing this call is shown below. It has been simplified showing only one of the requests, since they are the same except for one parameter in the header. The complete file can be found in the attached files.

tasks.py

```
@shared_task
def emotions_api(fileName, visualizationid):
    api1 = '205a0373d21a4c3baafbde484d359d7d'
    logger.info('Calling API')
    headers1 = {
        'Ocp-Apim-Subscription-Key': api1,}
    params1 = urllib.parse.urlencode({
        'outputStyle': 'aggregate',})
    body1 = {
        "url": "https://recomotions.me/temp/" + fileName,}
    body1 = json.dumps(body1)

    data1 = None
    while data1 is None:
        conn1 = http.client.HTTPSConnection('westus.api.cognitive.micro
        ↪ soft.com')
        conn1.request("POST", "/emotion/v1.0/recognizeinvideo?%s" %
        ↪ params1, body1, headers1)
        response1 = conn1.getresponse()
        logger.info('Status of the response1: ' + str(response1.status))
        logger.info('Checking first Oid')
        if response1.status != 202:
            logger.info('Too many calls to the API, waiting 1
            ↪ minute...')
            sleep(60)
        data1 = response1.getheader('operation-location')
        sleep(20)
        conn1.close()
    oid1 = data1.rsplit('/', 1)[1]
    logger.info('Got the operation ID1: ' + oid1)


    data3 = {'status': ''}
    delete_video = 1

    while data3['status'] != 'Succeeded':
        logger.info('the status3 is: ' + data3['status'])
        logger.info('waiting 200 seconds...')
        sleep(200)
        logger.info('waiting  done, calling API for answer')
        conn3 = http.client.HTTPSConnection('westus.api.cognitive.micro
        ↪ soft.com')
        logger.info('Conexion stablished, waiting 60 seconds')
        sleep(60)
```

```
        conn3.request("GET", "/emotion/v1.0/operations/" + oid1, "{body}", headers1)
        response3 = conn3.getresponse()
        aux3 = json.loads(response3.read().decode())
        if 'status' in aux3:
            logger.info('the new status3 is: ' + aux3['status'])
            data3 = aux3
            if 'Failed' == aux3['status']:
                logger.info('Something went wrong, the analysis failed to get aggregate emotions of:' + fileName)
                data3['processingResult'] = {'emotionsAggregate': 'failed'}
                delete_video = 0
                conn3.close()
                break
        else:
            logger.info('the response3 didnt contain status')
        conn3.close()

    if data3['status'] == 'Succeeded':
        logger.info('Got the aggregate emotions')
        Visualization.objects.filter(pk=visualizationid).update(emotionsAggregate=data3['processingResult'])
        logger.info('aggregate emotions saved')

    if delete_video == 1:
        os.remove('/home/alicia/myproject/temp/' + fileName)
        logger.info('\n')
        logger.info('File deleted: ' + fileName)
```

The method *emotions_api()* starts by defining the parameters of the request. In the body, the URL where the video can be found is defined. The file name is assed to the method as a parameter when it is called.

An HTTP POST request is sent using the *HTTPSConnection()* and the *request()* methods from the HTTP client library. The response is obtained with the method from the same library *getresponse()* and the status of the response is checked, if the status is not 202 (accepted) the connection is closed and the program waits one minute for creating a new request. If the request was accepted still it did not contain an operation id, needed to retrieve the emotions posteriorly, it waits 20 seconds and closes the connection and tries again.

Once the operation id is obtained a new connection is opened and a GET request is sent containing the operation id to retrieve the emotions. The connection is opened, the request is sent and the connection is closed repeatedly until the status in the response is succeeded. An exception is when the response's status is Failed, this case is left not handled, the loop stops and that visualization does not get the corresponding emotions. Once the video is analyzed and the emotions are obtained, they are stored in the corresponding visualization entry and the video is deleted from the storing folder. If the emotions have not been acquired, the video is not deleted for sending it later manually to the API.

5.4.2.3 Django Project Templates

The front-end was developed using the template solution provided by Django together with the technologies mentioned earlier in section 5.4: HTML5, JavaScript and Bootstrap. As it was also mentioned, a boostrap desing was used, obtained from [99]. For so, the JS and CSS files describing the template were downloaded and added to the static files of the website.

As mentioned earlier, the template system of Django allows the inheritance of templates, to avoid repetition of code and the reloading of static content in the client side. A base template was defined (*base.html*) with the common elements for all the pages. It can be found in the attached files.

In the code, tags of the form: {{ *variable* }} or {*% python code %*}. These are the elements provided by Django to insert python code and variables into the html code. In the *base.html* file the {*% block name %*} and {*% endblock%*} tags are used to indicate that in between them, more html code will be inserted from the templates extending the base one. It is also used to define if statements in the form of {*% if condition %*}, {*% else %*} and {*% endif %*}.

The language selection, as explained in the previous section, is done by setting a variable in the user session with information about the desired language. The if statements check what is the value of that variable and display the correspondent test.

The rest of the pages are implemented using common HTML tags combined with some bootstrap elements. All of them contain at beginning the tag {*% extends 'base.html' %*} which indicates the mentioned inheritance. The files can be found in the attached files.

Most of the documents do not need of further explanation. However, it is worth to mention the use of the {*% csrf_token %*} which is part of the Django middleware contributing to the security of the page by inserting a code in each POST request that is checked at the receival. In the *survey.html* a {*% for key, value in questions.items %*} is used to iterate over all the questions in the questionnaire.

The two files that are not basic HTML are the *check.html* and the *video_player.html*. These two files use the RecordRTC libraty to access the user media, record a video and send it to the server. Both of them can also be found in the attached files. An explanation of their function is given below.

The file *video_player.html* contains JavaScript to control the recording of the video. When the page is loaded the method *captureUserMedia()* is called and by using *getUserMedia()* the camera is accessed and the video starts recording by calling *RecordRTC()* and *startRecording()*. Afterward the video player is created. An event listener is set to capture when the video starts playing, and save the time, and when the video is finished. At the video finished event it calls the method *stopRecording()*. When this method is called the video is sent to the server by appending the time variables and the video in a blob format in an HTTP POST request.

Figs. 5.8 and 5.9 shows the final aspect of the data collection platform.

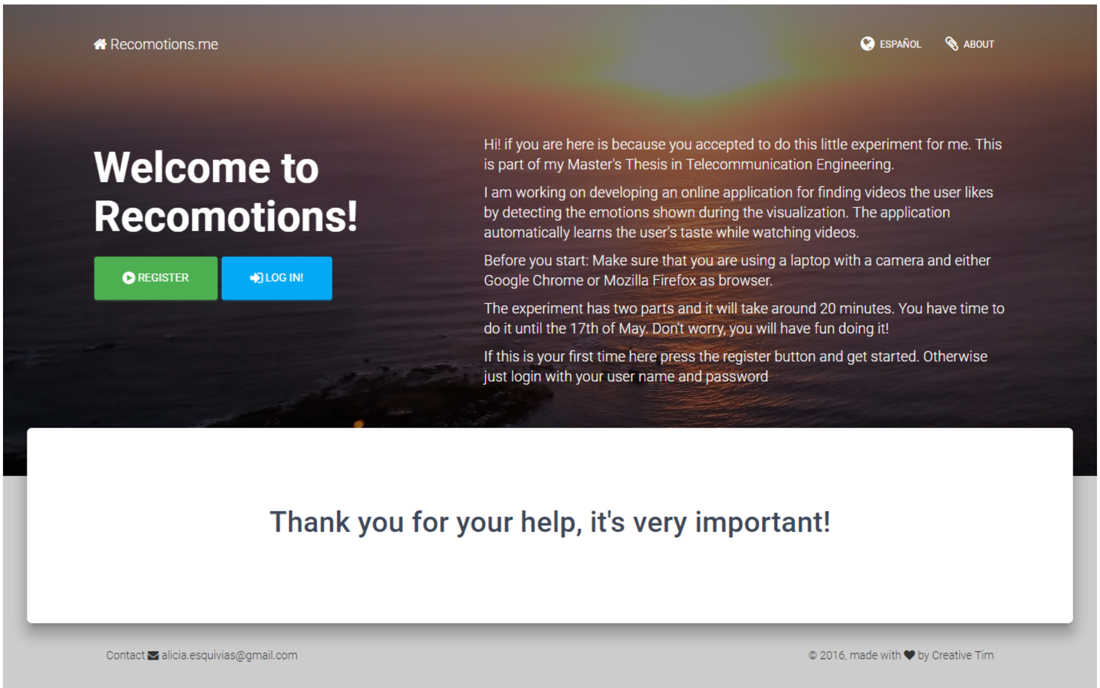

FIGURE 5.8: Introductory page

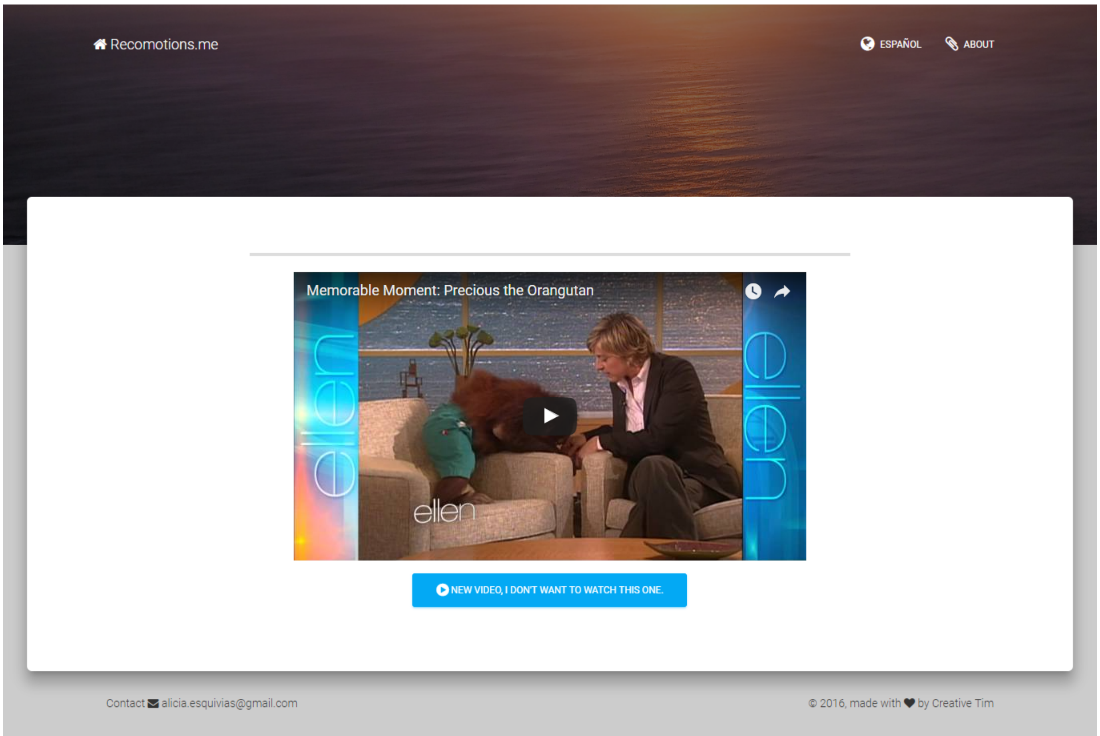

FIGURE 5.9: Video Player

Chapter 6

Analysis of the Collected Data

This chapter presents an analysis of the data that has been collected following the steps explained in chapter 4 and using the data collection platform prototype described in chapter 5. Afterwards, an analysis of the adoption model is presented. Finally, different options for creating recommendations with the collected data, based on the literature on the topic are proposed and analysed. [1]

6.1 Analysis of Collected Emotional data

The current section presents the data about emotions that have been gathered during the realisation of the experiment and discusses the possible relevance of these results.

After the period of realisation of the experiment **497** visualisations were obtained. Out of these visualisations, **60** had to be discarded due to the lack of the correspondent emotions. The causes of the missing emotions were two: either the video of the person watching the content was not sent properly to the server due to malfunctioning of the front-end or the Microsoft Cognitive Services Emotion API response was *Failed* for three different trials.

Table 6.1 shows the number of videos that were watched and the number of subjects that watched videos.

TABLE 6.1: Info about visualizations

Number of Visualizations	437
Number of Watched (unique) Videos	282
Number of Users	91

[1] Collected data accessible in: $https://www.dropbox.com/s/ln02w703b72pz03/emotion\_info.zip?dl=0$

6.1.1 The collected dataset

The dataset gathered in the data collection process has the following characteristics:

- Contains **435** video visualizations of **282** videos and **91** users
- The visualizations are not uniform in duration.
  - Minimum Duration: 10 seconds
  - Maximum Duration: 4 minutes and 15 seconds
- For each visualization the following information is available:
  - Video id (assigned by YouTube)
  - User id
  - Aggregate Emotions [2]
  - Per Frame Emotions [3]
  - The following time information:
    * Start of the video recording
    * Start of the visualisation
    * End of the visualisation
    * End of the video recording
  - Feedback from the user [4]:
    * Answer to: "Did you like the video?" (yes, no or unknown)
    * Answer to: "would you watch something similar again?" (yes, no or unknown)
- For the users:
  - age
  - gender
  - native country

[2]The API response for the aggregate emotions and the per frame emotions is a set of eight emotions (anger, contempt, disgust, fear, happiness, neutral, sadness, surprise) for intervals or frames of video, respectively.

[3]The frequency of the emotions is not equal for all the visualisations since the framerate of the recorded video depended on the settings of the browser.

[4]when the answer to the questions is unknown it means the user decided to skip the visualisation since the feedback was required to continue the experiment. It is not possible to have unknown for one of the questions and yes or no for the other

6.1.2 Emotion API response

A stated in section 5.4.1.3 the emotional information for each visualisation was obtained using the services provided by Microsoft. The API allows choosing between two kinds of results: Aggregate Emotions over time and over faces in the video, and the raw emotions per frame.

Both options were used to retrieve the emotions. Examples and descriptions of the response by the API in the two cases are shown below.

In the case of aggregate emotions, the response is as follows:

Aggregate Response

```
{'fragments': [{'duration': 2000,
   'events': [
     ...
    [{'windowFaceDistribution': {'anger': 0,
       'contempt': 0,
       'disgust': 0,
       'fear': 0,
       'happiness': 0,
       'neutral': 1,
       'sadness': 0,
       'surprise': 0},
      'windowMeanScores': {'anger': 0.00137681,
       'contempt': 0.037883799999999995,
       'disgust': 0.00131425,
       'fear': 0.0013009,
       'happiness': 0.033777299999999996,
       'neutral': 0.651744,
       'sadness': 0.272037,
       'surprise': 0.000566194}}],
    [{'windowFaceDistribution': {'anger': 0,
       'contempt': 0,
       'disgust': 0,
       'fear': 0,
       'happiness': 0,
       'neutral': 1,
       'sadness': 0,
       'surprise': 0},
      'windowMeanScores': {'anger': 0.0005176040000000001,
       'contempt': 0.035431199999999996,
       'disgust': 0.000415276,
       'fear': 0.0007552720000000001,
       'happiness': 0.0129375,
       'neutral': 0.58952,
```

```
        'sadness': 0.36008399999999996,
        'surprise': 0.000339304}}],
        ...

    'interval': 500,
    'start': 0},...

  {'duration': 2000,
  ...
    'interval': 500,
    'start': 2000},
  {'duration': 2000,
    ...,
    'interval': 500,
    'start': 4000},
  ...
 'framerate': 30,
 'height': 480,
 'offset': 0,
 'timescale': 1000,
 'version': 1,
 'width': 640}
```

The example shown above is only a part of a response file. The response is a JSON file containing a set of fragments and additional information about the video that had been sent, which can be found at the end of the response example. This additional information regarding the videos provides the frame rate (in frames per second) the height and width of the multimedia content (in pixels), the version of the API, an offset [5](at the moment it is always 0 but it is intended to be implemented in the future), and a time scale.

The time scale provides information on the number of *ticks* per second in the video. This information is necessary because the starting and duration times of the fragments are given in ticks, as well as the length of the intervals inside each fragment.

In the response, each fragment contains intervals denominated by the name of events. In the previous example, it can be seen that every fragment contains information for each of the events and additional information stating the start and duration of the fragment and the length of the intervals. Each event contains information about the emotions detected in the correspondent interval in two different forms. The first one, **windowFaceDistribution**, give a value of “0” or “1” for each emotion, obtaining “1” only the predominant emotion through the interval. The second set of values, **windowFaceDistribution**, assign a value between “0” and ”1” to each emotion. The addition of all the values is the unity. These values should be understood as the relative importance of the emotion. This value can also be considered as the confidence on the reliability of the detected emotion, as the closer the

[5]defined in the documentation as “The time offset for timestamps”

value is to the unity, the more reliable is the detection [26]. The returned emotions are the mean values of the detected emotions for each frame of the video sent to the API. The documentation does not specify if the mean values are calculated as a simple arithmetic average or as a more complex one such as a running average.

The second response provides the emotions detected in each of the frames. An example is shown below:

Per Frame Response


```
{'facesDetected': [{'faceId': 0}],
 'fragments': [{'duration': 167, 'start': 0},
  {'duration': 33,
   'events': [[{'height': 0.39375000000000004,
      'id': 0,
      'scores': {'anger': 0.00110905,
       'contempt': 0.016029599999999998,
       'disgust': 0.000788161,
       'fear': 0.0012400900000000001,
       'happiness': 0.060108299999999996,
       'neutral': 0.763054,
       'sadness': 0.157102,
       'surprise': 0.000568109},
      'width': 0.29531199999999996,
      'x': 0.40312499999999996,
      'y': 0.44375000000000003}]],
   'interval': 33,
   'start': 167},
...
```


As in previous aggregate emotions response, the example only shows a part of a complete response. It also contains fragments with intervals; however, in this case, the intervals correspond to a frame, and therefore the amount of intervals or sets of emotions obtained per video depends on the frame rate.

In this example, it can be seen that the intervals are shorter (33 ticks versus the 500 ticks in the previous example). The response, in this case, contains only a single set of emotions per frame, given in a “0” to “1” scale and adding up the unity altogether. The response per frame would also provide the emotions for each of the faces detected in the video. In this project, the group topic has not been considered, as stated in the limitations section 1.4, and consequently, this option was not tried. In the example it can be observed that within the information regarding the events there is a *key* indicating the “id”, and at the beginning of the example, the first entry is a dictionary representing the “ids” for the faces detected (only one in the example). Together with the emotions, for each frame, the API returns the position of the face in the analysed image.

6.1.3 Grouping the responses

As it was explained in section 4.5, after each visualisation the users were asked to answer to simple "yes" or "no" questions:

- Did you like the video?
- Would you watch something *similar* again?

The purpose of including the questions is collecting data for being able in the future to train an RS, allowing the possibility to use supervised machine learning techniques by evaluating if the predictions obtained from the system are correct. In order words, even though the system is aimed to use only implicit feedback, it was decided to obtain explicit feedback in the data collection for having later more options when designing and implementing the RS.

There are four combinations of the answers to the two questions. A fifth group can be added by considering those videos that do not have the feedback, which corresponds to those that were skipped by the user. The five combinations are used throughout the analysis to group the visualisations by the explicit feedback and to study if there is any pattern in the responses and in the detected emotions. The groups, with the corresponding ratings, are presented in table 6.2.

TABLE 6.2: Groups according to the feedback

L	R	Group	Description
-	-	*0*	The user did not want to watch the video and skipped it before finishing the visualization. It is understood as the lowest possible rating, however this group is sligtly more complex beacuse the reason to not watching the video might have been other, such as having already watched the video
0	0	*1*	The user did not like the video and would not watch something similar again. This two ratings make clear that the user at the time does not like videos as the shown one
1	0	*2*	The user like the video but would not watch something similar again, which for the recommendations means that the category to where the video belongs should not be recommended
0	1	*3*	The user did not like the video and would watch sothing similar again, having for the recommedatios more positive weight this options than the
1	1	*4*	The used like the vide and would watch something similar again, being this the best possible rating

As it can be seen in the table, these groups are targeted to create the recommendations using a collaborative filtering algorithm combined with content-based filtering. There are different combinations of the explicit ratings obtained that could be used and could be dequate for other purposes.

Table 6.3: Number of visualizations per groups of ratings

Group	nº of visualizations
0	136
1	78
2	17
3	48
4	158

The number of videos obtained for each group is shown in 6.3. Group 2 is the one with the lowest number of visualisations, a fact that is understandable since it is more probable when someone likes a video that he would like to watch something similar than the opposite. The subjects providing these answers could be those that liked the video; however, they would not watch something similar because they do not usually watch videos online. The second lowest group is Group 3; nonetheless, it has a substantial number of visualisations, proving that the approach of including two questions was correct to extract more information about the users. Disregarding group zero, as it is a special case because skipping videos could be done as many times as desired by the users, the most popular groups are 4 and 1, in which the answers to both questions are the same.

6.1.4 Aggregate Emotions

This subsection presents an initial analysis of the retrieved aggregate emotions. For this purpose, a mean value of the emotions through the visualisation was calculated and compared for the different groups. When examining the results obtained in the aggregate emotions, it was noticed that some of the responses (concretely 27), had all the values of the emotions in the responses set to zero. This fact was correlated with the shorter videos, and it happened mostly for category 0, in which the user was typically recorded for a few seconds, instead of a long video. It is possible, even though it is not stated in the API documentation, that for a short video the aggregation framework of the emotions API does not function properly.

As a first result, it can be seen that the most predominant emotion throughout the groups is neutral. This result does not surprise since it is the normal state of a human face. The second more detected emotion is happiness, which is also an expected result since the selected videos have an entertaining purpose and many of them are comical.

