Extract Emotional Tags from Movie Synopses

Dimitris Fanis

This dissertation was submitted in part fulfilment of requirements for the degree of MSc Information Management

Department of Computer & Information Sciences

University of Strathclyde

August 2020

Declaration

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to place a copy of the dissertation in a publicly available archive.

Yes [✔] No [ ]

I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, acronyms & initialisms, table of contents, list of illustrations-tables-equations, and bibliography is: 21,682.

I confirm that I wish this to be assessed as a Type 1 2 3 4 **5**

Dissertation

Signature: Dimitris Fanis

Date: 17/08/2020

Abstract

Context: Although a wide variety of studies has shown a strong correlation between several movies/ movie metadata and emotions that they evoke, however not much research has been conducted for an automatic extraction of emotions, as well as with regard to the relationship of the latter with user ratings.

Objectives/Purpose: The purpose of this project was initially to find or manually label emotional tags respecting movies' content. The ultimate goal would be the prediction of emotions to any dataset which encompasses movie overviews and metadata. This would lead to an automatic emotion identification and generation, a fact which can contribute to a more personalized content from Recommender Systems and advertising companies.

Methods: A wide variety of Natural Language Processing tools were examined, along with a set of machine and deep learning models. Ideally, a movie would have more than one emotional tag, therefore it is about a multilabel classification problem, with the model evaluated using classification accuracy score.

Results: A set of emotional tags was identified and used to label 300 movies. In addition a machine learning model was built which successfully predicted the emotion of 55,577 unlabeled movie data with regard to their emotional tags.

Conclusions: Various correlation tests were conducted which indicated a relationship among user ratings and the respective emotions evoked from those voted movies. Overall, it was shown that the notion of emotion in the movies industry can be an important feature to be utilized from Recommender Systems and to be further examined.

Acknowledgments

Extend warm thanks to my supervisor Mr. William Wallace for his cooperation and pieces of advice and guidance throughout the dissertation phase. I would also like to thank all professors of Strathclyde University for giving me the chance to enhance my knowledge in a wide variety of sectors in the department of Computers & Information Sciences

In addition, I want to thank Mohamed Amine Belabbes, PhD student at Strathclyde University. He was my lab demonstrator during the class module “CS982 Big Data Technologies”, and shared with me useful resources about the dissertation’s topic. He was also willing to help and guide me through the practical part of the thesis.

I also want to thank Benedikt Muiler, MSc student in Data Analytics at Strathclyde University, who shared his code with regard to our common assignment on the second part of the CS985 class in Deep Learning, and I had the chance to study it.

Lastly, I thank my family for their support during my postgraduate studies, and especially my sister for her support and pieces of advice.

Acronyms & Initialisms

API	Application Program Interface
AUC	Area Under (the) Curve
AUROC	Area Under (the) Receiver Operating Characteristics
CDRS(s)	Cross-Domain Recommender System(s)
CNN(s)	Convolutional Neural Network(s)
DNN(s)	Deep Neural Network(s)
EARs	Emotion Aware Recommender Systems
EDA	Exploratory Data Analysis
GAN(s)	Generative Adversarial Networks GAN(s)
GloVe	Global Vectors for Word Representation
IDF	Inverse Document Frequency
IMDb	Internet Movie Database
LDA	Latent Dirichlet Allocation
LSTM	Limited Short-Term Memory
MLP	Multi-layer Perception
NFL	No Free Lunch (theorem)
NLKT	Natural Language Toolkit
NLP	Natural Language Processing
NMF	Non-negative Matrix Factorization
POS	Part of Speech
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic (curve)
RQ	Research Question
RS(s)	Recommender System(s)
SGD	Stochastic Gradient Descent
SVC	Support Vector Classifier
TF	Term Frequency
TF-IDF	Term Frequency – Inverse Document

TMDb	The Movie Database
VADER	Valence Aware Dictionary for sEntiment Reasoning

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1|

1. Introduction

1.1Research Context

The main purpose of this dissertation is to build a model that could extract emotional tags from movie plots. Identifying emotional information located in movie or television show abstracts could have various and useful applications in recommender systems (RSs) for finding similar content, or in advertising for placing advertisements.

It falls into the field of Natural Language Processing (NLP) including techniques such as entity extraction and sentiment analysis, which could be deployed along with machine and deep learning tools. It is also closely related to folksonomication: A folksonomy is a collaborative tagging system where various pieces of information -in this case regarding movies or TV shows- are displayed, such as genres, plots, metadata and tags (Kar, Maharian & Solorio, 2018:p.2879), and therefore, the dissertation's research could have various applications in this field.

1.2 Research Problem - Domain

The production of the model would presuppose the existence of some emotional classes-labels. The nature and definition of emotion detection have not strictly been defined and various researchers have proposed several emotional categories. One of the most proposed is that of Ekman's six discrete emotional classes (Ekman, 1992; Lazemi & Ebrahimpour-Komleh, 2016; Tkalčič *et al.*, 2016) because they can be easily linked with facial communication and expressions, and these are: happiness, anger, sadness, fear, disgust, and surprise (Ekman, 1992; Farzindar & Inkpen, 2015:p.50). Other emotional tags have also been proposed for tasks such as investigating the emotion dimensions of users' comments. In this case, the emotional categories often are: sensitivity, aptitude, attention and pleasantness (Topal, Koyutürk & Özsoyoğlu, 2017:p.2). Lastly, emotional inclinations have also been distributed to six categories as love, joy, anger, surprise sadness, and fear (Chakraverty & Saraswat, 2017:p.21).