The mean percentage of the emotions for each group is shown in Fig 6.1. These values were calculated by: i) averaging the emotions returned by the API for each visualisation and ii) averaging again the emotions for those visualisations belonging to each group.

Figure 6.1 represents, for each emotion, the percentage of appearance in each group. As it was said the predominant emotion is neutral, followed by happiness and sadness. The emotions that show the lowest appearance are disgust and fear, which are the two emotions being in beta mode as advised by the API. [26]. The relation between the emotion values can be noticed in the figure: when the happiness value increases there is a similar decrease in the neutral value.

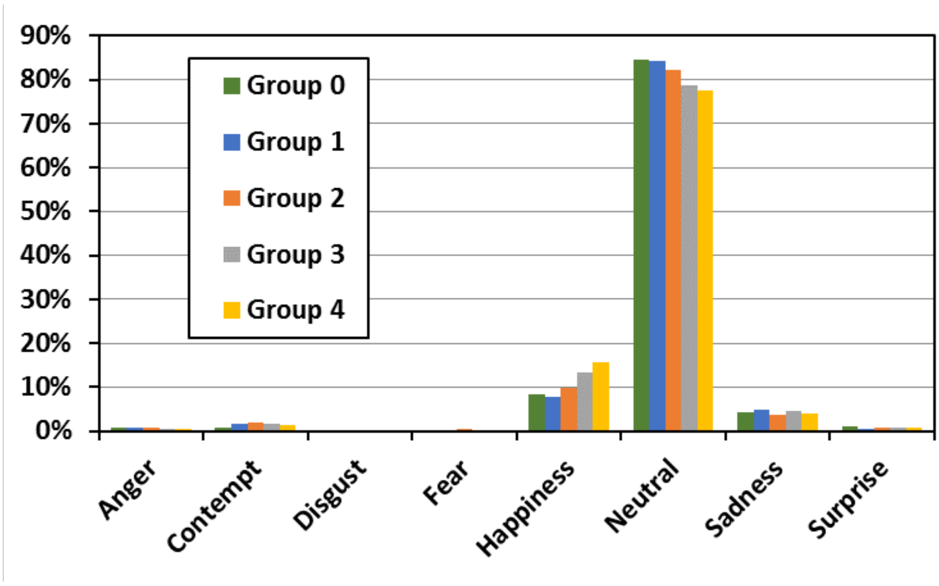

FIGURE 6.1: Distribution of emotions

In Figs. 6.2 and 6.3 the mean percentage of happiness and neutral emotions for each of the groups can be seen more in detail. It can be noticed that the values of the y-axis are different for each graph: happiness presents a maximum value in group 4 of 16% while neutral presents 82% in *group 0*. In Fig. 6.2 it can be seen how the percentage of happiness increases with the group number, indicating that there is a correlation between the presence of happiness and the feedback of the users about the watched video. This figure is relevant for the research because it proves that there is information that can be extracted from the automatically detected emotions that could be used for predicting what a user would like and therefore for generating recommendations.

The decrease of the neutral emotion is explained by the increase of happiness. However, the decrease of this emotion is less abrupt than the increase of happiness, meaning that in the lower groups there are other detected emotions. It is interesting to see that group 0 does not exactly follow the same pattern: the percentage of neutral emotion is the higher for all the

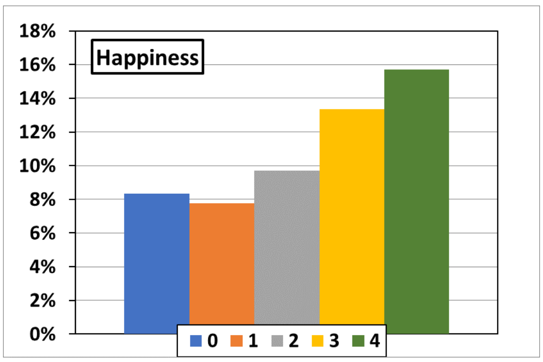

FIGURE 6.2: Detected Happiness by group

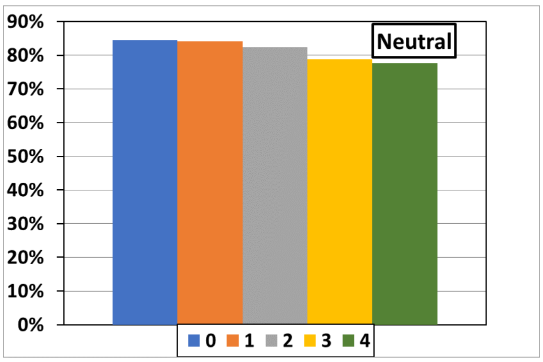

FIGURE 6.3: Detected Neutral by group

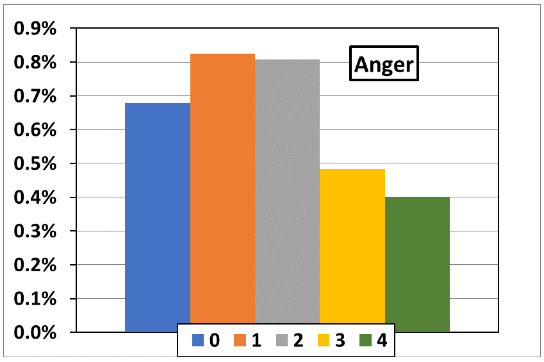

FIGURE 6.4: Detected Anger by group

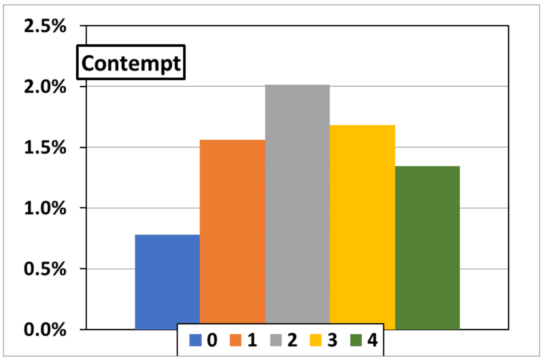

FIGURE 6.5: Detected Contempt by group

FIGURE 6.6: Detected Sadness by group

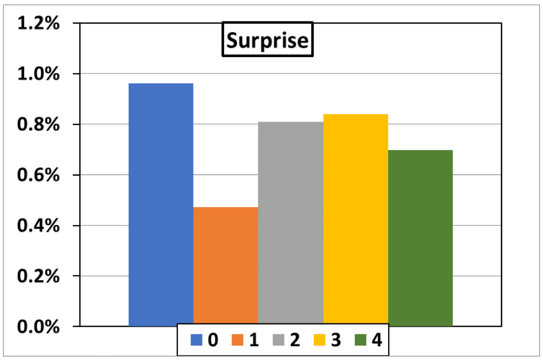

FIGURE 6.7: Detected Surprise by group

groups, however, it shows more happiness and less sadness than group 1. This could be due that in the entry stage (as defined in the framework presented in [12]) the emotions tend to be more positive; however, further analysis should be done considering the three stages to confirm this fact.

Figures from 6.4 to 6.7 show the percentage of anger, contempt, sadness and surprise for each group. These values are small compared with the happiness value and much smaller compared with the neutral emotion. However, it is interesting to analyse their distributions. In Fig. 6.4 it is noticeable that the anger levels decreases when the group number increases, i. e., when the user liked the video. The contempt emotion(6.5) is maximum in the middle group, a result that also makes sense since the visualisations belonging to the third group are those that the users neither like or disliked extremely. Sadness, Fig. 6.6, is more present in the lower categories and has its lowest value in the middle category. In this preliminary analysis, it is difficult to state the reason to sadness being higher for groups 3 and 4 than for 2. Lastly, the surprise (Fig. 6.7) is maximum group 0 and the minimum in group 1. Again it is difficult to determine in a preliminary analysis the reasons for it.

From analysing the aggregate emotions, it can be determined that there is a correlation between the emotions detected by the API and the opinions of the users about the visualised content. Otherwise, the change of the emotions would not have a pattern throughout the established feedback groups. It is also concluded that the predominant emotion during the visualisations is the neutral one. And the second most relevant is happiness. These initial results are a proper first step towards an Affective Recommender System.

6.1.5 Per Frame Emotions

The second option of the Microsoft emotion recognition service returns the emotions detected in each of the frames of the input video, i.e., there is no processing of the results before being sent.

The response, as explained earlier, contains fragments and the fragments contain events. Providing the response in this format allows the client to retrieve the response in fragments instead in a whole, and this is useful for not having to wait until the end of the entire video to be processed, facilitating real-time applications.

Although the acquisition of the emotions in real time was not applied for the prototype of the data collection platform due to time limitations, it is a considerably interesting option for the proposed system as it would allow detecting the emotions faster and then to use them for proposing the next video to the user.

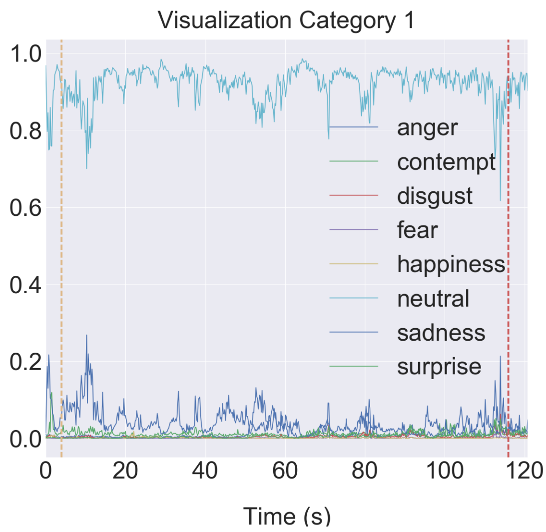

FIGURE 6.8: Emotions over time, group 1, example 1

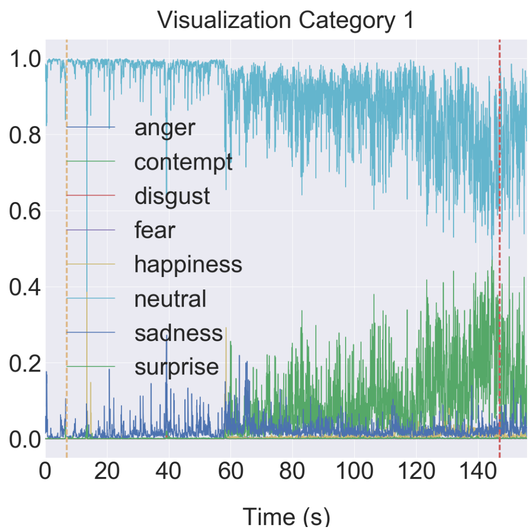

FIGURE 6.9: Emotions over time, group 1, example 2

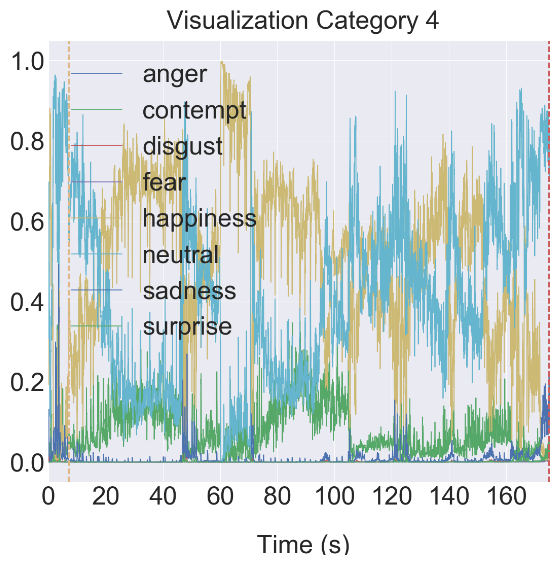

FIGURE 6.10: Emotions over time, group 4, example 1

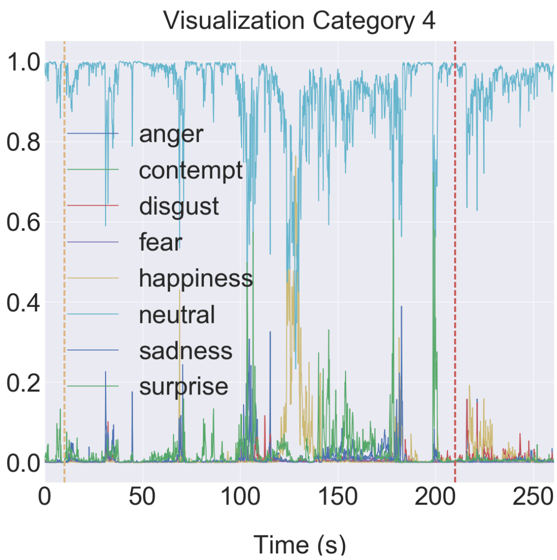

FIGURE 6.11: Emotions over time, group 4, example 2

In Figs. 6.8, 6.9, 6.10 and 6.11 (full size in appendix C)the set of obtained emotions for four visualizations are plotted. The x-axis represents the time, in seconds, and the y-axis represents the relative value of the emotions. Each emotion throughout time is represented in a different colour. The vertical dotted lines represent the instants were the YouTube video playback started (the yellow line) and ended (the red line).

It can be appreciated that the visualisation represented in Fig. 6.8 shows a lower variation in the value of the emotions, followed by the one represented in Fig. 6.11, while in the visualizations in Figs. 6.9 and 6.10 the variation is high. The changes in the emotions shown may vary for different people and videos; however, such strong difference is due to the frame rates of the recorded videos. Those videos with a higher frame rate have more information about the emotions per second, which is translated in a larger variation.

Figs. 6.8 and 6.9 represent the responses for visualizations belonging to group 1 (users who did not like the video and would not watch something similar) and figs. 6.10 and 6.11 for group 4 (users who liked the video and would watch something similar again). In the first visualizations of group 1 the predominant emotion in neutral. In the second figure of the group (6.11) it can be noticed at the end the detection of a degree of surprise.

For the visualisations in group 4, the neutral emotion is again predominant; this result confirms what had been detected in the analysis of the aggregate emotions. The visualization represented in 6.10 presents a high content of happiness. These four visualisations allow detecting in this preliminary analysis how different the results are, not only across the different groups of ratings but also within the same groups. The heterogeneity of the emotions proves that machine learning techniques and complex analysis would be needed to determine from a given set of emotions the opinion of a person on a video.

As it was explained in chapter 4, the instants when the watcher started and stopped being recorded as well as the time when the playback started and ended were gathered. The availability of this information allows to combine it with the response obtained from the API and to differentiate between the emotions in the three phases of the consumption of content exposed by [12]. As stated in chapter 3 the authors of this paper propose a framework for the behaviour of the emotions in the online world. They divided the process of in three steps: the entry stage, the consumption stage and the exit stage. The names of the steps are self-explanatory: the first one refers to the emotions shown before the visualisation, the second to the emotions detected during the visualisation and the third one to the state afterwards. As presented in the paper, the third stage might become the entry stage for the next consumption, which would happen usually in the proposed RS. Finding patterns in the three stages across them are also possibilities for determining if a person liked a video.

6.1.6 Conclusions of the emotion analysis

From this section it is learned that, in basis of a preliminary analysis, the emotions detected during the visualization of videos are an input for the generation of recommendations for entertaining videos.

It has also been observed that the emotions do not show a clear pattern according to the different user feedback, implying that for finding the patterns it will be necessary to use machine learning techniques. This fact leads to conclude that out of the two options offered by the Emotion API, the second one - *the per frame emotions retrieval* - would be more convenient for the system because it would allow more options when dealing with the emotions and finding meanings for them.

The collected data leaves the door open to a great amount of possibilities for studying the data and working with them to find out the best approach for the recommending task.

6.2 Analysis of the Adoption Model

As described in section 4.4 an UTUAT2 model was used with some slight modifications to fit the case that is being studied. The model can be found in Fig. 4.2 and the hypotheses being tested are summarized in table (further on in the section) 6.10.

Gender	Male	52.40%
	Female	47.60%
Age	30 or Below	72.60%
	Above 30	27.40%
Country	Spain	77.40%
	Other	22.60%

FIGURE 6.12: Demographic information of the respondents

As described in section 5.1, a questionnaire was distributed online for measuring the different constructs that compound the model. A multi-measure criteria was used to measure each of them except of *trust* which was measured with a single item. For the rest of the constructs either two or three items were used.

A total of 84 responses to the questionnaire were obtained. It is noted that the amount of subject that answered the questionnaire is less than the amount of video watchers. The reason is that some of the individuals who took part started doing the experiment but they did not finish it. The respondents had the demographic characteristics shown in table 6.12:

As a first approach to the results the mean, standard deviation and median values of each of the items used in the measurement were calculated and they are presented in table 6.4

The minimum and maximum values for the items are not presented because in all the cases these values were 1 and 7, respectively.

As a reminder to the reader, each of the items in the distributed questionnaire were measured using a seven point Likert Scale, with the value 1 meaning *Strongly Agree* and the value 7 meaning *Strongly Disagree*.

In table 6.4 it can be seen that the mean value in the majority of the item responses is close to four, the middle value, leading to think that there might have been many misunderstandings when answering the questionnaire and that the respondents opted to answer the *indifferent/unknowledge option*. However, when the standard deviations are checked it is found that they have a high value (between 1.5 and 2) for the used scale. This means that there has been a big range of different responses when answering the survey and not that most of the responses were close to the central value.

TABLE 6.4: Basic Statistics

Indicator	Mean	Median	Standard Deviation
PE1	2.92	2.00	1.84
PE2	3.12	3.00	1.69
EE1	2.44	2.00	1.89
EE2	2.35	2.00	1.86
SI1	3.80	4.00	1.47
SI2	3.95	4.00	1.58
SI3	3.42	3.00	1.67
FC1	2.36	2.00	1.96
FC2	2.46	2.00	1.91
FC3	3.50	4.00	1.51
HM1	3.33	3.00	1.78
HM2	3.13	3.00	1.68
HM3	3.27	3.00	1.78
H1	4.26	4.00	1.68
H2	3.89	4.00	1.69
T1	3.45	3.00	1.64
IE1	3.45	3.00	1.47
IE2	2.86	3.00	1.70
BE1	3.30	3.00	1.74
BE2	3.33	3.00	1.71
P57	3.99	4.00	1.90
P56	4.56	5.00	1.92
BI1	3.51	3.00	1.85
BI2	4.17	4.00	1.68
BI3	3.38	3.00	1.87

To verify this fact the densities of responses for each value in the Likert Scale and for each measured item have been plotted. The items measuring the same construct have been combined in the same plot. Figs. 6.13 to 6.23 display the results (bigger size in appendix D).

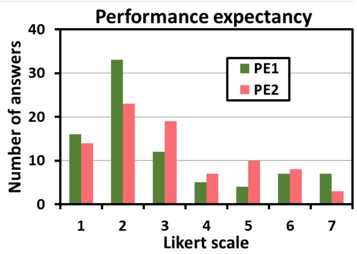

FIGURE 6.13: nº ans. per item PE

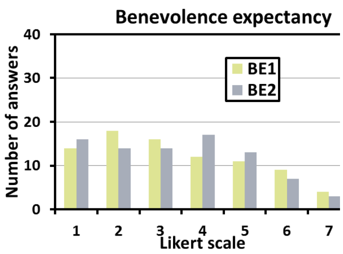

FIGURE 6.14: nº ans. per item BE

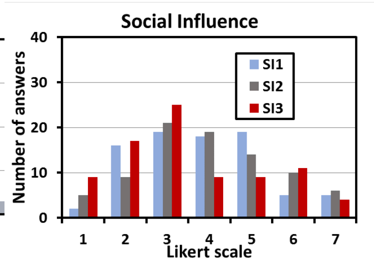

FIGURE 6.15: nº ans. per item SI

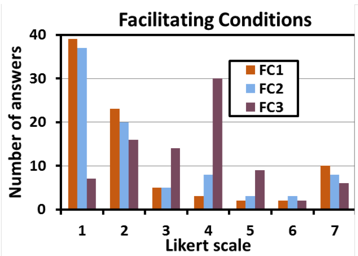

FIGURE 6.16: nº ans. per item FC

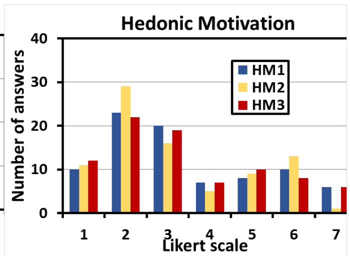

FIGURE 6.17: nº ans. per item HM

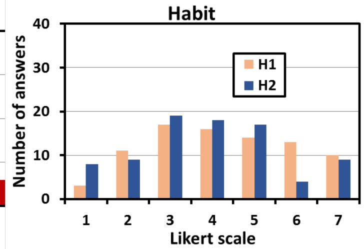

FIGURE 6.18: nº ans. per item H

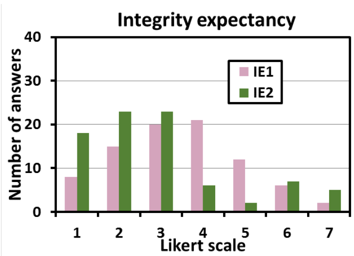

FIGURE 6.19: nº ans. per item IE

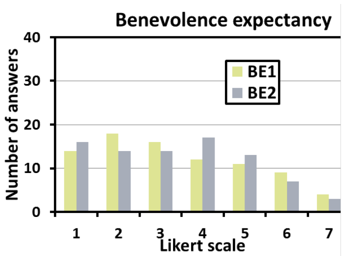

FIGURE 6.20: nº ans. per item BE

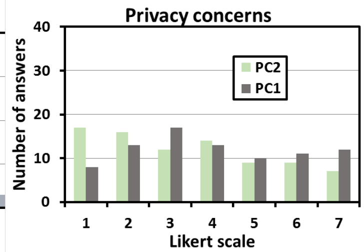

FIGURE 6.21: nº ans. per item PC

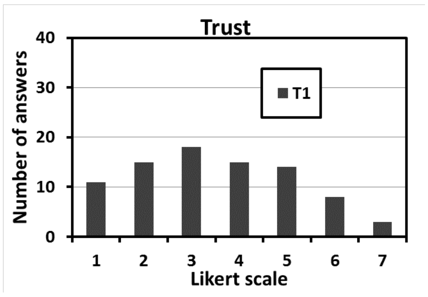

FIGURE 6.22: nº ans. per item T

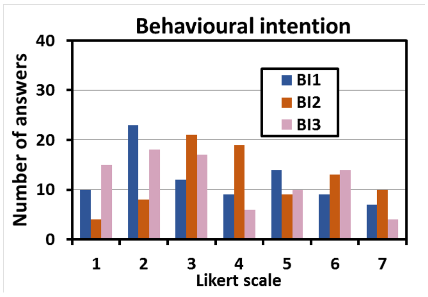

FIGURE 6.23: nº ans. per item BI

The previous affirmation is confirmed by observing the plots. It can also be stated that there is no recognisable probability distribution in the answers. Except in the case of Trust, that is measured by a single item[6], the other constructs show that there is a high correlation

[6]item makes reference to the questions meassuring each construct; the latent variables conected to trust in the model are not considered here

between the different items, as it was expected. In Fig. 6.16, it can be seen that the item FC3, shown in purple (the one in the right), differs from the other two in the distribution of answers when measuring the facilitating conditions. Something similar happens in the construct Behaviour Intention, shown in Fig. 6.23; the item in orange (the middle one) has a different distribution than the other two. This might have been due to missunderstading of the quiestions or a wrong formulation of the meassurements. As a reminder, the questions were: "*I think that I could get help from others when using the Recomotions application if I needed to*" and "*If the Recomotions application existed I would try to use it in my daily life*"

The correlations between item were calculated and are presented in Fig. D.12 in appendix D. It is shown that most the items show high correlations with the items intended to measure the same construct. There are some exceptions, such as the items measuring Privacy concerns, which show no correlation between them or with any other item. The third item for measuring facilitating conditions presents as well very low correlation with the other two items measuring that construct or with any other item. Lastly, the second item measuring Behavioural intention shows a relatively lower correlation with the rest of items, including the other two items measuring the same aspect.

The analysis of the model was done using the Structural Equation Model (SEM). This is a method typically used when the model has latent variables, i.e., the variables are unobservable and hard to be measured [126].

There are several approaches to SEM, being the best known the Covarianzed-based SEM (CB SEM) and the Partial Least Squares SEM (Partial Least Squares).

The CB SEM has been widely applied and it is the preferred method for confirming and rejecting hypotheses. However, it requires having a big sample size and that the data are normally distributed. If the collected data do not satisfy these requirements PLS SEM is commonly used as an alternative.

PLS SEM makes no assumption about the distribution of the data and it works reasonably well if the sample size is small. Moreover, this method is suitable if the theory about what is being studied is not broadly available. The studied case fulfils the three just mentioned characteristics. As inferred from the basic statistics of the questionnaire results and from the plots, the data do not present a normal distribution, the sample size is 84 and the used model has some modifications with respect to the original to make it fit to the studied RS.

The PLS SEM method focuses on the analysis of the variance of the different variables. Other researchers using the UTUAT2 model, such as the author that proposed the original model [19] or the authors in [18], use this method to obtain the results. For the computation of the model both research groups use a software called *SmartPLS* [127].

SmartPLS is a software developed by Ringle, C. M., Wende, S., and Becker, J.-M. It provides the means to compute the analysis without the need of having knowledge on coding. It provides a nice user interface and it presents the results in a very intuitive way [127].

The current research uses this software to compute the path analysis by means of PLS SEM. To compute the analysis the questionnaire data were loaded to the software as a .CVS file containing the responses for each one of the items separated with comas, as well as information about the respondent age, gender, and country. The software allows running the PLS algorithm for path analysis and to combine it with other techniques or procedures to obtain more information from the data. Previously to running the algorithm the model was defined as shown in Fig. 6.24 (image obtained from the SmartPLS user interface). The image represents how the items are connected to their latent variable and the relations between the latent variables. The PLS algorithm was run using the default values: the maximum number of interactions was set to 300 and the stop criteria $10xe^{-7}$. A bootstrapping procedure was also used for testing the statistical significance of the values obtain in the PLS analysis. Bootstraping is a method that increases the number of observations by randomly picking them from the original set of samples. As suggested in [126] the number of samples for bootstrapping was set to 5000 and the sign changes to the option "No sign changes". The option "Individual changes" was also tried for this parameter and as the results were similar the default option was left unchanged (if the results had been different the "Construct Level Changes Option" would have been used [126]). After running the PLS algorithm the amount of interactions needed until the convergent point was checked. As it is an algorithm that stops when convergence is reached or when the maximum number of interactions is reached, the stopping point is a conditioner of the quality of the estimations. In this case, the convergence was reached after five interactions, meaning that the estimations are of good quality.

The analysis is composed of two models: the outer model or measurement model, relating the observed variables with the latent construct, and the inner model or structural model that relates the latent constructs to find their relationships. The variables of the model can also be of two kinds, exogenous or endogenous. The exogenous variables are those that their effect on the endogenous ones is studied.

As a first instance, the measurement model was analysed. The items used in this study are reflective, which means that the items measuring same construct are highly correlated and interchangeable. A different case would be to have formative items, meaning that there is no correlation needed between the items. For making it more illustrative for the reader an example is given: when measuring customer satisfaction in a restaurant, different aspects can be used to infer a value, such as food quality, waiting time, or price. The three aspects are highly correlated and it is difficult to determine them individually, therefore this is an example of reflective measurements. If the study were to discover the causes of bad results at university, which is a difficult thing to measure directly, different aspects should be considered such as bad environment at home, difficulty understanding the content or demotivation with the studies, aspects that do not necessary imply a relationship between them.

As in the researched case the items are reflective, it is necessary to verify their reliability and validity. For this purpose, different statistical parameters, calculated by the software, are used.

The verification of the validity is done following the approach presented in [126]. The used data are presented in table 6.5.

The **Indicator Reliability** is checked as a first step. This is done by considering the squared value of the outer loadings, a value that measures the correlation between the item and the

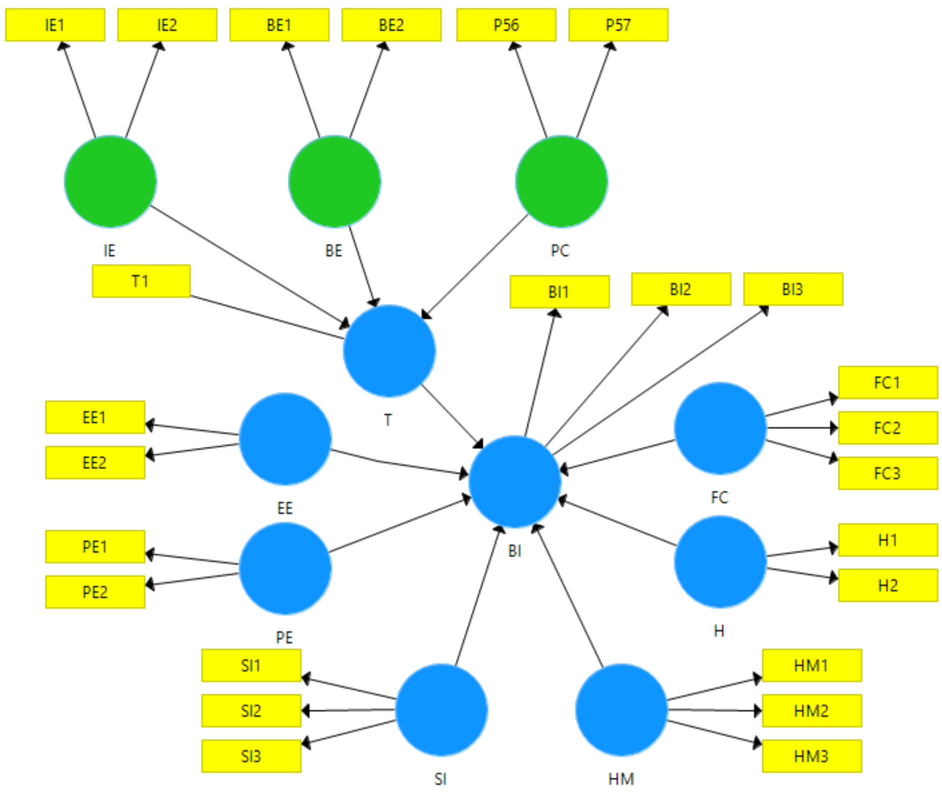

FIGURE 6.24: Model in SmartPLS

correspondent latent variable. As stated in [126] this parameter should be greater than 0.7 to assure reliability. In the paper, it is also said that for exploratory research it would be acceptable if the values were greater than 0.4. Table 6.5 shows that the majority of the values are above the recommended 0.7. However, five out of the twenty-five items do not attain this criterion: SI1, SI2, FC3, PC2 and BI2, being FC3 that with the lower value.

The next parameter to be examined is the **Internal Consistence Reliability**. This measures whether the parameter used to determine the value of a construct can be trusted or not. The normal approach is to measure this characteristic with the "Cronbach's alpha"; however, this practice is not recommended for the chosen analysis, being the composite reliability a better option [126]. As reflected in table 6.5 all the constructs are over the 0.7 recommended value, being the Privacy Concerns the one with the lower value.

Afterward, the validity (i.e., how well they represent the reality) of the items is tested. For that the Average Variance Extracted (AVE) is used in two different ways. For each construct, it is checked that the value is greater than the recommended 0.5 and that it is greater than the correlation of the construct with the others. The AVE indicates the amount of variance captured by the construct in relation to the amount of variance caused by errors in the measurement.

In the first place, AVE is used to measure convergent validity, which determines to what degree two items that are supposed to measure coincide in the results. As shown in table 6.5 all of the values are above 0.5; Performance Expectancy and Effort Expectancy have values

Table 6.5: Reliability and Validity

Reliability and validity	Item	Outer Loadings	Indicator Reliability	Composite Reliability	AVE
Performance Expectancy	PE1	0.97	0.93	0.97	0.94
	PE2	0.97	0.94		
Effort Expectancy	EE1	0.98	0.96	0.98	0.96
	EE2	0.98	0.96		
Social Influence	SI1	0.80	0.64	0.86	0.68
	SI2	0.78	0.61		
	SI3	0.89	0.78		
Facilitating Conditions	FC1	0.92	0.85	0.86	0.68
	FC2	0.91	0.83		
	FC3	0.59	0.35		
Hedonic Motivation	HM1	0.93	0.86	0.96	0.88
	HM2	0.94	0.88		
	HM3	0.96	0.91		
Habit	H1	0.89	0.79	0.91	0.84
	H2	0.94	0.89		
Trust	T1	1.00	1.00	1.00	1.00
Intregrity Expectancy	IE1	0.89	0.80	0.88	0.79
	IE2	0.89	0.79		
Benevolence Expectancy	BE1	0.92	0.85	0.89	0.80
	BE2	0.87	0.75		
Privacy Concerns	PC1	0.86	0.74	0.82	0.69
	PC2	0.80	0.64		
Behavioural Intention	BI1	0.93	0.87	0.92	0.79
	BI2	0.80	0.64		
	BI3	0.93	0.87		

close to one. The lowest values are shown by Social Influence, Facilitating Conditions and Privacy Concerns.

The second assessment done with the AVE is the Discriminant Validity, which measures to what extent variables that are not supposed to be related do not show relations. This is done, as said earlier, by comparing the squared values of AVE with the correlations with the other constructs. The results are shown in table 6.6. It is observed that some of the constructs are not discriminant valid. The ones that fulfil the requisite are Effort Expectancy and Privacy concerns; the rest of them show some value of the correlation that is higher than the AVE, even if the difference is not too big.

Up to this point it can be summarized that the findings show an item with very low reliability and issues in the discriminant validity of the majority of measures. It was decided to discard the measures that did not show the appropriate *Item Reliability*, being aware of the possible loss of information by taking this decision. The new results of the stated measures are shown in tables 6.7 and 6.8.

TABLE 6.6: Discriminant Validity

Discriminant Validity	**BE**	**BI**	**EE**	**FC**	**H**	**HM**	**IE**	**PC**	**PE**	**SI**	**T**
Benevolence Expectancy	**0.64**										
Behaviour Intention	0.58	**0.63**									
Effort Expectancy	0.46	0.56	**0.92**								
Facilitating Conditions	0.46	0.51	0.90	**0.46**							
Habit	0.47	0.75	0.27	0.19	**0.71**						
Hedonic Motivation	0.66	0.84	0.72	0.66	0.62	**0.78**					
Integrancy Expectancy	0.52	0.63	0.80	0.76	0.33	0.71	**0.63**				
Privacy Concerns	0.12	-0.02	-0.20	-0.30	0.02	-0.09	-0.13	**0.48**			
Performance Expectancy	0.59	0.78	0.75	0.68	0.54	0.89	0.79	-0.09	**0.87**		
Social Influence	0.61	0.78	0.50	0.55	0.66	0.76	0.58	-0.05	0.69	**0.46**	
Trust	0.72	0.61	0.37	0.28	0.57	0.60	0.46	0.19	0.60	0.53	**1**

TABLE 6.7: Reliability and Validity without the rejected items

Reliability and validity	**Item**	**Outer Loadings**	**Indicator Reliability**	**Composite Reliability**	**AVE**
Performance Expectancy	PE1	0.97	0.93	0.97	0.94
	PE2	0.97	0.94		
Effort Expectancy	EE1	0.98	0.96	0.98	0.96
	EE2	0.98	0.96		
Social Influence	SI3	1.00	1.00	1.00	1.00
Facilitating Conditions	FC1	0.98	0.95	0.97	0.95
	FC2	0.97	0.94		
Hedonic Motivation	HM1	0.93	0.86	0.96	0.88
	HM2	0.94	0.87		
	HM3	0.96	0.91		
Habit	H1	0.88	0.78	0.91	0.84
	H2	0.95	0.90		
Trust	T1	1.00	1.00	1.00	1.00
Intregrity Expectancy	IE1	0.89	0.80	0.88	0.79
	IE2	0.89	0.79		
Benevolence Expectancy	BE1	0.92	0.85	0.89	0.80
	BE2	0.87	0.75		
Privacy Concerns	PC1	1.00	1.00	1.00	1.00
Behavioural Intention	BI1	0.96	0.92	0.96	0.92
	BI3	0.96	0.92		

TABLE 6.8: Discriminant Validity without rejected items

Discriminant Validity	BE	BI	EE	FC	H	HM	IE	PC	PE	SI	T
Benevolence Expectancy	**0.64**										
Behaviour Intention	0.62	**0.85**									
Effort Expectancy	0.46	0.61	**0.92**								
Facilitating Conditions	0.46	0.53	0.93	**0.90**							
Habit	0.48	0.69	0.28	0.15	**0.71**						
Hedonic Motivation	0.66	0.88	0.72	0.66	0.62	**0.78**					
Integrancy Expectancy	0.52	0.66	0.80	0.75	0.33	0.71	**0.63**				
Privacy Concerns	0.05	-0.12	-0.29	-0.27	-0.02	-0.23	-0.21	**1.00**			
Performance Expectancy	0.59	0.82	0.75	0.68	0.54	0.89	0.79	-0.20	**0.87**		
Social Influence	0.55	0.80	0.63	0.56	0.62	0.77	0.65	-0.14	0.74	**1.00**	
Trust	0.72	0.61	0.37	0.32	0.57	0.60	0.46	0.17	0.60	0.47	**1**

Both tables show how the rejection has improved all the Reliability and Validity metrics. Using this new model all the values in 6.7 fulfil the recommended value and the number of values in 6.8 that do not fulfil it has decreased. Moreover, in the majority of the cases the values that do not fulfil the criterion are slightly over the AVE squared, in concordance to what is done in [68] this items are not removed from the study.

Once the Reliability and Validity have been examined, the analysis can be moved forward to the structural model. The structural model finds the relationships between the latent variables. The obtained values for the path coefficients and the corresponding p-values, as well as the R^2 values are presented in Fig. 6.25

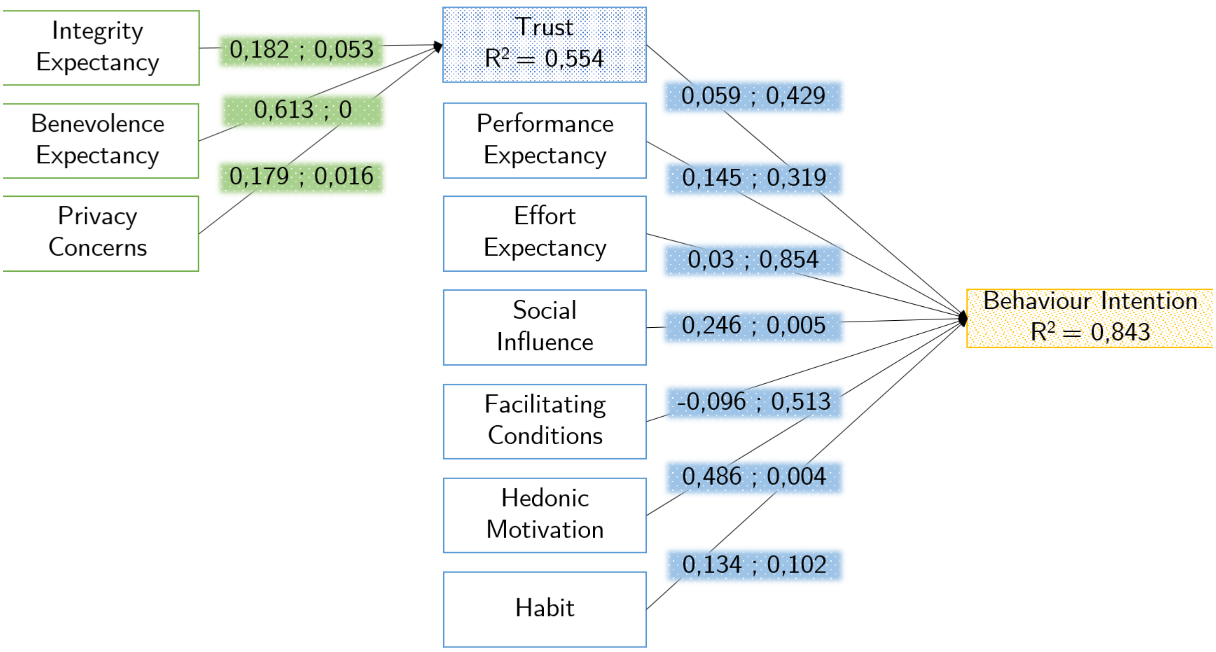

FIGURE 6.25: Results of the UTAUT2 analysis

The coefficient of determination, or R^2, measures the amount of the variance of the endogenous variable that is produced by the variance in its related exogenous variance. In the model, it can be seen that the value obtained for Trust is $R^2 = 0.554$, which means that the three constructs (Integrity Expectancy, the Benevolence Expectancy and the Privacy concerns) can explain more than one half (55.4%) of the variance in Trust, meaning that appart from the considered ones, there are more factors conditioning the adoption of an affective RS. For Benevolence Expectancy a value of R^2 =0.843 was obtained. Once again, it can be inferred that the 84.3 % of the change produced in the Behaviour Intention is caused by the factors that had been hypothesised.

The path coefficient, showed in Fig. 6.25 as the first number on the line connecting constructs, is a measure of the causal relation between the two variables. It is relative to the other independent variables affecting the dependent one, which makes the comparison possible. The obtained results show that, as it had been hypothesised, the three constructs chosen do have an effect on the trust on the system. It is also extracted from the results that out of the three the one that has the biggest effect on a user trusting an Affective Recommender System is the *benevolence expectancy*, in other words, the extent to which the users believe that the system would use their information only for the stated purposes.

Moving forward to the path coefficients of the latent variables producing an effect in the variance of the Behaviour Intention, it can be seen that the values corresponding to Trust, Effort Expectancy and Facilitating Conditions are lower than 0.1 (0.059, 0.04 and 0.096, respectively). The common practice in this analysis is to consider the path coefficients which absolute value is lower than 0.1 not significant[7] [54, 68, 70]. Therefore, this results show that Trust, Effort Expectancy and Facilitating Conditions are not significant for the intention of adoption of an affective RS.

The p-values (the second value over the lines connecting constructs) are the main metric used to determine the statistical significance of the results and are obtained from the bootstrapping analysis. The statistical significance represents the probability of accepting the null hypothesis when the not null happened. In other words, if the significance is 5% it means that for one out of 20 observations the prediction is wrong. The p-value for obtaining such significance is 0.05 or lower; a p-value of 0.1 or lower represents a significance of 10%. The obtained p-values confirm what had already been induced from the path coefficients: the constructs Trust, Effort Expectancy and Facilitating Conditions are not significant for the behavioural intention. The Performance Expectancy path shows a high p-value as well: 0.319, which means that its significance is also low, although not as low as the previous three. For the other constructs, it can be inferred that the one having a higher impact in the Behavioural Intention is the Hedonic Motivation, followed by the Social Influences, the Performance Expectancy [8], and lastly, the Habit.

Once the analysis was done the data were segmented in subsamples groups for comparing the results for different demographic groups. Even though the model does not contain moderating conditions it is interesting to detect possible changes in the demographic groups.

[7]The conceptual definition of statistical significance is the probability that the cause of the relation between two variables is something different than randomness.

[8]Reminder to the reader of the low significance of the measurement

TABLE 6.9: Segmentation Results

Segmentation	General		Gender				Age				Country			
			Male		*Female*		*Age ≤ 30*		*Age > 30*		*Spain*		*Other*	
R^2 *(Trust)*	*0.55*		*0.65*		*0.83*		*0.83*		*0.93*		*0.88*		*0.92*	
	path coe.	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*
IE → T	0.18	0.06	0.13	0.21	0.26	0.14	0.18	0.13	0.19	na	0.25	0.03	0.24	0.44
BE → T	0.61	0.00	0.67	0.00	0.52	0.00	0.62	0.00	0.70	na	0.59	0.00	0.60	0.00
PC → T	0.18	0.01	0.20	0.06	0.14	0.28	0.28	0.00	-0.06	na	0.15	0.07	0.35	0.04
R^2	0.84		0.90		0.48		0.59		0.59		0.58		0.64	
(Behavioural I.)	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*
T → BI	0.06	0.43	0.24	0.00	-0.04	0.76	0.01	0.96	0.09	na	-0.07	0.36	0.22	0.92
PE → BI	0.15	0.32	0.12	0.37	0.18	0.52	0.26	0.21	0.16	na	0.02	0.86	0.41	0.84
EE → BI	0.03	0.85	0.31	0.09	-0.47	0.13	0.07	0.77	-0.16	na	-0.03	0.85	0.05	0.93
SI → BI	0.25	0.00	0.20	0.18	0.40	0.01	0.16	0.17	0.01	na	0.17	0.04	0.34	0.91
FC → BI	-0.10	0.51	-0.32	0.12	0.22	0.35	-0.16	0.43	0.17	na	-0.09	0.60	0.02	0.99
HM → BI	0.49	0.00	0.44	0.03	0.56	0.04	0.47	0.03	0.00	na	0.86	0.00	-0.11	0.88
H → BI	0.13	0.10	0.05	0.65	0.15	0.26	0.17	0.11	0.47	na	0.09	0.30	0.22	0.70

Table 6.9 (bigger in appendix D) shows the obtained results for the subsamples. The segmentation was done in terms of gender (male, female), age (30 or less, over 30) and country (Spain, other).

It the table some interesting facts can be observed. The coefficient of determination for the Trust construct is substantially bigger when the data are segmented; however the opposite happens with the same value for the Behavioural Intention construct, that decreases with the segmentation. The values with low statistical significance have been marked in blue and orange. The values in blue correspond to a low significance according to the path coeficient and in orange accoriding to the p-values.

Regarding Trust, the Behavioural Intention remains as the variable with the biggest influence. It is noticeable that the Privacy Concerns gain more weight in several of the segmentations, especially in those who are not from Spain and in those 30 years old or younger.

Regarding the constructs affecting the Behavioural Intention, drawing conclusions is complicated since the statistical significance is low for most of them. However, it is interesting to notice that for women the Hedonistic Motivation is more relevant than for men, as well as the Social Influence.

To conclude the section of the analysis of the Adoption Model the proposed hypotheses will be reviewed and accepted or rejected. For commodity of the reader the hypotheses are gathered in table 6.10, withe the correspondent conclusion about them.

As it was stated during the section, the results were not adequate to study the effect of Trust, Effort Expectancy and Facilitating Conditions on the Behavioural intention, which means that hypotheses H1, H3 and H5 are not supported.

Regarding the influences in trusting the system, hypotheses H1a, H1b and H1c are confirmed, adding that the most relevant of the three is the Benevolence Expectancy. It can also be added that gender, age and country of origin are moderators for the Privacy Concerns variable.

Table 6.10: Hypotheses

H1	Trusting the system has a positive impact on the intention of adoption.	Rejected
H1a	A positive integrity expectancy has a positive impact on the trust of the system.	Accepted
H1b	A positive benevolence expectancy has a positive impact on the trust of the system.	Accepted
H1c	Privacy concerns have a negative impact in trusting the system.	Accepted
H2	The performance expectancy has a positive effect on the intention of adoption of an Affective Recommender System	Accepted
H3	A low effort expectancy has a positive influence on the intention of adoption of an Affective Recommender System.	Rejected
H4	The social influence will have a positive impact on the intention to adopt the system.	Accepted
H5	The facilitating conditions will have a positive impact on the intention to adopt.	Rejected
H6	A positive hedonic motivation has a high impact on the adoption intention of an Affective Recommender system.	Accepted
H7	Habit will have a positive impact on the adoption intention.	Accepted

H2, H4, H6 and H7 are also be accepted after the analysis of the data, being the Hedonic Motivation the construct with the higher impact in the variance of the user intention to adopt, followed by the Social Influence, the Performance Expectancy and the Habit.

It can also be concluded from the conducted research that some of the items chosen for measuring the model are not appropriate for this kind of RS or for this target group. Those items are: SI1, SI2, FC3, PC2 and BI2. These items were omitted in the model for conducting the analysis.

6.3 Recommending Options

Considering the amount of collected data and the recommender problem exposed in chapter 4, there are many possible ways in which the recommendations could be computed and the data be used and interpreted.

6.3.1 Presentation of the Recommendations

One of the points to consider is how the outcome of the recommending engine should be. As explained earlier in the report chapter 3 the outcome of the recommender engine can be a single item, a collection of items that combine together, a set of top N items or a sequence of them, among others; a full list of the options can be found in [5, p. 390]. Given the recommending task studied in this research the two options to be considered are recommending a single item or a sequence. The idea is to present videos playing one after each other, which eliminates other options such as top-N items or collections of similar videos.

There is little literature in sequence recommendations [5, p. 337]; however, in the industry, some of the main players in the RS environment are making use of this kind of recommendations, such as Spotify and YouTube. The difference between the Top item recommendations and sequence recommendations is that in the second case the items before and after are taken into account.

According to J. Masthoff in [5, c. 21] in a sequence of videos, the sequence order matters for the satisfaction of the consumers. From this affirmation, it can be argued that the sequential approach would be the most appropriate one for the proposed recommender task. It is also stated in the same chapter that the affective state shown in the consumption of an item conditions the following one, which indicates that the recommended sequence should be adjusted each time a video is visualized. This supports the previous conclusion in section 6.1 in which it was stated that the per frame emotions solution would be more appropriate for the system. As claimed in that section this option allows retrieving the emotions faster by requesting them by frame instead of waiting to the whole video to be processed.

The same author suggests to group similar content together (i.e., if in the recommender sequence there are two videos related with the same topic to reproduce one after the other). Moreover, it is suggested to be consistent with the mood of the videos player; it recommends not playing a sad video in between two happy ones. This aspect is not as relevant for the proposed RS because all the videos are expected to have a positive mood, however it could be taken into account in certain approaches since the positive mood shown could vary from video to video (i.e., a video might be entertaining by making a person laugh or by explaining an interesting concept). Finally, it is also proposed in [5, c. 21] that the last video of the sequence should be the one of higher relevance to the user. This idea could only be used if the amount of videos in the sequence is known to the system - parameter to consider in the design of the system.

6.3.2 Preprocessing of the Data and the Recommending Engine

A crucial part of any RS is the recommendations engine, which is the part in charge of generating recommendations for a given input. Traditional RSs take as input the ratings from the users, and use those ratings to predict how the users would rate other products and recommend those with the higher predictions. Therefore, when explicit feedback techniques are used, the pre-processing of the data is not as important, however, when it comes to implicit feedback, more processing is needed to transform the data into a possible input to a recommender algorithm. The input for the proposed RS is a complex form of implicit feedback. As explained in section 6.1 the acquired emotions for each video are either a set of averages of eight emotions for video fragments or the eight emotions over time for each of the frames of the video. This results in having a lot of information for each video that should be processed before being able to generate any recommendation. As remarked during the chapter, dealing with the emotions is a machine learning problem, and as any machine learning problem it would need pre-processing of the data for afterwards training the system and finishing with its evaluation. The process should be an iterative process in which the pre-processing part consists on finding features from the data that could be used as inputs to the system. The training of the system consists on the division of the data in training, validation and testing sets. The first set is used to train a system with an algorithm, then the validation set is used to choose the most adequate parameters, and, finally, the test set is used in the evaluation step to assess the quality of the recommendations.

It has been seen in the literature of RSs taking into account emotions that the main focus is the translation of the detected emotions into data that can be used as input to the learning model. Two different approches, reprsented in Fig. 6.26 have been published in the literature for this purpose.

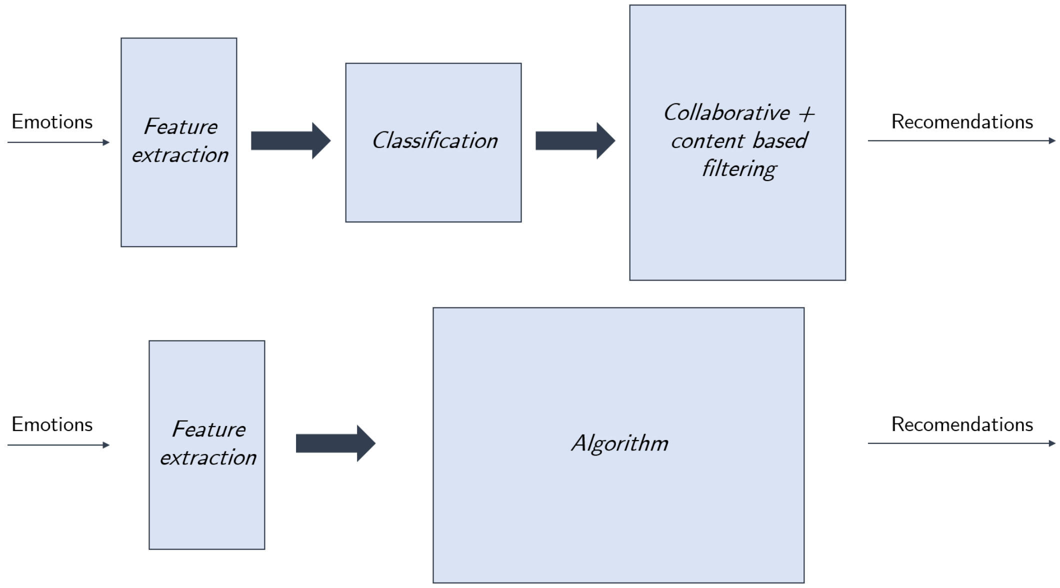

FIGURE 6.26: Two approaches for generating the recommendations

In both approaches, the first step would be to extract features from the obtained data. This step is considered to be important from this problem because there is a lot of information available for each visualization and for any data mining problem, it is necessary to have a set of features. There are several options for the creation of this features. Most of the work done on the topic take the time out of the information and use some measure for the overall consumption of the content. Some examples of other approaches taken in research that also detects emotions periodically during the interaction time follow. In [128] the emotions are simplified by representing them as a probability of being positive or negative during the overall time of the consumption. Another approach is the one taken in [34], where the mean and the variance values of emotions are taken as the features for the machine learning problem. In the work presented in [36] the most significant emotion is taken as the relevant one, and all the emotions shown during the consumption period are taken as the input. Some other option for removing the time could be transforming the data for each video to frequency through a Fast Fourier Transform (FFT) algorithm, as done in [129], and then use the intensity of frequency bands for each emotion as features. Another option, although it has not been seen in the literature of this field, would be to maintain the time. This would imply a multi-dimensionality problem, which could be treated with dimensonality reduction techniques, such as Single Value Decomposition [5, c. 2].

The difference between the two approaches is the step after the feature extraction. In the first approach, the researches translate the implicitly detected emotions into ratings. This could be done in several ways. For instance, the authors in [45] use whether the emotion is negative or positive as a binary rating. In [128], the authors set some intervals and thresholds and convert the probability of a positive or negative emotion into a rating from 1 to 5. A similar approach could be taken in the case of the discussed RS for entertaining videos by using some classification technique to transform the emotions into ratings. It would be possible to take this approach since the dataset contains users' explicit feedback, which could be used for training a supervised classification problem. Further research should be done to prove that this classification would be feasible.

Once the ratings are obtained, they could be used as the input to a traditional recommender method. As explained in section 3.1, the traditional recommender approaches are collaborative filtering and content based filtering. A common procedure is to combine this two techniques in an hybrid approach. The data collection was designed with the target of testing the data on an hybrid system, combining content based and collaborative filtering approaches. Nonetheless, other traditional approaches could be taken on the collected data set.

Regarding the CB approach, the idea in the design of the data collection was to do it in a simple way as a proof of concept, by clustering the videos in the categories explained in 5.3. However, different approaches could be taken by using the metadata available in the YouTube video (common for all of them), such as title or length. Another procedure including the emotions in the content based module could be to take a similar approach to what was done in [45], where they determine the genre of a movie trailer by the emotions shown during the visualization.

Regarding the CF approach, two options should be chosen. As explained in 3.1, it can be model based or memory base and user base or item based.

For the suggested system, the choice between model based and memory based is a questionable topic. A model based approach would be a good option, since, as stated in 6.3.1, the system should compute the recommendations every time a video is watched and model based CF conducts of the computations offline, and performs better in real time. However, in a memory based approach, which computes all the calculations each time a new item is recommended, implying that it considers the latest information gathered from the user.

Regarding the choice between item-based or user-based, it is hypothesized, needing further confirmation, that an item-based approach would be a better option for this system. The reason for it is that the emotions shown during the visualization depend on the personality of the user [130, 131].This means that if two people like the same video they do not necessarily express the same emotions. A user-based approach predicts the ratings by finding similar users; this would compare the *emotions*, or ratings induce from emotions, from one user to another in the same video. On the other hand, an item-based approach compares similar items, i.e., it would compare the *emotions* shown by the same user on different videos.

Therefore, using an item-based algorithm would remove the emotional differences which might be presented between users.

The second approach represented in Fig. 6.26 is taken by other researchers and implies directly introducing the extracted features to a more complex recommender algorithm (skipping the step of translating them into ratings). This approach is taken in [34, 131–133]. However, further discussion on this aspect is left out of this research.

From this section it is extracted that the dataset opens the possibility to move forward in the research and find what solutions are the most adequate for an affective RS recommending entertaining videos. As is stated, more research is needed to cover this topic and confirm the initial suppositions.

Chapter 7

Discussion

This chapter presents a discussion about the work done. It starts by recapitulating the purpose, achievements and findings of the research. It continues by relating the conducted work with previous research on the topic. Afterwards, it moves on to state the limitations it might present. And it ends with presenting a set of implications and possible future work.

7.1 Recapitulation

The purpose of this research was to investigate the use of automatically detected emotions through facial expression in Recommender Systems; a topic that until this moment has not deeply been studied and that has a big potential. As the recommender content is an important factor in the design of a recommender system, the research problem was limited to recommending videos with an entertaining purpose.

The concept of affective RS was studied from two perspectives: sociological and technical. Regarding the first one, the possible adoption of an RS including emotions detected through facial expression was studied in terms of behavioural intention. From a literature research, it was determined that the Unified Theory of Acceptance and Use of Technology 2 (UTAUT2) was an appropriate model to study the adoption of this kind of systems, as it had been previously used successfully in the study of RS [18, 68]. The model was modified to drop the construct Price and to add Trust, an important factor regarding affective computing [69]. Three new constructs were added to study the factors conditioning the trust in an affective RS.

Respecting the technical aspects, it was found in the early stages of the research that the lack of datasets containing emotions detected during the visualisation of content was a relevant factor conditioning the development of an affective RS. Only one dataset containing users' emotive reactions when presented images was found [33]. Such dataset could be used for the research of a video affective RS; however, it would imply a lot of assumptions since the beginning. Therefore, it was decided to focus the thesis in the generation of a dataset which could open the possibilities for more research on the topic.

Considering that RS are online systems, a web platform was built for carrying out the data collection. A compilation of YouTube videos was gathered, creating a simple classification on topics that could be used to implement a basic content based recommender algorithm. The platform was designed and implemented with the purpose of the collection of the data and that it (the platform) could be used subsequently as the basis of an RS. For the detection of the emotions, the Microsoft Cognitive Services Emotion API was used. The platform was implemented using python's web server framework Django. The functionalities in the front end, such as managing the recording of the user during the visualisations, were implemented with HTML5 and JavaScript.

The address of the platform was distributed and 91 users participated in the experiment. They were asked to watch at least three short videos (maximum four minutes) while being recorded to process their emotions afterwards. The web platform also provided the participants with a questionnaire for measuring the constructs conforming the adoption model.

The questionnaire, composed of 25 questions using a 7 point Likert scale, was answered by 85 participants; not everyone taking part in the experiment reached the questionnaire, presented after the three visualisations of the videos. After executing a PLS-SEM analysis on the answers, it was determined that five of the items (questions) used for the measurement were not reliable, therefore they were discarded from the analysis, concluding those five questions were not appropriate to study the adoption of an affective RS. The findings suggested that the *hedonic motivation* is the factor that has the higher influence in the intention to adopt the system. The fact was not a surprise since the purpose of the studied system is to entertain the users. It is coherent that if the users perceive they would have fun or enjoy the use of the system they would use it. The following factor influencing the adoption was found to be *Social Influence*, i.e., the perception the user has on what the other think about using the system; suggesting that it could be an interesting option to add a social component to this kind of systems. Three of the considered constructs turned out to be not significant for the intention of adoption of an affective RS: trust, effort expectancy and facilitating conditions. The chosen model for the study, UTAUT2, was validated as appropriate for the study of the adoption of these systems, resulting in that the constructs influence the 84% of the intention to adopt. Regarding the factors influencing the trust in an affective recommender system, the benevolence expectancy, or the certainty that the system would use the emotions only with the stated purpose - creating recommendations - is the factor having the biggest influence with a marked difference; suggesting that when designing such a system, it would be important to make clear for the user that the emotions are only used for this purpose.

The data collection ended with a corpus composed by 435 visualisations of 282 different videos watched by 91 users. Each visualisation is presented with information about the emotions in the three phases defined by the framework presented in [12]. Moreover, for each visualisation, there is information about the feedback provided by the user in the form of binary answers to two questions: "did you like the video" and "would you watch something similar again". This dataset allows numerous research options as it contains the detected emotions for each frame of the video recorded while the user visualised content.

A first approach for analysing the data is presented in chapter 6. From this analysis, it can be seen that there is a correlation between the emotions shown and the feedback given by the user. For the first time, or at least regarding the literature found during the development of this work, these results demonstrate that the emotions detected during a visualisation of entertaining videos can be correlated with the feedback of the users. This implies that an RS based on the detected emotions is feasible. The most present emotion throughout the visualizations is the *neutral* followed by *happiness*. It was also extracted from this analysis that the pattern of the emotions detected is very different among visualisations, even when the feedback by the user was the same. This implies that machine learning techniques would be needed to process the data if the emotions are meant to be used as the feedback for the user. Some options for generating the recommendations were suggested at the end of the data analysis chapter (6). As a first premise, the system should be designed using a modular architecture, to allow trying different combinations and modifying components without changing the whole system. It was suggested that for the proposed system, the best approach would be to use a recommending algorithm producing a sequence of recommended items and that the computation of the recommendations should be done in real time since the affection in visualisation would condition the watcher reaction to the following visualisation. It was also suggested to use a classifier that would classify the visualisations in different levels, and afterwards using the ratings as the input to a traditional hybrid system, combining collaborative and content-based filtering. The collaborative filtering approach was argued to be more appropriate if it was an item based, needing further research to confirm if.

7.2 Relation to previous research

Studying behavioural intention as the indicator of how actual behaviour would be has been done previously in other research works in RS [18, 51, 53, 57, 60, 73]

In concordance with the findings from [54], it was found that the Hedonic Motivation has the more significant effect on the intention to adopt an Affective RS. As stated in [70] the proposed system is a pleasure-oriented system, which implies that the perceived enjoyment is a more relevant predictor than the perceived usefulness. Contrary to the results in [54], Hedonic Motivation was more relevant for women than for men.

Besides Hedonic Motivation, Social Influence, Performance Expectancy and Habit were found to be significant for the intention of adoption of the studied system. These findings are supported by the findings in the original UTAUT2 model [54].

The low significance of the Trust construct differs with the findings on [73] and [51]. The two mentioned studies investigate RS, however, there is not an affective component in them. However, the fact that the respondents of the research knew the researcher might have conditioned the results on this construct. The experiment explanations were approached in the first person, which, if the respondents did not adopt a high level of abstraction might have lead them to think that if the system was developed by a person they knew, they would trust it.

The irrelevance of the Effort Expectancy also contrasts with previous research in which it was found to be an influencing construct [54]. The low relevance might be due to the fact that all the respondents considered the effort expectancy very low.

The low relevance of the Facilitating Conditions construct has also been proved in other studies [18]. It is possible that it is related to the perception that the system is easy to use.

Regarding the constructs affecting trust, the three results are aligned with the findings in [69]. It can be said that Benevolence Expectancy, Integrity Expectancy and Privacy Concerns are three factors conditioning more than half of the variation of the trust on an affective RS. Benevolence Expectancy, or the perception that the system would only use the emotional information with the purpose it is collected, is the most relevant one.

In the technical aspect, the data collection was successfully conducted through an online platform, as done for obtaining the LDOS-CoMoDa dataset [63]. A dataset of emotional information was created, similar to the work presented in [33], although in the case of this Master's Thesis, only the detected emotions were saves, not the recorded video of the participants.

The emotion found to be more present in the videos was the neutral one, not surprising since it is the normal state of the human face. The emotions that followed, in order of relevance were: happiness, sadness, contempt and surprise. The presence of happiness, sadness and surprise correlates with the findings in [39], were the emotions shown in the comments for YouTube videos are studied.

The fact that the emotions correlate with the user feedback goes in concordance with the results obtained in [34] were the emotions were found to improve the performance of an RS for images.

7.3 Limitations

Even though the initial purpose of this Master's Thesis was to design and implement a prototype of an RS using emotions as feedback, it was found out in early stages of the research that there was no available data that could be used for the purpose. However, this kind of research needs data to prove that the design does it function, therefore, it was decided to change the focus and create such a dataset, not being able to design and implement the RS due to the time limitations. The platform developed to acquire the dataset can be considered as the first step towards the Affective Recommender System, as it is able to provide part of the functions of the future RS.

Regarding the dataset, some considerations should be taken into account:

- The emotions depend on the person, not everyone reacts in the same way to the same stimulus. A factor that could influence the results when working with the data
- The chosen video catalogue was composed by content in English, despite a high number of the participants were Spanish. It is possible that a Spanish speaking person had to visualise an English video and that he did not like it because he or she could not understand the language. This fact might have created some noise to the data and should be taken into account.

Another factor to have into account is the demography of the respondents. Although the study intended to be representative of the whole population of internet users, it was decided to use a convenience approach for finding the subjects of the experiment. The lack of selection and filtering of the subjects might have caused that the sample is not representative of the whole internet users as the amount of Spanish people and young people outnumbers other nationalities and ages, respectively. Moreover, the most of the respondents were gathered by distributing the web page through social networks (Whatsapp and Facebook), which implied that most of the respondents had some social network connection with the author. This might have biased the results of the questionnaire by their knowledge it was a Master's Thesis study and of the person that was behind of the system.

Moreover, the study was done on an affective RS for entertaining videos. Some of the results are conditioned by this factor, such as the Hedonic motivation. Therefore, the results cannot be generalised to any affective recommender system.

7.4 Implications

This study appears to support the argument presented in [33], claiming that the emotions can be used as the source of feedback for determining whether a person likes a video or not.

A dataset was gathered that could be used for future research on the topic, allowing to learn more about the relation of emotion and RSs and to the possibility of incorporating affective information in such systems.

Regarding the findings on the adoption, from the theoretical perspective, it has been proved that the UTAUT2 model can be applied to the study of the adoption, having the studied constructs an 84% on the influence on the intention to adopt. From a practical perspective, these findings could be applied in the design of affective RSs, a concept that has not been developed yet but that has a promising future.

7.5 Future Work

As mentioned previously, the collected data paves the way to an investigation on how to translate the detected emotions into recommendations. As suggestions for future work the following indications are given:

- The most direct future work is the analysis of the data and trying different combinations of algorithms to generate the recommendations. Different algorithms should be tried and evaluated offline to find the best combinations of data preprocessing and recommendation generation.
- Building up from the data collected in this thesis and further research on what combinations of algorithms and techniques would work, a system could be implemented and tested online.

- Once such a system is built, its actual adoption could be studied, to confirm the results obtained previously.
- Similar datasets could be created, changing some of the variables. A suggestion would be to create a similar dataset but for groups of people or for a different kind of videos.
- The adoption is a significant factor conditioning the success of a technology but is not the only one. Further research could be done in this area to study the new proposed technology.
- The research on the adoption could be extended to other affective recommender systems.

Chapter 8

Conclusion

The Master's Thesis investigated the combination of two already existing technologies: Recommender Systems and Affective Computing. The object of study was an RS of entertaining videos which included emotional information as user feedback. The research on affective RS is still in its infancy, therefore, the first steps towards the investigation and creation of the studied system were taken, answering the research question: "*What are the considerations in the design of a Recommender System for entertaining videos that uses automatic emotion detection through facial expression as feedback?*". For answering the research question, two approaches were taken.

From a sociological point of view, the factors conditioning the intention to adopt such RS were studied. For this purpose, the following sub research questions were answered:

- *What Information and Communication Technologies (ICT) adoption model is adequate for studying the adoption in this case?*
- *What are the factors that influence the adoption of such RS?*

The first question was answered by conducting a literature review of previous works and comparing the presented problem with the problems studied by them. UTAUT2 was chosen as the model to study the adoption, adding modifications to fit in the context. The adoption was measured by means of intention to adopt. The responses of 84 internet users proved that the model was adequate for the context, and an 84% of the variance in the intention to adopt such a system was measured.

Regarding the second question, it was found that the biggest influence on the intention to adopt is the *Hedonic Motivation*, followed by the *Social Influence*. Surprisingly, *Trust* was found not to influence in the intention of adoption. The adoption of an innovation is a big determinant of how successful a technology would be. The findings could be used in the design of an affective RS for entertaining videos to propitiate its adoption.

Respecting a technical point of view, the necessity of data resources to test different algorithms and recommendation techniques was found. Consequently, the following sub research questions were established:

- *How could a set of emotions be created with the purpose of using it in the study of the proposed affective RS?*
- *Could the collected affective information be used for the investigation of affective RSs?*
- *In what ways could the emotional data be processed and used to generate recommendations?*

The first question was answered by designing and implementing a data collection process as a web platform. The design was done based on the literature findings. The well functioning of the implemented platform, in the form of a proof of concept, was validated by distributing it online. It was successfully collected emotional data from 91 male and female users of distinct age groups. The participants were recorded while they visualised entertaining videos obtained from YouTube. The emotions shown while watching the content, in form of eight variables (anger, contempt, disgust, fear, happiness, neutral, sadness and surprise), were detected by the Microsoft Cognitive Services Emotion API per each frame of the recorded video. The collection of the data concluded in a dataset of 435 visualisations of 282 distinct videos by 91 users.

Having this amount of information opens the possibility to investigate the role of emotions in the determination of what a user likes and how to create recommendations with that information. From an analysis of the collected data, a correlation between the emotions shown during the visualisation and the feedback provided to the user was found. This correlation is a good beginning to the long path that affective RSs have until being a mature technology. The second research question can be answered by these affirmations: The correlation of the emotions and the user feedback, as well as the high diversity of emotions shown in the visualisations, prove that further investigation can be done by using the collected data. Moreover, the amount of visualisations obtained allows trying different recommending techniques.

As an answer to the third question, several options for working with the collected data were presented. It was suggested that the main focus of the research should be on the pre-processing of the emotional data. Some recommending choices were suggested for the studied system. It was stated that the recommendations should be computed to generate a sequence of recommended items. Moreover, a hybrid approach was proposed as an initial way to test the data in an RS, using a simplified content based approach combined with an item-based collaborative filtering algorithm. For achieving so, the emotions should be translated into ratings that could be used as input to the recommender algorithm.

The research answered the proposed question by taking the first steps in the design of an affective RS for entertaining videos. Knowing the factors affecting users' intention to adopt and having collected a dataset with emotional information opens the path for further research on how such system should be designed. These considerations could be used to create a successful design of the RS which could be subsequently implemented and tested.

This work can be seen as the settlement of the ground for a field that has a long way to go and a bright future.

Appendix A

Appendix

In this appendix, the first version of the questionnaire for studying adoption is presented. This version had to be adapted after a first revision for difficulties in the understanding of the questions. Some questions were dropped for making the realization of the experiment more pleasant to the subject.

The first version of the text providing the contextual information is presented here:

> What I'm working on in this for my thesis is a recommender system. You might be confused by the name, but you know for sure what it is, and you've used it many times. It is a system that little by little learns about you and works on generating suggestions for you about the different options when you are on-line. For example, Amazon suggests you products that other people bought at the same time as the product you are buying and Netflix recommends you movies based on what you have watched before.
>
> So, what these recommender systems are doing is helping us in the decision making process. Emotions play a very important role in human decision making and, however, they are normally not taken into account in these systems that support us in the decision making process.
>
> In the last years, the technology for recognizing emotions has progressed a lot. Now, just with a picture, a machine can detect from your facial expression how you are feeling. A little bit scary, right? But also very exciting to imagine the wide range of opportunities these improvements lead to.
>
> What I'm doing in my thesis is integrating the technology I just mentioned with a recommender system. It will recommend entertaining videos that make people laugh, and they do so base on the emotions shown in the process of watching the video. The system would work in a way that while a video is playing it is already generating a recommendation for what video to play next. When the current video is over, the next one will automatically start, minimizing the interaction you needs to have with the system, but having a lot of information for being able to display the best possible content for you.

> When using the system, for being able to detect the emotions, you face has to be recorded by the laptop and send to a service able to analyze these emotions. It is important to understand that there would be no human watching the videos, it would be a program designed to detect emotions and then the video would be deleted.

The initial questions are presented in the following table:

Table A.1: Old questions

Performance Expectancy	'I would find the use of an ARS of entertaining videos useful in my daily life
	'I believe that using the ARS would increase my chances of discovering videos that entertain me '
	'I think that using such the ARS would help me discover entertaining videos I like quicker',
Efort expectancy	'I would find learning how to use operate the ARS easy'
	'I think that the interaction with the ARS would be clear and understandable'
	'I believe I would find the ARS easy to use'
	'It would be easy for me to become skillfull using the ARS',
Social influence	'I believe that people who are important to me would think that I should use the ARS '
	'I think that people who influence my behaviour would think that I should use the ARS '
	'I believe that people whose opinions I value would think that I should use_be in favour of me using the ARS'
	'I think my friends would find this ARS attractive'
Facilitating Conditions	'I have the resources necessary to use the ARS'
	'I have the knowledge necessary to use the ARS'
	'I think this ARS would be compatible with other technologies I use
	'I could get help from others when using the ARS'
Hedonic Motivation	'I think using the ARS would be fun'
	'I think using the ARS would be enjoyable'
	'I think using the ARS would be very entertaining'
Habit	'I believe that the use of such ARS could become and habit to me',
	'I think I could become addicted to such technology'
	'I think that using the RS could become natural to me'

Table A.2: Old questions(cont.)

Trust	**'The ARS seems dependable'**
	'The ARS seems reliable'
	I would trust the ARS
Integrity expectancy	'I believe that the ARS can properly detect my emotions'
	'I believe that the ARS would be able to generate proper recommendations play videos I like from the detected emotions
Benevolence expectancy	'I believe that the ARS would use my emotions in my best interest ',
	'I believe that the ARS would use my emotions only for generating recommendations'
Privacy concerns	'My use of the ARS would cause me to lose control over the privacy of my emotions'
	'Signing up for and using the ARS would lead to a loss of privacy for me because my personal and emotive information could be used without my knowledge.'
	'Others might take control of my information if I used the ARS'
Use intention	'If an ARS like this existed I would use it to discover entertaining videos '
	'If an ARS like this existed I would try to use the ARS in my daily life'
	'I would recommend the ARS to others'

The Spanish version of the explanation was as follows:

> En mi Trabajo Fin de Máster estoy estudiando Sistemas Recomendadores Afectivos (SRA). Estos son sistemas que pueden recomendar una actividad basándose en las emociones del usuario adquiridas de forma automática. La idea es usar el SRA en una aplicación online (llamada Recomotions.me) para recomendar videos de entretenimiento. El usuario ve un video, la aplicación detecta las emociones a través de la expresión facial usando la cámara y cuando el video acaba empieza a reproducirse uno nuevo que el sistema piensa que le gustaría al usuario. Estas recomendaciones son personalizadas para cada usuario y mejoran con el tiempo; cada vez que un usuario ve un video más información tiene el sistema sobre él y por lo tanto crea mejores recomendaciones. Más abajo encontrarás un cuestionario para evaluar la aceptación de esta aplicación.

The Spanish version of the questions is shown in the following table:

Table A.3: Spanish questions

Performance Expectancy	**Creo que usar la aplicación *Recomotions.me* aumentaría mis posibilidades de descubrir videos que me entretengan.**
	Creo que usar la aplicación *Recomotions.me* me ayudaría a descubrir más rápido videos de entretenimiento que me gusten.
Efort expectancy	Creo que aprendería a usar la aplicación *Recomotions.me* fácilmente
	Creo que encontraría la aplicación *Recomotions.me* fácil de usar
Social influence	Creo que las personas que son importantes para mí me recomendarían usar la aplicación *Recomotions.me* (ej.: mi familia, mis amigos…)
	Creo que las personas que influyen mi comportamiento me recomendarían usar la aplicación *Recomotions.me* (ej.: gente famosa, gente a la que admiro…)
	Creo que mis amigos encontrarían la aplicación *Recomotions.me* atractiva.
Facilitating Conditions	Tengo los recursos necesarios para usar la aplicación *Recomotions.me*
	Tengo los conocimientos necesarios para usar la aplicación *Recomotions.me*
	Creo que si lo necesitara podría conseguir ayuda de otros al usar la aplicación *Recomotions.me*
Hedonic Motivation	Creo que usar la aplicación *Recomotions.me* sería divertido
	Creo que usar la aplicación *Recomotions.me* sería agradable
	Creo que usar la aplicación *Recomotions.me* sería entretenido
Habit	Creo que el uso de la aplicación *Recomotions.me* podría convertirse en un hábito para mí
	Creo que el uso de la aplicación *Recomotions.me* podría convertirse el algo natural para mí

Table A.4: Spanish questions(cont.)

Trust	**Creo que confiaría en la aplicación *Recomotions.me***
Integrity expectancy	Creo que la aplicación *Recomotions.me* podría detectar mis emociones correctamente
	Creo que la aplicación *Recomotions.me* sería capaz de recomendarme videos que me gusten de las emociones detectadas
Benevolence expectancy	Creo que la aplicación *Recomotions.me* usaría mis emociones en mi mejor interés
	Creo que la aplicación *Recomotions.me* usaría mis emociones sólo para generar recomendaciones
Privacy concerns	Creo que usar la aplicación *Recomotions.me* me causaría perder control sobre la privacidad de mis emociones
	Creo que otros podrían tomar control de mi información si uso la aplicación *Recomotions.me*
Use intention	Si la aplicación *Recomotions.me* existiera la usaría para descubrir videos entretenedores
	Si la aplicación *Recomotions.me* existiera intentaría usarlo en mi vida diaria
	Si la aplicación *Recomotions.me* existiera se la recomendaría a otros.

Appendix B

Appendix

This appendix present the complementary information on how the videos were obtained.

As explained in section 5.3, a open question survey was distributed with the question:

> Hi! we are conducting a study about entertaining videos, and right now we are trying to figure out what kind of "making-you-laugh" videos people watch. We are asking something very easy, just gives as some categories for this kind of videos. It could be the ones you normally watch or something you know that exist, it will take you one minute!!
>
> Here are some examples to boost your creativity:
>
> - Contagious laughs
> - Funny puppies
> - Dumb falls
> - Pranks

Twenty responses were obtained, in Fig. B.1 the answers can be seen.

The search queries obtained using the method explained in 5.3 are presented in table B.1

The relevance of the categories acquired with the survey is shown in table B.2

Tables B.3 and B.4 presents the channels and the IDs of the playlists that were associated to each category.

TABLE B.1: Search queries

funny...	**videos...**	For kids
		animal
		of cats
		that make you laugh so hard you cry
		of babies
		minecraft
	moments...	clash royale
		GTA 5
		football
		rainbow six siege
		BTS
		overmatch
		battelfield 1
		LOL
	cats...	compilations
		videos
		vines
		fails
		and dogs
		dancing
		talking
	fails...	compilations
		GTA 5
		Kids
		animals
		try not to laugh
		vines
		clash royale
	vines...	compilation
		kids
		try not to laugh
		fails
		animals
		cats
		clash royale

TABLE B.2: Category Relevance classification

High relevance	**Medium Relevance**	**Low Relevance**
Dumb falls	Acting	Babies laughing
Comedy monologues	Animal videos	Babys stories
Parodies	Animations	Big Bang Theory Best Moments
Pranks	Babies	Casually Explained
	Cats	cats falling
	Children interviews	family guy best moments
	Comedies	Friends Best Moments
	Comedy movies trailers	How I met you mother Best Moments
	comedy sketches	Modern Family Best Moments
	Contagious laughs	songs parodies
	darwin awards	The simpsoms best moments
	everyday issues	
	Fail compilations	
	Falls	
	football best moments	
	Funny stories	
	Funny subtitles	
	Hidden camera pranks	
	Instant karma videos	
	Kids	
	political sketches	
	product reviews	
	puppies	
	sports best moments	
	Stand-up comedy	
	TV exercise contests	

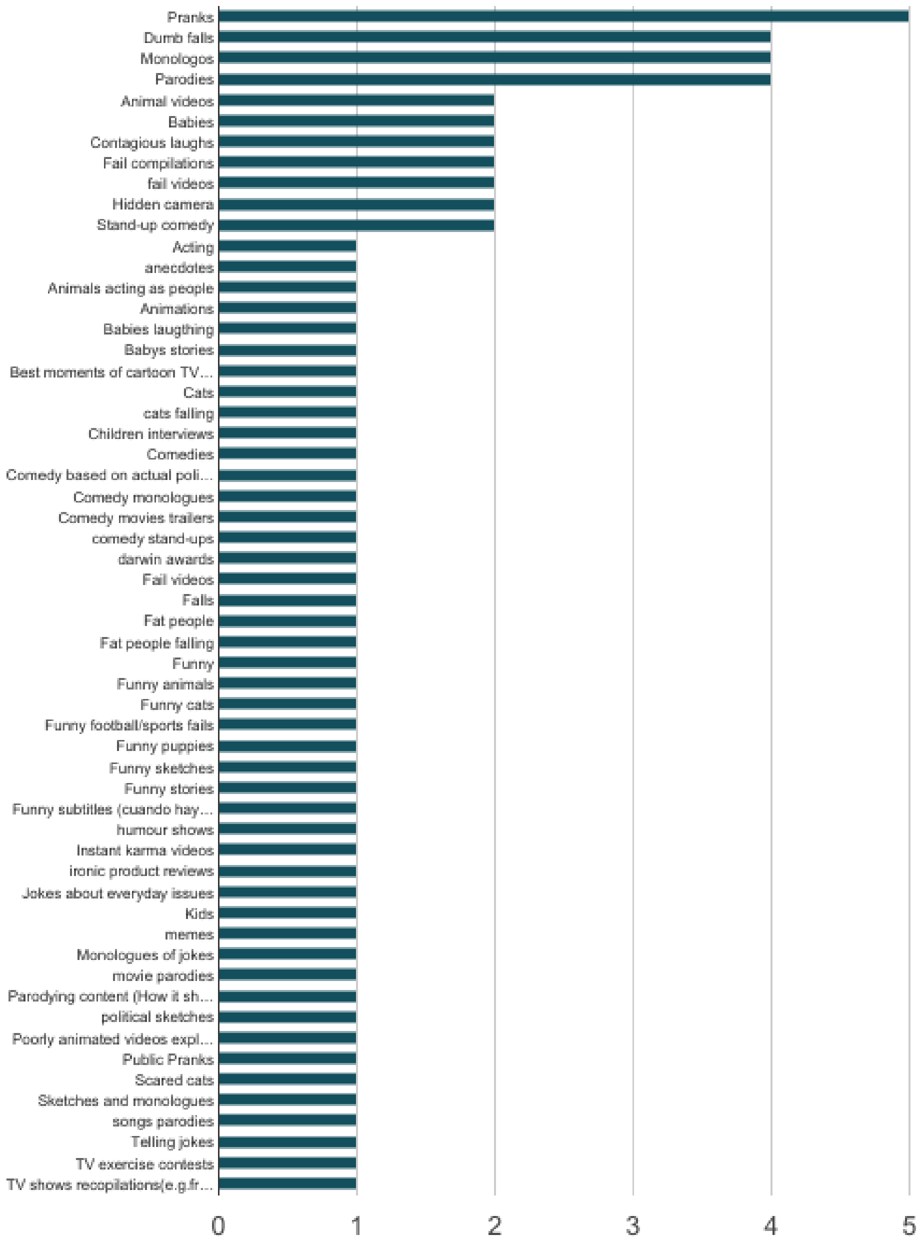

Figure B.1: Video Categories - Number of Responses

TABLE B.3: Categories, channels and playlists (1)

Category	Channels	Playlist	Playlist ID
Adventure	CaseyNeistat	best of CASEY NEISTAT	PLTHOlLMWEwVy2ZNmdrwRIRlVfZ8fiR_ms
	GoPro	GoPro Awards: Official Selections	PLSSPBo7OVSZsljlGMHtOnWm8HXdUGl2Xt
	FunForLouis	Liked videos	LLVrvnobbNGGMsS5n2mJwfOg
Animals	Tiger Productions	Best & Funniest Animal Compilations on YouTube	PLtDp75hOzOlaQcPfx-Za_Dd1sOBPtdBw3
	MashupZone	Pets And Animals Compilations	PL-GeRX7sMZdZpfQGMibbfwlkzUZ5utK77
	Life Awesome	Funny Animals Videos 2017 | Funny Cats & Dogs Videos	PLGTScjq3z9bjBXGEUADkq27kmtfO-s25G
	Funny Videos Channel	Funny Dogs - funny animals 2016	PLI-cuUno6CUhEBEZ_SROzBFbSYNQ0ZIf5
	funnyplox	Animal Compilations	PLvKo-7UxiUK5RuhJdbF0mqN1aqPdY6heZ
Animations	TheFearRaiser	Animations!	PLFQr2i3iRRu4lTuaYKSpdPJB9BgACIkvl
	Simon's Cat	Liked videos	LLH6vXjt-BA7QHI0KnfL-7RQ
	Domics	Favourites	FLn1XB-jvmd9fXMzhiA6IR0w
	sWooZie	Featured Videos	PL50E2211EBC39DC15
	Casually explained	All Casually Explained	PLbdSi72ah3puMBGa4eLQMNKn4OqaVPsHF
Babies	funnyplox	Baby Compilations	PLvKo-7UxiUK7eGN7STEBXYr4STqebw7r3
	MashupZone	Babies Compilations	PL-GeRX7sMZdY_Q8yEe0vwoNS72wDkmpYj
	Funny Videos Channel	Best Funny Kids Video	PLI-cuUno6CUgup5L-bdO_jMp2clbdbxfy
	Baby Clips Daily	Funniest Babies	PLvUMgpmq3DmWo2k8QR2a61nxO14xUmMx8
Best of TV shows	Jimmy Kimmel Live	Best Of: Recent Clips	PLs4hTtftqnlCrhT0aQnErcGDKtYz7gnTz
	TheEllenShow	All-Time Favorite Memorable Moments	PLuW4g7xujBWeSwXTonWXh-OpwkKC85dj4
	The Tonight Show Starring Jimmy Fallon	Liked videos	LL8-Th83bH_thdKZDJCrn88g
	The Tonight Show Starring Jimmy Fallon	Liked videos	LL8-Th83bH_thdKZDJCrn88g
	Jimmy Kimmel Live	Best Of: Lie Witness & Pedestrian Question	PLs4hTtftqnlB5b12gaErZ54al4sQ7iINJ
	The Tonight Show Starring Jimmy Fallon	#BestOfFallon 2016	PLykzf464sU9-0QSdgYy52uW7HKSfpc2dP
	Last night tonigh	What to watch next	PLmKbqjSZR8TZa7wyWoVq2XMHxxWREyiFc
Comedy	Comedy Central	@midnight	PLD7nPL1U-R5rDpeH95XsK0qwJHLTS3tNT
	Funny Or Die	FOD has Mashups	PLRcB4n4CGcy-1dNOE8Ni6imukruah6WqO
Everyday life jokes	CollegeHumor	Hardly Working	PLuKg-WhduhklMSnK_EVV3GbsGtbsLMZT2
	Andrew Trabass	Comedy Sketches | Trabass TV	PLGQyLHJIdRPDu1Ubwovbg AmaEDaQfGDnh
	BuzzFeedVideo	For The First Time...	PL5vtqDuUM1DmLUNRY70igk0iNWCTN1__f
	Smosh	Every Blank Ever	PLShD8ZZW7qjljIZ-36_rqcWUNKDnbTMRS
	Nerdist	Featured Videos	PLI4T6p7km9dbGAHMyZ09nyW9-yhVK8zTg
	CollegeHumor	Adam Ruins Everything	PLuKg-WhduhkksJoqkj9aJEnN7v0mx8yxC

TABLE B.4: Categories, channels and playlists (2)

Category	***Channels***	***Playlist***	***Playlist ID***
Fails	FailArmy	FailArmy Best Epic Fail Compilations	PLOWSbqDrrl5Tb6W1EP9WZEaWTP6lj-6bg
	Break	Winter Fails - by Break	PLFFBC1458E25F1AB2
	FAIL Blog	Classic FAILs Playlist	PL85310CC7C61EFAEF
	Break	Man vs Nature - Top 20 Countdown	PL7B36A54220771D93
Interesting facts	MrGear	Interesting	PLnBRCTk6a_GtfHfk6UTH5VfxknnhYK9Nz
	TheRichest	Brain Food	PLXK3uPjJ22FU65ma1anJR5M94kXCrYm2E
	Funny Or Die	FOD Has Politics	PLRcB4n4CGcy_xbD8CHLVSU3_Th_aQ_v4q
	Alltime10s	Alltime10s Ultimate List Part 4	PLec1lxRhYOzvy5GFfvGYitqrL2KgOwdBJ
	Vsauce	Liked videos	LL6nSFpj9HTCZ5t-N3Rm3-HA
Kids	Life Awesome	Funny Kids Fails Compilation | Funny Kids Videos	PLGTScjq3z9bg-WTq2_dxa1ZUpYhKUlvHT
	SoulPancake	Kid President	PLzvRx_johoA-Yabl6FWcU-jL6nKA1Um-t
	SoulPancake	Little Kids. Big Questions.	PLzvRx_johoA-iT5zK8EhRFA3C7DmOPVlo
	Jimmy Kimmel Live	YouTube Challenge - I Told My Kids I Ate All Their Halloween Candy	PLs4hTtftqnlBuue-6Q1WKBHVAjLKV3H_D
	WatchCut Video	HiHo Kids	PLJic7bfGlo3pHYEcliNDle6TS_J7jwyEJ
	WatchCut Video	Kids Explain	PLJic7bfGlo3qjESLRBpJ9mV1SsKIlF7rn
	TheEllenShow	Adorable Kids	PLuW4g7xujBWfU26JUTW1DGs3hk4LD5KaL
Magic	MrGear	All Videos	PLnBRCTk6a_Gue8q6_Pn_sEBCjYsOx7PC7
	Disturb Reality	Liked videos	LLba2ulYq75m6SNuK8TtmG9A
	Justin Flom	All My Videos	PLzVaSLIcQbfUY7MNs4KuEAL-UafCZWgOo
	Funny Vines	Best magic show in the world 2016 - Best magic trick ever	PLnIST2IBA34uPrgpfUwkISkjCaVAvrN89
Parodies	Bart Baker	Parodies	PL4722096DA7FECEFD
	BuzzFeedVideo	Disney Reimagined	PL5vtqDuUM1Dnw_J7rQ9ggAzJ5PGYSNnZz
	Smosh	Music Videos	PL21A272CC1406CF4A
	The Key of Awesome	The Key of Awesome!	PL1C90BDF46E6EACFD
Pranks	Jimmy Kimmel Live	Kimmel Pranks	PLs4hTtftqnlAfhawBAgdjiCck6Nhj22N8
	HoomanTV	PUBLIC PRANKS ON PEOPLE:	PLzy8z75ub5ahTxHEe6rUQ9utx6xlBgfW8
	whatever	MOST VIEWED	PL2uZhEhKQPWYeyAsRV9Rk3IvewORrZ18e
	TheEllenShow	Hidden Camera Pranks	PL783C8A51F5E848CE
	Rémi GAILLARD	Recommended	PLPipABjP1XLgevZIq5Ux8f5i0GksZo9kG
	AndrewSchrock	Pranks!	PLF1EB5D29AE4B86B2
Stand-up comedy	Comedy Central	CC:Stand-Up	PL06EFE28BC2DD3678
	BBCWorldwide	Stand Up Comedy	PL2179AB0D78E27E0B
	Team Coco	Stand-Up on CONAN	PL82F269F11AB50FA4

Appendix C

Appendix

This appendix presents in full size the figures 6.8 to 6.11 from section 6.1.5

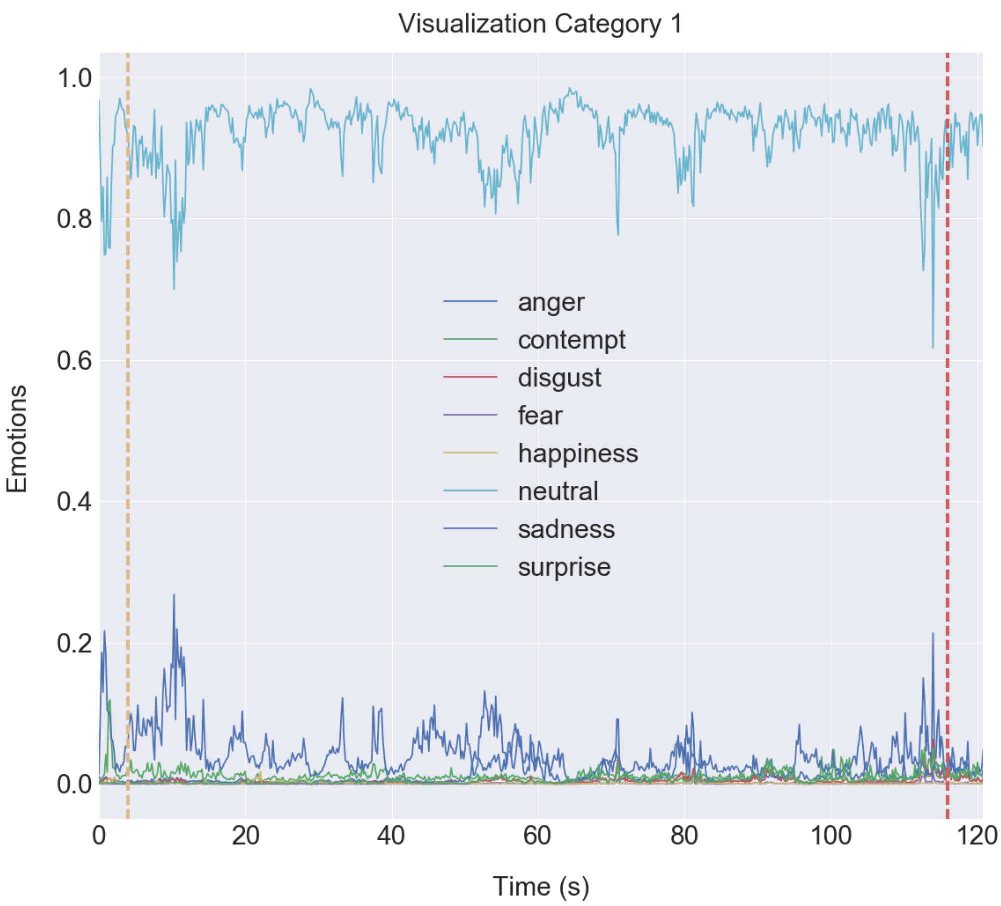

FIGURE C.1: Emotions over time, group 1, example 1 (full size)

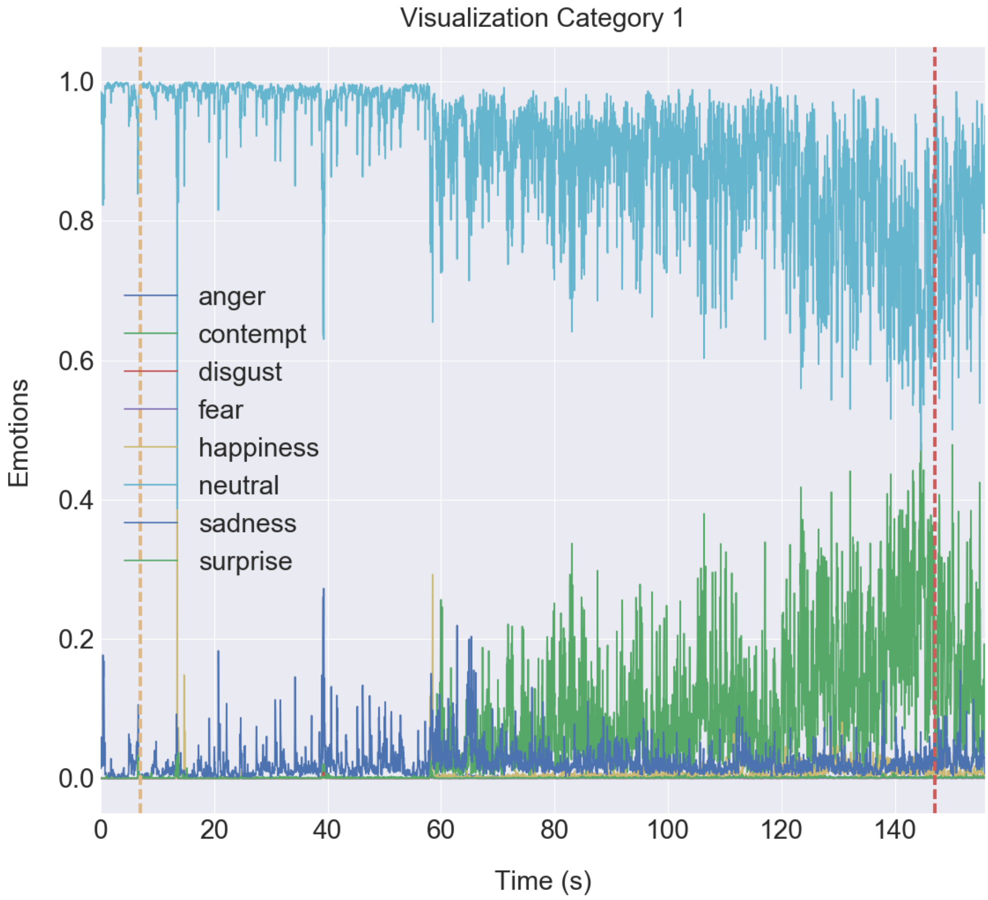

FIGURE C.2: Emotions over time, group 1, example 2 (full size)

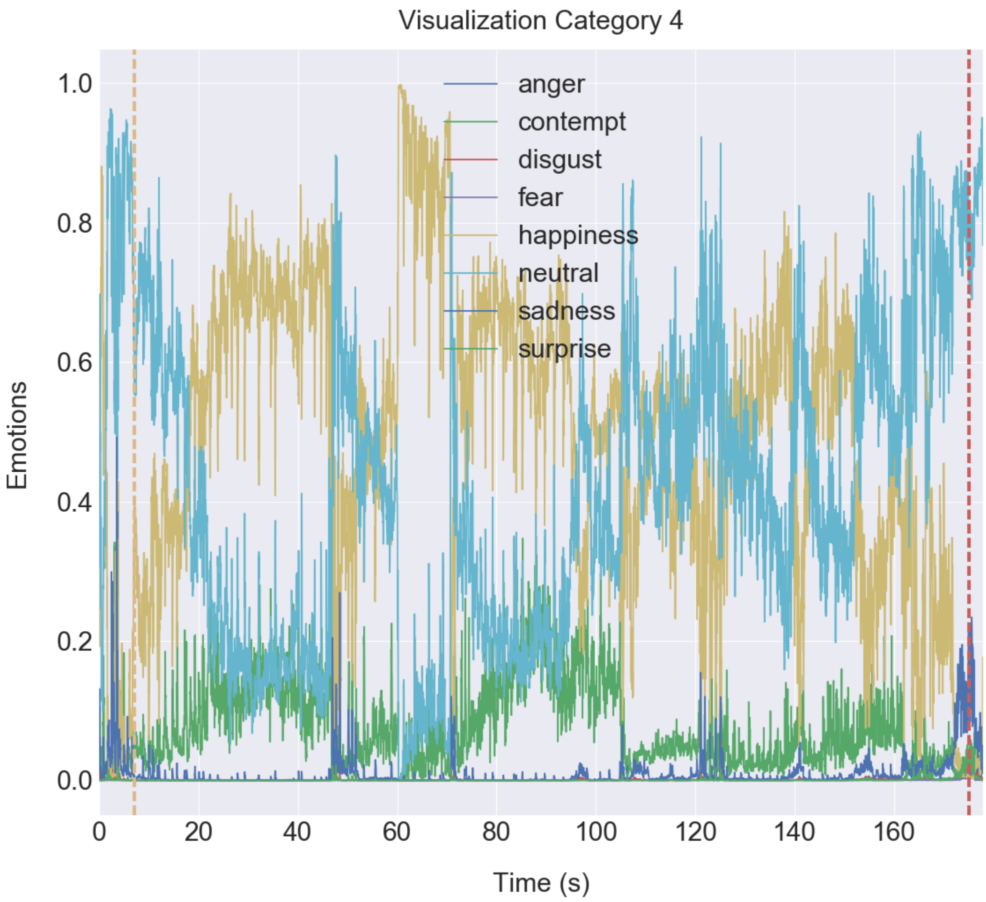

FIGURE C.3: Emotions over time, group 4, example 1 (full size)

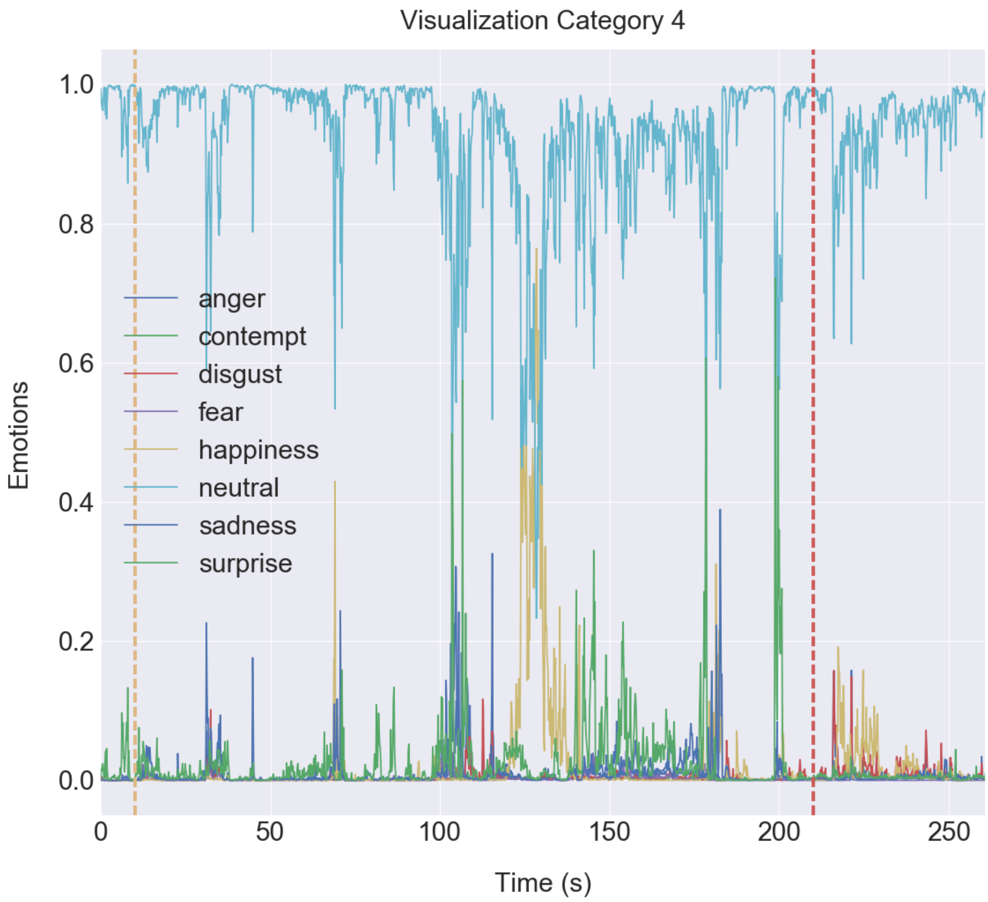

Figure C.4: Emotions over time, group 4, example 2 (full size)

Appendix D

Appendix

Figures from 6.13 to 6.23 are shown here in bigger size:

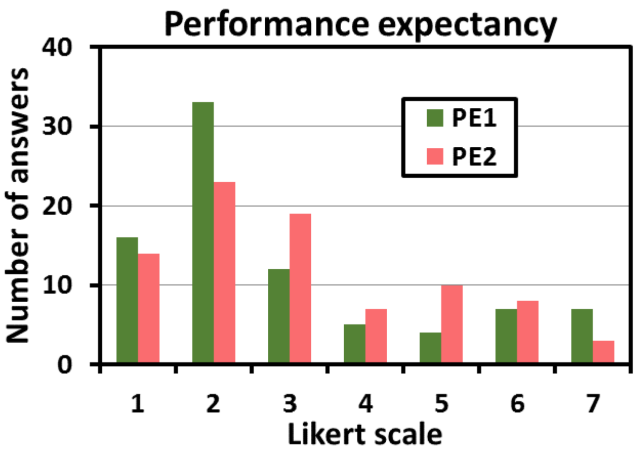

FIGURE D.1: nº ans. per item PE (full size)

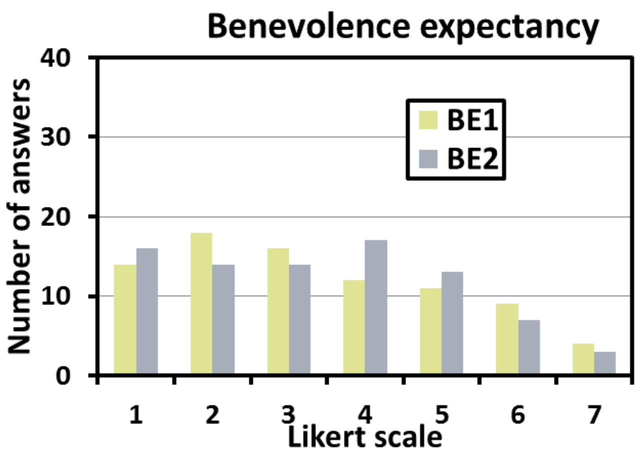

FIGURE D.2: nº ans. per item BE (full size)

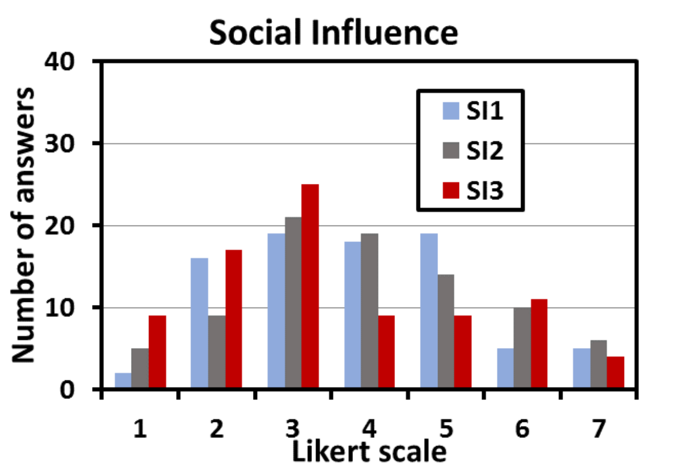

FIGURE D.3: nº ans. per item SI (full size)

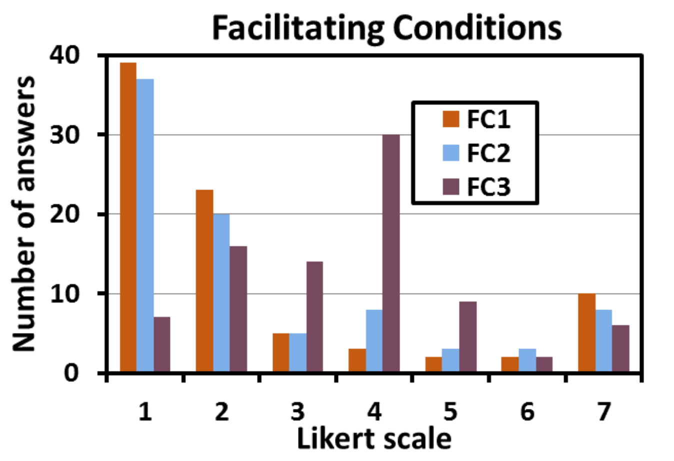

FIGURE D.4: nº ans. per item FC (full size)

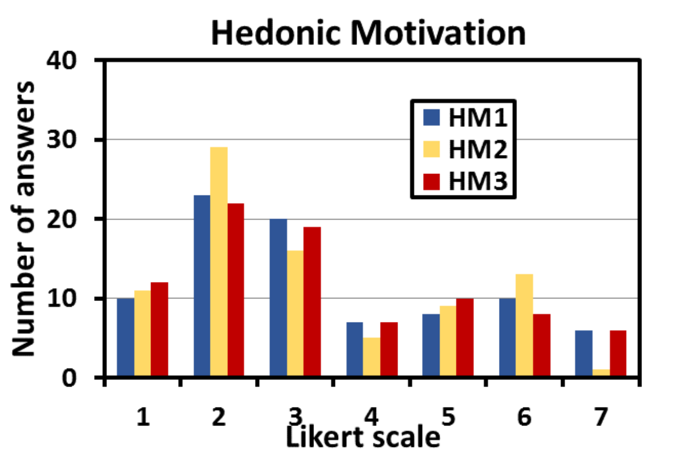

FIGURE D.5: nº ans. per item HM (full size)

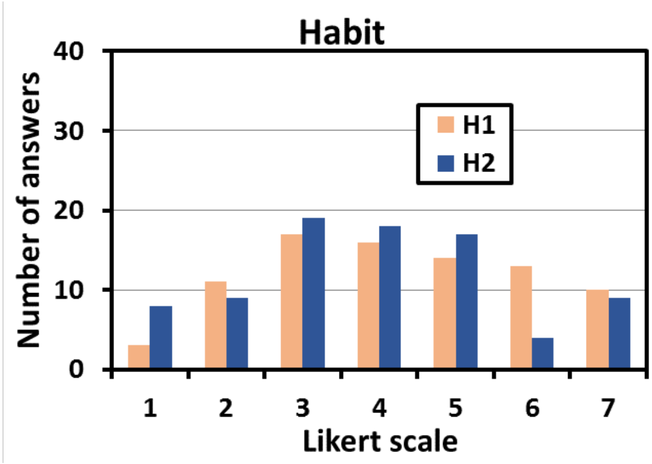

FIGURE D.6: nº ans. per item H (full size)

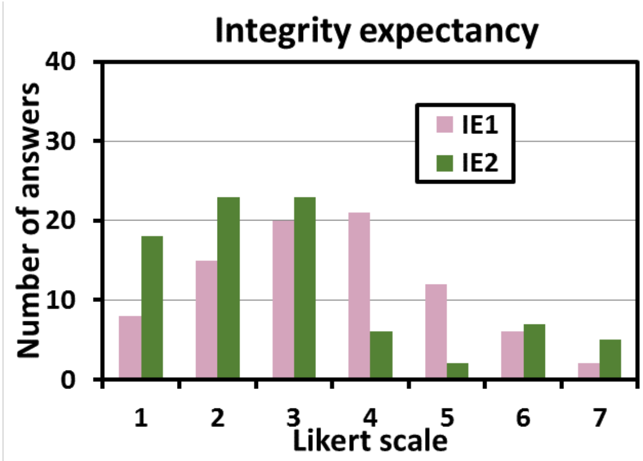

FIGURE D.7: nº ans. per item IE (full size)

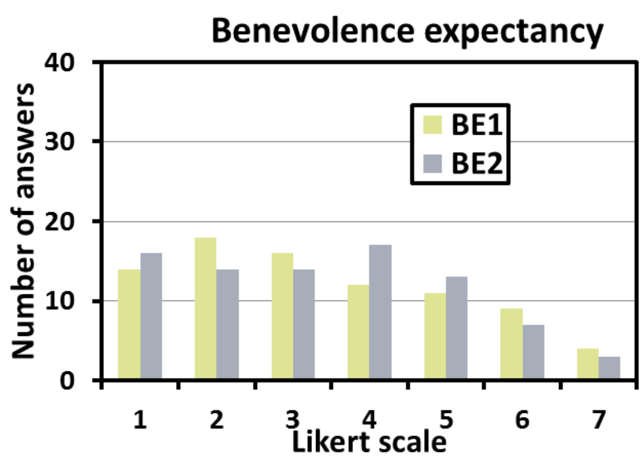

FIGURE D.8: nº ans. per item BE (full size)

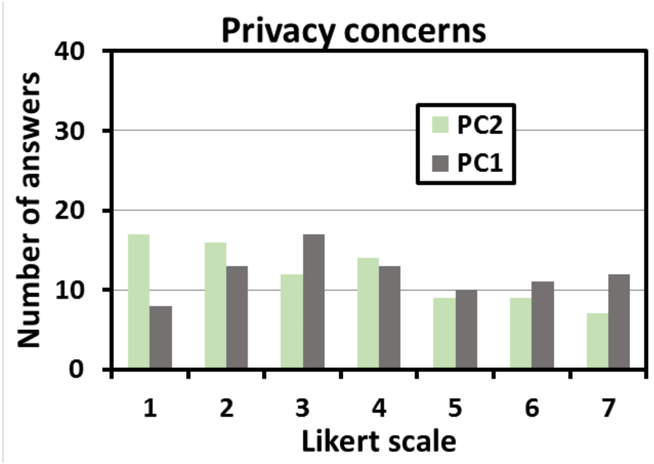

FIGURE D.9: nº ans. per item PC (full size)

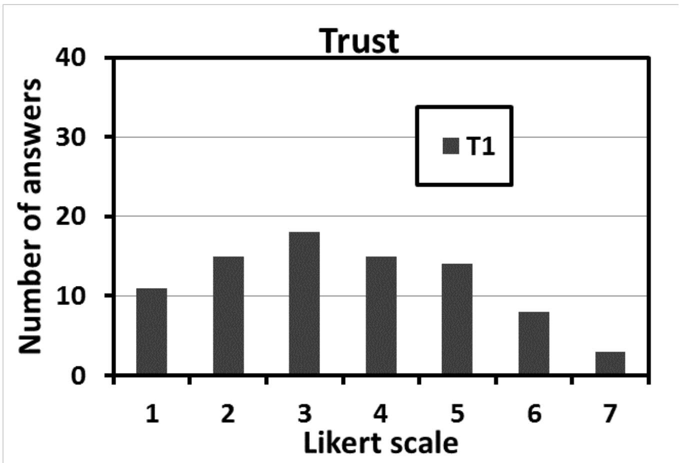

FIGURE D.10: nº ans. per item T (full size)

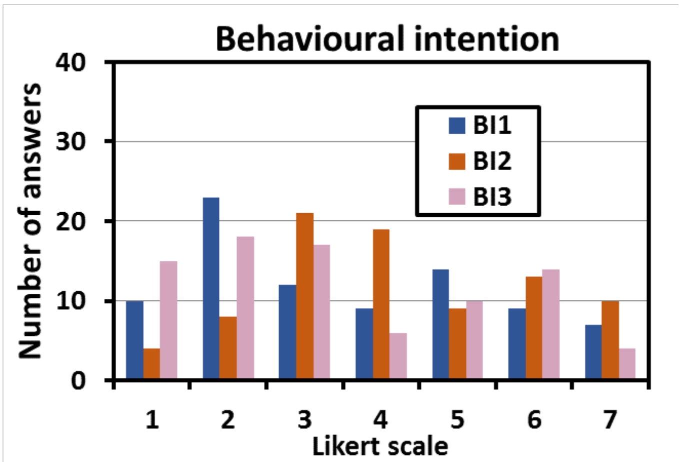

FIGURE D.11: nº ans. per item BI (full size)

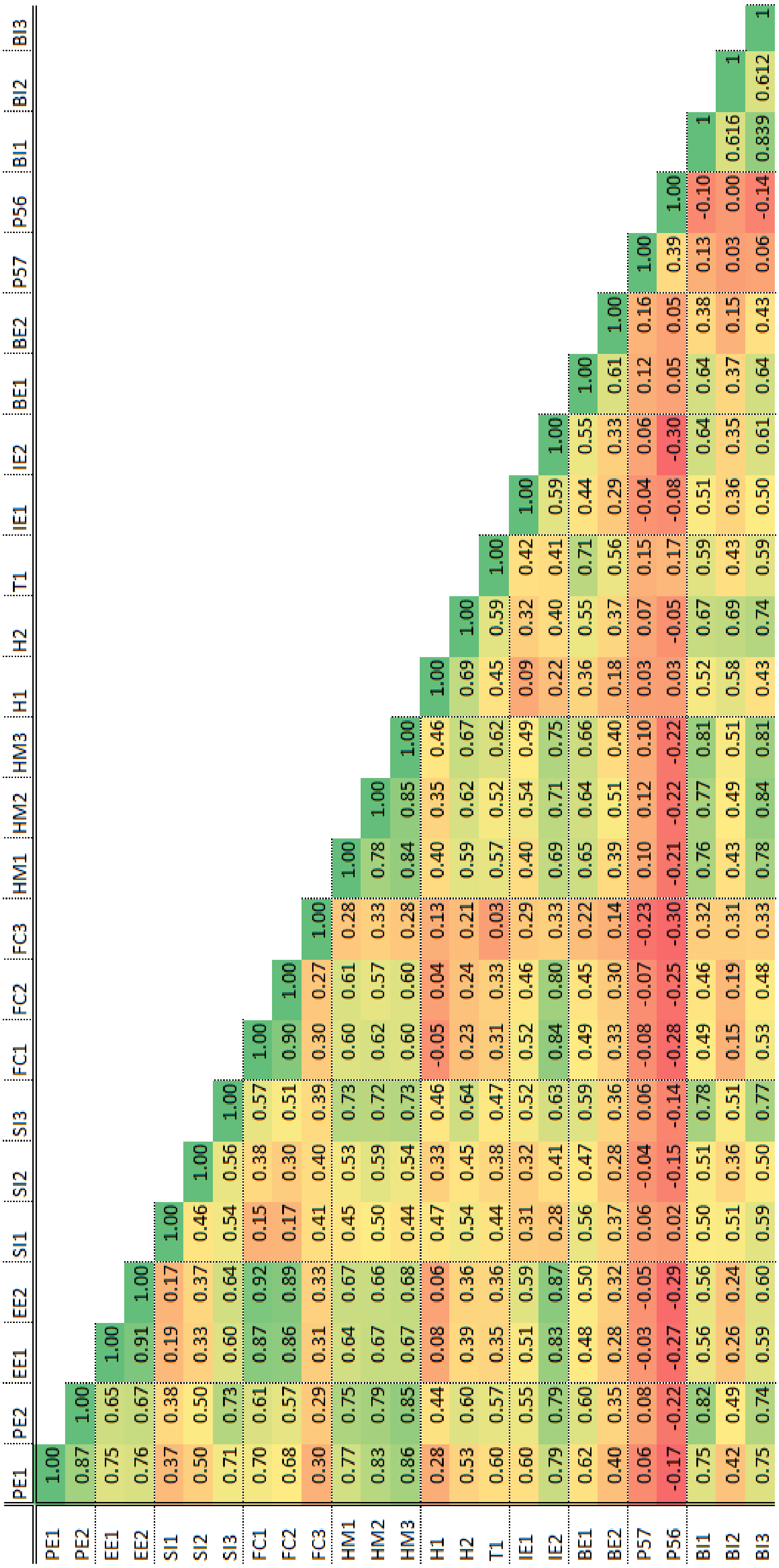

FIGURE D.12: figure
Correlations between Items

TABLE D.1: Segmentation Results (full size)

Segmentation	General		Gender				Age				Country			
			Male		***Female***		***Age ≤ 30***		***Age > 30***		***Spain***		***Other***	
R^2 (Trust)	*0.55*		*0.65*		*0.83*		*0.83*		*0.93*		*0.88*		*0.92*	
	path coe.	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*
IE → T	0.18	0.06	0.13	0.21	0.26	0.14	0.18	0.13	0.19	na	0.25	0.03	0.24	0.44
BE → T	0.61	0.00	0.67	0.00	0.52	0.00	0.62	0.00	0.70	na	0.59	0.00	0.60	0.00
PC → T	0.18	0.01	0.20	0.06	0.14	0.28	0.28	0.00	-0.06	na	0.15	0.07	0.35	0.04
R^2	0.84		0.90		0.48		0.59		0.59		0.58		0.64	
(Behavioural I.)	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*	*path coe.*	*p-value*
T → BI	0.06	0.43	0.24	0.00	-0.04	0.76	0.01	0.96	0.09	na	-0.07	0.36	0.22	0.92
PE → BI	0.15	0.32	0.12	0.37	0.18	0.52	0.26	0.21	0.16	na	0.02	0.86	0.41	0.84
EE → BI	0.03	0.85	0.31	0.09	-0.47	0.13	0.07	0.77	-0.16	na	-0.03	0.85	0.05	0.93
SI → BI	0.25	0.00	0.20	0.18	0.40	0.01	0.16	0.17	0.01	na	0.17	0.04	0.34	0.91
FC → BI	-0.10	0.51	-0.32	0.12	0.22	0.35	-0.16	0.43	0.17	na	-0.09	0.60	0.02	0.99
HM → BI	0.49	0.00	0.44	0.03	0.56	0.04	0.47	0.03	0.00	na	0.86	0.00	-0.11	0.88
H → BI	0.13	0.10	0.05	0.65	0.15	0.26	0.17	0.11	0.47	na	0.09	0.30	0.22	0.70

Bibliography

[1] Iterative Model: What Is It And When Should You Use It? URL https://airbrake.io/blog/sdlc/iterative-model.

[2] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016. ISSN 14321300. doi: 10.1007/s00799-015-0156-0.

[3] IBM - What is big data? URL https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html.

[4] White Paper. Cisco Visual Networking Index: Forecast and Methodology Cisco Visual Networking Index: Cisco Visual Networking Index: Forecast and Methodology. *Forecast and Methodology*, pages 2015–2020, 2015.

[5] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer US, Boston, MA, 2015. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3. URL http://link.springer.com/10.1007/978-1-4899-7637-6http://www.springer.com/us/book/9780387858203.

[6] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1):1–49, 2012. ISSN 03701573. doi: 10.1016/j.physrep.2012.02.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0370157312000828{%}5Cnhttp://www.sciencedirect.com/science/article/pii/S0370157312000828.

[7] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. Setting goals and choosing metrics for recommender system evaluations. In *CEUR Workshop Proceedings*, volume 811, pages 78–85, 2011.

[8] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, 2012. ISSN 09574174. doi: 10.1016/j.eswa.2012.02.038. URL http://dx.doi.org/10.1016/j.eswa.2012.02.038.

[9] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013. ISSN 09507051. doi: 10.1016/j.knosys.2013.03.012. URL http://dx.doi.org/10.1016/j.knosys.2013.03.012.

[10] R W Picard. Affective Computing. *Science*, (321), 1995.

[11] Marco de Gemmis, Nadja De Caroli, Andrej Košir, and Marko Tkalčič. *Emotions and Personality in Personalized Systems*, volume 28. 2016. ISBN 978-3-319-31411-2. doi: 10.1007/978-3-319-31413-6.

[12] Marko Tkalčič, Andrej Košir, Jurij Tasič, and Matevž Kunaver. Affective recommender systems : the role of emotions in recommender systems. *Proceedings of the RecSys 2011 Workshop on Human Decision Making in Recommender Systems*, (January):9–13, 2011.

[13] Claudia Orellana-rodriguez, Ernesto Diaz-aviles, and Wolfgang Nejdl. Mining Affective Context in Short Films for Emotion-Aware Recommendation. *Hypertext 2015: Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 185–194, 2015. doi: 10.1145/2700171.2791042.

[14] Rafael A Calvo, Sidney K D 'mello, Jonathan Gratch, and Arvid Kappas. Introduction: A Guided Tour to the Handbook of Affective Computing. 2010.

[15] Harsh Barry. What is Waterfall model- advantages, disadvantages and when to use it?, 2012. URL http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/.

[16] Harsh Barry. What is Waterfall model- advantages, disadvantages and when to use it?, 2012. URL http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/.

[17] Harsh Barry. What is Waterfall model- advantages, disadvantages and when to use it?, 2012. URL http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/.

[18] Oliver Oechslein, Marvin Fleischmann, and Thomas Hess. An application of UTAUT2 on social recommender systems: Incorporating social information for performance expectancy. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 3297–3306, 2014. ISSN 15301605. doi: 10.1109/HICSS.2014.409.

[19] Xin Venkatesh, Viswanath., Thong, James, Y.L. & Xu. Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology. *MIS Quarterly*, 36(1):157–178, 2012. ISSN 02767783. doi: 10.1111/j.1540-4560.1981.tb02627.x. URL http://s3.amazonaws.com/academia.edu.documents/36422124/Venkatesh{_}utaut2.pdf{%}3DUnified{_}theory{_}of{_}acceptance{_}and{_}use{_}of.pdf.

[20] Kelly Caine. EngageMe : The Design and Implementation of a Reflective Tool for Evaluating Student Engagement. (May), 2015.

[21] Per E Pedersen and Rich Ling. Modifying adoption research for mobile Internet service adoption : Cross- disciplinary interactions. *Proceedings of the 36th Hawaii International Conference on System Sciences*, 00(C):1–10, 2003. doi: 10.1109/HICSS.2003.1174217.

[22] John Dudovskiy. Convenience sampling. Research methodology, 2016. URL http://research-methodology.net/sampling-in-primary-data-collection/convenience-sampling/.

[23] Tübingen. Machine Learning Summer School. URL http://mlss2014.com/index.htmlhttp://www.mlss.cc/tuebingen07/program.

[24] Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5535 LNCS:247–258, 2009. ISSN 03029743. doi: 10.1007/978-3-642-02247-0_24.

[25] Bamshad Mobasher. Recommender Systems. *Artificial Intelligence*, pages 1–29, 2007. doi: 10.1007/978-3-319-29659-3.

[26] Microsoft Cognitive Services - Emotion API. URL https://www.microsoft.com/cognitive-services/en-us/emotion-api.

[27] Affectiva Inc. Affectiva Developer Portal - Accuracy, 2016. URL https://developer.affectiva.com/http://developer.affectiva.com/accuracy/.

[28] Nviso. Artificial Intelligence Emotion Recognition Software — nViso. URL http://www.nviso.ch/http://www.nviso.ch/index.html.

[29] Vision API - Image Content Analysis — Google Cloud Platform. URL https://cloud.google.com/vision/https://cloud.google.com/vision/{%}0Ahttps://cloud.google.com/vision/?utm{_}source=google{&}utm{_}medium=cpc{&}utm{_}campaign=2016-q1-cloud-latam-ML-skws-freetrial.

[30] Maya Hristakeva. Overview of Recommender Algorithms – Part 1 — A Practical Guide to Building Recommender Systems on WordPress.com, 2015. URL https://buildingrecommenders.wordpress.com/2015/11/16/overview-of-recommender-algorithms-part-1/.

[31] Maya Hristakeva. Overview of Recommender Algorithms – Part 1 — A Practical Guide to Building Recommender Systems on WordPress.com, 2015. URL https://buildingrecommenders.wordpress.com/2015/11/20/overview-of-recommender-algorithms-part-4/https://buildingrecommenders.wordpress.com/2015/11/16/overview-of-recommender-algorithms-part-1/.

[32] Kim Falk. Practical Recommender Systems. chapter 1.

[33] Marko Tkalčič, Andrej Košir, and Jurij Tasič. The LDOS-PerAff-1 corpus of facial-expression video clips with affective, personality and user-interaction metadata. *Journal on Multimodal User Interfaces*, 7(1-2):143–155, 2013. ISSN 17837677. doi: 10.1007/s12193-012-0107-7.

[34] Marko Tkalčič, Urban Burnik, and Andrej Košir. Using affective parameters in a content-based recommender system for images. *User Modeling and User-Adapted Interaction*, 20(4):279–311, 2010. ISSN 15731391. doi: 10.1007/s11257-010-9079-z.

[35] Rahul Katarya and Om Prakash Verma. Recent developments in affective recommender systems. *Physica A: Statistical Mechanics and its Applications*, 461: 182–190, 2016. ISSN 03784371. doi: 10.1016/j.physa.2016.05.046. URL http://dx.doi.org/10.1016/j.physa.2016.05.046.

[36] Abhishek Mahata, Nandini Saini, and Sneha Saharawat. Intelligent Movie Recommender System Using Machine Learning. 10127:94–110, 2017. doi: 10.1007/978-3-319-52503-7. URL http://link.springer.com/10.1007/978-3-319-52503-7.

[37] L Pauly and D Sankar. A novel online product recommendation system based on face recognition and emotion detection, 2015. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84974824429{&}partnerID=40{&}md5=4c088ef40b94c12134741c266b783787.

[38] Hyon Hee Kim. A Semantically Enhanced Tag-based Music Recommendation Using Emotion Ontology. *Proceedings of the 5th Asian Conference on Intelligent Information and Database Systems - Volume Part II*, pages 119–128, 2013. doi: 10.1007/978-3-642-36543-0_13. URL http://dx.doi.org/10.1007/978-3-642-36543-0{_}13.

[39] Eleanor Mulholland, Paul Mc Kevitt, Tom Lunney, and Karl-michael Schneider. Analysing Emotional Sentiment in People ' s YouTube Channel Comments.

[40] Yashar Moshfeghi and Joemon M Jose. An effective implicit relevance feedback technique using affective, physiological and behavioural features. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, pages 133–142, 2013. doi: 10.1145/2484028.2484074. URL http://dl.acm.org/citation.cfm?doid=2484028.2484074.

[41] Karzan Wakil and Karwan Ali. Improving Web Movie Recommender System Based on Emotions. 6(2), 2015.

[42] Yong Zheng, Bamshad Mobasher, and Robin Burke. Emotions in context-aware recommender systems., 2016. URL http://link.springer.com/10.1007/978-3-319-31413-6{_}15http://ovidsp.ovid.com/ovidweb.cgi?T=JS{&}PAGE=reference{&}D=psyc13{&}NEWS=N{&}AN=2016-40474-015.

[43] A Košir, A Odić, M Kunaver, M Tkalčič, and JF Tasič. Ldos-comoda dataset, 2013.

[44] Derick Leony, Hugo A. Parada Gélvez, Pedro J. Mũnoz-Merino, Abelardo Pardo, and Carlos Delgado Kloos. A generic architecture for emotion-based recommender systems in cloud learning environments. *Journal of Universal Computer Science*, 19 (14):2075–2092, 2013. ISSN 0948695X.

[45] An Affect-Based Multimodal Video Recommendation System An Affect-Based Multimodal. (March), 2016.

[46] Emotion-based analysis and recommendation of lectures — InEvent - Accessing Dynamic Networked Multimedia Events. URL https://www.inevent-project.eu/demos/emotion-based-analysis-and-recommendation-of-lectures.

[47] Mixed Emotions. URL https://mixedemotions-project.eu/.

[48] Skyscanner
Gives You Tips Based on Your Emotions — Sightcorp. URL http://sightcorp.com/skyscanner-offers-you-travel-recommendations-based-on-your-emotions/.

[49] R. Kaliouby, M. Bahgat, R.W. Picard, R.S. Sadowsky, and O.O. Wilder-Smith. Video recommendation based on affect, August 30 2012. URL https://www.google.com/patents/US20120222058. US Patent App. 13/406,068.

[50] Tai-Kuei Yu, Mei-Lan Lin, and Ying-Kai Liao. Understanding factors influencing information communication technology adoption behavior: The moderators of information literacy and digital skills. *Computers in Human Behavior*, 71:196–208, 2017. ISSN 0747-5632. doi: http://dx.doi.org/10.1016/j.chb.2017.02.005. URL http://www.sciencedirect.com/science/article/pii/S074756321730078X.

[51] Yen Yao Wang, Andy Luse, Anthony M. Townsend, and Brian E. Mennecke. *Understanding the moderating roles of types of recommender systems and products on customer behavioral intention to use recommender systems*, volume 13. Springer Berlin Heidelberg, 2014. ISBN 1025701402. doi: 10.1007/s10257-014-0269-9. URL http://dx.doi.org/10.1007/s10257-014-0269-9.

[52] Stephan Hammer, Kim Kirchner, Elisabeth André, and Birgit Lugrin. Touch or Talk ? - Comparing Social Robots and Tablet PCs for an Elderly Assistant Recommender System. pages 129–130, 2017. doi: 10.1145/3029798.3038419.

[53] Donghyun Kim and Tony Ammeter. Predicting personal information system adoption using an integrated diffusion model. *Information and Management*, 51(4):451–464, 2014. ISSN 03787206. doi: 10.1016/j.im.2014.02.011. URL http://dx.doi.org/10.1016/j.im.2014.02.011.

[54] Xin Venkatesh, Viswanath., Thong, James, Y.L. & Xu. Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology. *MIS Quarterly*, 36(1):157–178, 2012. ISSN 02767783. doi: 10.1111/j.1540-4560.1981.tb02627.x. URL http://s3.amazonaws.com/academia.edu.documents/36422124/Venkatesh{_}utaut2.pdf{%}3DUnified{_}theory{_}of{_}acceptance{_}and{_}use{_}of.pdf.

[55] Michael D. Williams, Yogesh K. Dwivedi, Banita Lal, and Andrew Schwarz. Contemporary trends and issues in IT adoption and diffusion research. *Journal of Information Technology*, 24(1):1–10, 2009. ISSN 02683962. doi: 10.1057/jit.2008.30.

[56] Fred D . Davis Viswanath Venkatesh , Michael G . Morris , Gordon B . Davis, Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3): 425–478, 2003. ISSN 02767783. doi: 10.2307/30036540. URL http://www.jstor.org/stable/10.2307/30036540http://www.jstor.org/stable/30036540.

[57] Xitong Guo, Xiaofei Zhang, and Yongqiang Sun. The privacy-personalization paradox in mHealth services acceptance of different age groups. *Electronic Commerce Research and Applications*, 16:55–65, 2016. ISSN 15674223. doi: 10.1016/j.elerap.2015.11.001. URL http://dx.doi.org/10.1016/j.elerap.2015.11.001.

[58] Hadeel Alharbi, Kamaljeet Sandhu, and Trevor Brown. The Acceptance of E-learning Recommender System for Saudi Universities. In *Proceedings of the The International Conference on Engineering & MIS 2015 - ICEMIS '15*, pages 1–6, 2015. ISBN 9781450334181. doi: 10.1145/2832987.2833066. URL http://dl.acm.org/citation.cfm?doid=2832987.2833066.

[59] Marcelo G. Armentano, Roberto Abalde, Silvia Schiaffino, and Analia Amandi. User Acceptance of Recommender Systems: Influence of the Preference Elicitation Algorithm. *2014 9th International Workshop on Semantic and Social Media Adaptation and Personalization*, (Figure 1):72–76, 2014. doi: 10.1109/SMAP.2014.18. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6978956.

[60] Panos Kourouthanassis, Costas Boletsis, Cleopatra Bardaki, and Dimitra Chasanidou. Tourists responses to mobile augmented reality travel guides: The role of emotions on adoption behavior. *Pervasive and Mobile Computing*, 18:71–87, 2015. ISSN 15741192. doi: 10.1016/j.pmcj.2014.08.009. URL http://dx.doi.org/10.1016/j.pmcj.2014.08.009.

[61] Joseph A Konstan and John Riedl. Recommender systems: From algorithms to user experience, 2012. ISSN 09241868.

[62] Ioannis Arapakis, Joemon M. Jose, and Philip D. Gray. Affective feedback: an investigation into the role of emotions in the information seeking process. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, (January):395–402, 2008. doi: 10.1145/1390334.1390403. URL http://doi.acm.org/10.1145/1390334.1390403.

[63] LDOS-CoMoDa dataset — LUCAMI. URL https://www.lucami.org/index.php/research/ldos-comoda-dataset/.

[64] William H Mitchell and Mitchell Software. UML , the Unified Modeling Language What is UML ? *Engineering*, pages 1–28, 2003. URL http://www.uml.org/what-is-uml.htm.

[65] Chi Yo Huang and Yu Sheng Kao. UTAUT2 Based Predictions of Factors Influencing the Technology Acceptance of Phablets by DNP. *Mathematical Problems in Engineering*, 2015, 2015. ISSN 15635147. doi: 10.1155/2015/603747.

[66] Margaret Rouse. What is Internet? - Definition from WhatIs.com, 2014. URL http://whatis.techtarget.com/definition/phablethttp://searchwindevelopment.techtarget.com/definition/Internet.

[67] Malik Bader Alazzam, Abd Samad Hasan Basari, Abdul Samad Sibghatullah, Yousif Monadhil Ibrahim, Mohamad Raziff Ramli, and Mohd Hariz Naim. Trust in stored data in EHRs acceptance of medical staff: Using UTAUT2. *International Journal of Applied Engineering Research*, 11(4):2737–2748, 2016. ISSN 09739769.

[68] Vinodh Krishnaraju, Saji K. Mathew, and Vijayan Sugumaran. Web personalization for user acceptance of technology: An empirical investigation of E-government services. *Information Systems Frontiers*, 18(3):579–595, 2016. ISSN 15729419. doi: 10.1007/s10796-015-9550-9. URL http://dx.doi.org/10.1007/s10796-015-9550-9.

[69] Oliver Heger and Henrik Kampling. Towards a Theory of Trust-Based Acceptance of Affective Technology. 2016.

[70] Joel Jarvinen, Roope Ohtonen, and Heikki Karjaluoto. Consumer Acceptance and Use of Instagram. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, number 2, pages 2227–2236, 2016. ISBN 978-0-7695-5670-3. doi: 10.1109/HICSS.2016.279. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7427462http://ieeexplore.ieee.org/document/7427462/.

[71] Elsa Macías, Alvaro Suárez, Raquel Lacuesta, and Jaime Lloret. Privacy in Affective Computing based on Mobile Sensing Systems. *Proceedings of 2nd International Electronic Conference on Sensors and Applications*, (November):E016, 2015. doi: 10.3390/ecsa-2-E016. URL http://sciforum.net/conference/ecsa-2/paper/3238.

[72] Gitte Lindgaard. Adventurers versus nit-pickers on affective computing. *Interacting with Computers*, 16(4):723–728, 2004. ISSN 09535438. doi: 10.1016/j.intcom.2004.06.006.

[73] Y.-Y. Wang, A.M. Townsend, A.W. Luse, and B.E. Mennecke. The determinants of acceptance of recommender systems: Applying the UTAUT model. *18th Americas Conference on Information Systems 2012, AMCIS 2012*, 3:2238–2246, 2012.

[74] Tiago Oliveira, Miguel Faria, Manoj Abraham Thomas, and Aleš Popovič. Extending the understanding of mobile banking adoption: When UTAUT meets TTF and ITM. *International Journal of Information Management*, 34(5):689–703, 2014. ISSN 02684012. doi: 10.1016/j.ijinfomgt.2014.06.004.

[75] Nicholas Jones and Pearl Pu. User Technology Adoption Issues in Recommender Systems. *Proceedings of NAEC, ATSMA*, (January 2007):339–379, 2007.

[76] Benjamin Lieberman. UML Activity Diagrams : Detailing User Interface Navigation, 2004. URL https://www.ibm.com/developerworks/rational/library/4697.htmlhttp://www.ibm.com/developerworks/rational/library/4697.html.

[77] Rana el Kaliouby: This app knows how you feel – from the look on your face — TED Talk Subtitles and Transcript — TED.com. URL https://www.ted.com/talks/rana{_}el{_}kaliouby{_}this{_}app{_}knows{_}how{_}you{_}feel{_}from{_}the{_}look transcript?language=en.

[78] Eyeris. EmoVu emotion recognition software, 2016. URL http://emovu.com/e/.

[79] Face Recognition, Emotion Analysis & Demographics — Kairos. URL https://www.kairos.com/.

[80] crowdemotion. URL http://www.crowdemotion.co.uk/.

[81] Jim. Conallen. *Building Web applications with UML*. Addison-Wesley, 1999. ISBN 0-201-61577-0. URL http://dl.acm.org/citation.cfm?id=560225.

[82] Malik B Alazzam, A B D Samad, Hasan Basari, and Abdul Samad Sibghatullah. 42. EHRS ACCEPTANCE IN JORDAN HOSPITALS BY UTAUT2.pdf. 78(3):473–482, 2015.

[83] Pearl Pu and Li Chen. A user-centric evaluation framework of recommender systems. *CEUR Workshop Proceedings*, 612:14–21, 2010. ISSN 16130073. doi: 10.1145/2043932.2043962.

[84] YouTube video category name and id list — Vevlo. URL http://vevlo.com/youtube-video-category-name-and-id-list/.

[85] • YouTube: hours of video uploaded every minute 2015 — Statistic. URL https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/.

[86] Jose San Pedro, Stefan Siersdorfer, and Mark Sanderson. Content redundancy in YouTube and its application to video tagging. *ACM Transactions on Information Systems*, 29(3):1–31, 2011. ISSN 10468188. doi: 10.1145/1993036.1993037.

[87] YouTube Data API (v3) - YouTube — Google Developers, 2013. URL https://developers.google.com/youtube/v3/.

[88] (21) Popular on YouTube - YouTube. URL https://www.youtube.com/channel/UCF0pVplsI8R5kcAqgtoRqoA.

[89] Nikolaos Polatidis and Christos K. Georgiadis. A dynamic multi-level collaborative filtering method for improved recommendations. *Computer Standards and Interfaces*, 51:14–21, 2017. ISSN 09205489. doi: 10.1016/j.csi.2016.10.014.

[90] Python Software Foundation. Welcome to Python.org, 2016. URL https://www.python.org/https://www.python.org/about/.

[91] python-recsys 0.2 : Python Package Index, . URL https://pypi.python.org/pypi/python-recsys.

[92] Recommender Systems Framework in Python — scikit-recommender v0.1 documentation. URL http://muricoca.github.io/crab/.

[93] Django Software Foundation. Django: The Web framework for perfectionists with deadlines. *Djangoproject.Com*, pages 1–3, 2013. URL https://www.djangoproject.com/.

[94] Jo Hayward. *Django Javascript integration: AJAX and jQuery*. Packt Pub, 2010. ISBN 9781849510349. URL https://books.google.dk/books?id=IaD7t{_}BOBVwC{&}pg=PT230{&}lpg=PT230{&}dq=Create,Read,+Update+and+Delete+(CRUD)+philosophy{&}source=bl{&}ots=T9EOFxJUh4{&}sig=tL45Ny38OeLyb8Yw-fq83pHHekE{&}hl=es{&}sa=X{&}ved=0ahUKEwi9jZOflp{_}UAhWIhSwKHeNEBFQQ6AEIKzAA{#}v=onepage{&}q=Create{%}2CRea.

[95] Jacob Kaplan-Moss Adrian Holovaty. The Definitive Guide to Django: Web Development Done Right. (August):2004, 2011. ISSN 0196-6553. doi: 10.1002/ejoc.201200111.

[96] HTML5, . URL https://www.w3.org/TR/html5/.

[97] Bootstrap. Bootstrap · The world's most popular mobile-first and responsive front-end framework., 2016. URL http://getbootstrap.com/.

[98] JavaScript. URL https://www.javascript.com/.

[99] Premium Bootstrap Themes and Templates: Download @ Creative Tim. URL https://www.creative-tim.com/.

[100] Sarah Mei. Why You Should Never Use MongoDB, 2013. URL http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb/http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb.

[101] Rest From. *REST: From Research to Practice*. 2011. ISBN 978-1-4419-8302-2. doi: 10.1007/978-1-4419-8303-9. URL http://link.springer.com/10.1007/978-1-4419-8303-9.

[102] Wolfgang Woerndl, Christian Schueller, Rolf Wojtech, and Unternehmertum Gmbh. A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications. pages 871–878, 2007.

[103] Oliver Oechslein, Mario Haim, Andreas Graefe, Thomas Hess, Hans Bernd Brosius, and Anton Koslow. The digitization of news aggregation: Experimental evidence on intention to use and willingness to pay for personalized news aggregators.

Proceedings of the Annual Hawaii International Conference on System Sciences, 2015-March:4181–4190, 2015. ISSN 15301605. doi: 10.1109/HICSS.2015.501.

[104] Django Packages : Workers, Queues, and Tasks. URL https://djangopackages.org/grids/g/workers-queues-tasks/.

[105] Ask Solem. First steps with Django — Celery 3.1.18 documentation. URL http://docs.celeryproject.org/en/latest/django/first-steps-with-django.htmlhttp://celery.readthedocs.org/en/latest/django/first-steps-with-django.html.

[106] Executive Summary. RabbitMQ – Messaging that just works. *October*, (October), 2009. URL http://www.rabbitmq.com/.

[107] Using RabbitMQ — Celery 4.0.2 documentation. URL http://docs.celeryproject.org/en/latest/getting-started/brokers/rabbitmq.html{#}broker-rabbitmq.

[108] Capturing live video. URL https://helpx.adobe.com/adobe-media-server/dev/capturing-live-video.html.

[109] HTML5 vs. Apps: Here's Why The Debate Matters, And Who Will Win, . URL http://www.businessinsider.com/html5-vs-apps-why-the-debate-matters-and-who-will-win-2012-11?r=US{&}IR=T{&}IR=Thttp://www.businessinsider.com/html5-vs-apps-heres-why-the-debate-matters-and-who-will-win-2012-12.

[110] Get ready for plug-in free browsing (Internet Explorer). URL https://msdn.microsoft.com/en-us/library/hh968248(v=vs.85).aspx.

[111] Chris Hoffman. Why Browser Plug-Ins Are Going Away and What's Replacing Them, 2014. URL https://www.howtogeek.com/179213/why-browser-plug-ins-are-going-away-and-whats-replacing-them/http://www.howtogeek.com/179213/why-browser-plug-ins-are-going-away-and-whats-replacing-them/.

[112] Eric Bidelman. Capturing Audio & Video in HTML5, 2012. URL https://www.html5rocks.com/en/tutorials/getusermedia/intro/http://www.html5rocks.com/en/tutorials/getusermedia/intro/.

[113] Travis Leithead. MediaStream Recording. (September):1–10, 2013. URL https://www.w3.org/TR/mediastream-recording/{#}example2.

[114] Record Audio and Video with MediaRecorder — Web — Google Developers. URL https://developers.google.com/web/updates/2016/01/mediarecorder.

[115] RecordRTC: Index, . URL http://recordrtc.org/.

[116] RecordRTC — WebRTC Audio+Video+Screen Recording, . URL https://www.webrtc-experiment.com/RecordRTC/{#}other-demos.

[117] Unknown. w3counter - global web stats. URL https://www.w3counter.com/globalstats.phphttp://www.w3counter.com/globalstats.php?year=2012{&}month=4.

[118] PyCharm. URL https://www.jetbrains.com/pycharm/.

[119] DigitalOcean. DigitalOcean Cloud computing designed for developers, 2016. URL https://www.digitalocean.com/.

[120] Download PuTTY - a free SSH and telnet client for Windows. URL http://www.putty.org/.

[121] RealVNC. Download VNC Viewer. URL https://www.realvnc.com/download/viewer/.

[122] Domain Names - Cheap Domain Names — Namecheap.Com. URL https://www.namecheap.com/.

[123] How To Install an SSL Certificate from a Commercial Certificate Authority — DigitalOcean. URL https://www.digitalocean.com/community/tutorials/how-to-install-an-ssl-certificate-from-a-commercial-certificate-authority.

[124] Module Index, Advanced Tutorials, Raw Sql, Decorators Reference, Storage Api, Generating Csv, and Generating P D F Middleware. Django documentation — Django documentation — Django. pages 4–7. URL https://docs.djangoproject.com/en/1.11/https://docs.djangoproject.com/en/1.5/.

[125] Django Software Foundation. Cross Site Request Forgery protection, Django Documentation. URL https://docs.djangoproject.com/en/1.11/ref/csrf/https://docs.djangoproject.com/en/dev/ref/contrib/csrf/.

[126] Ken Kwong-kay Wong. Partial Least Squares Structural Equation Modeling (PLS-SEM) Techniques Using SmartPLS. *Marketing Bulletin*, 24(1):1–32, 2013. ISSN 0113-6895. doi: 10.1108/EBR-10-2013-0128. URL http://marketing-bulletin.massey.ac.nz/v24/mb{_}v24{_}t1{_}wong.pdf.

[127] Christian M. Ringle, Sven Wende, and Jan-Michael Becker. Smartpls 3. http://www.smartpls.com, 2015.

[128] Berardina De Carolis and Marco De Gemmis. A Multimodal Framework for Recognizing Emotional Feedback in Conversational Recommender Systems. *RecSys EMPIRE 2015: 3rd Workshop on Emotions and Personality in Personalized Systems 2015*, pages 11–18, 2015. doi: 10.1145/2809643.2809647.

[129] Saim Shin, Dalwon Jang, Jongseol J Lee, Sei Jin Jang, and Ji Hwan Kim. MyMusicShuffler: Mood-based music recommendation with the practical usage of brainwave signals. In *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, pages 355–356, 2014. ISBN 9781479912919. doi: 10.1109/ICCE.2014.6776039.

[130] Yong Zheng. Adapt to emotional reactions in context-aware personalization, 2016. ISSN 16130073.

[131] Sven Ewan Shepstone, Zheng-hua Tan, and Senior Member. Using Audio-Derived Affective Offset to Enhance TV Recommendation. 16(7):1999–2010, 2014.

[132] Claudia Orellana-rodriguez and Ernesto Diaz-aviles. Learning to Rank for Joy. pages 569–570.

[133] M. Polignano. The inuence of user's emotions in recommender systems for decision making processes. *CEUR Workshop Proceedings*, 1462:58–66, 2015.