Furthermore, movies with law popularity may present few or no tags. Based on a study in 2018, at least 34% of movies in IMDd include no tags (Kar, Maharian & Solorio, 2018:p.2879), and therefore, it is strongly believed that an automatic extraction of tag information, such as emotional tags from abstracts, can help many movies to be discovered. The above problem, i.e.

when a movie does not have any tag from the past, can be called “incompleteness in tag spaces” (Kar, Maharian & Solorio, 2018:p.2885), and hence it can be considered as one more challenge addressed when building a tagging model for movies with no “emotional” tag history.

What makes a movie e.g. suspenseful, dramatic or sci-fi, and which words hidden in movie plots can predefine this? The answer to that would help extract significant information from narrative texts and build an automatic system that could produce emotional tags (Kar *et al.*, 2018:p.7).

1.3 Research Challenges

Studies have indicated that there is a strong correlation between movie genres (e.g. action, comedy) and emotions they evoke (Chakraverty & Saraswat, 2017:p.10), and as a consequence, their corresponding emotional tags. Even though one important aim of cross-domain recommender systems (CDRSs) is to alleviate the cold start problem (e.g. new users) and the data sparseness or scarcity (Allison, Guthrie & Guthrie, 2006:p.327) by transferred learning (Sahu, Dwivedi & Kant, 2018:p.630), and although, at the same time, the power of emotion has been reaffirmed, based on the literature the existence of emotions in movie plots has not been utilized extensively from CDRSs. Finally, not much research has been conducted for the examination of the relationship between user ratings and the emotional content of movies (Chakraverty & Saraswat, 2017:p.4).

In addition, the biggest challenges of the project would be the definition of emotional labels to be used, and how they could be reproduced in case of no available datasets with labeled emotional tags. Furthermore, challenges of how a ML model could be built for the prediction of emotions should be addressed, as well as finding out which features should be used for those predictions, i.e. only movie plots or more movie metadata as well.

1.4 Research Objectives & Deliverables

The final aim is the construction of an ML model, capable of automatically extracting emotional information from movies. Having extracted those pieces of emotional meta information, the model would then be evaluated in its predictions for discovering its accuracy levels. However, in order the above to be implemented, some existed training data containing emotions should be found, otherwise, manual labeling of emotions should take place. In the scenario of the latter, the researcher will have to pre-define a set of emotions that will be used during the model production. Based on related studies in the context of folksonomication (Kar,

Maharjan & Solorio, 2018), one of the limitations and hence one challenge for this thesis is to build a model that will be able to generate tags even for movie synopses composed of a small number of sequences in text.

Some of the general contributions of this work can be attributed to the improvement of social tagging in the context of movies' metadata. The model aimed to be built, not only would help viewers know in advance what emotions they could expect from a movie, but this concurrently would enhance RSs by refining the retrieval of similar movies/TV shows. An automatic tagging system of this spectrum could also have various applications in several domains of multimedia RSs, and in general, it could be adapted in every field that would wish the extraction of useful emotional information from a corpus of unstructured data. In addition, this could help RSs recommend more personalized and purposeful items (in this case movies) to their users. Furthermore, advertising companies could benefit by delivering addressable advertising (Tapidze, 2017) with more relevant and precise content, where different ads can be broadcasted to different consumers based on a predefined audience segmentation (Invidi, 2018).

1.5 Research Questions

The research questions for this dissertation are the following:

1. What a sufficient number of emotional tags could be (and why) and which these tags are?
2. Which models are suitable for the production of those emotional tags, and is an automatic generation of that emotional information possible?
3. Is there any correlation among user preferences and emotions expressed in the tv show/movie plots? Is, in the end, the notion of emotion a useful feature in the context of recommender systems and advertising companies?
4. Is there any correlation in the series of movies in users' watchlist with regard to the underlying emotions that these movies elicit? In other words and for example, if a viewer has watched 100 movies, are the depicted emotions of the first 50 movies correlated with those of the subsequent 50 movies?

1.6 Chapters' Overview

The following chapter provides the literature background of the project's topic. After that, the chapters are divided into two main parts: chapter three and four are with regard to all processes conducted for and until the final model's production, while in the fifth chapter, correlation tests are constructed to address the research questions three and four. Finally, the

last chapter ends up with the overall conclusions and recommendations. It is noted that some data analysis conducted in chapter five would not be possible to be implemented in previous chapters, since it requires the existence of emotional tags which will not have been found and produced until chapter four.

Chapter 2|

2. Literature Review & Related Work

2.1 What is Emotion & Categories of Emotion

2.1.1Terminology & Physiology of Emotions

The exploration of emotions can be useful in the movie industry, and by extension, in every organization which interact with humans, in that they can be indicators of users' behaviour and preferences (Mizgajski & Morzy, 2019:p.345). Emotions can be considered as mental states which represent evaluative reactions towards events or objects which vary in intensity levels (Nabi, 2010:p.153), and which can influence humans' behaviour (Vlad, 2020:p.119). From another perspective, emotions can be viewed as "complex interactive entities encompassing subjective and objective factors consisting of affective, cognitive, conative, and physiological components" (Wirth & Schramm, 2005:p.3).

Table 2.1 illustrates details about the physiology of emotions, which can be considered important in the context of constructing the emotional tags for this project.

Components	Description	Manifestation
Affective	Subjective experience of situations	Feelings of arousal, pleasure, or dissatisfaction
Cognitive	How aspects of experiences/situations relative to emotions are perceived and evaluated	For example, one song or movie which is objectively considered as happy, might be neutral or sad for someone else
Conative	Expressive behaviour	Facial expression Vocal expression Gestures
Physiological	Peripheral reactions of body Nervous system (physiological arousal)	Blushing Heart rate changes Respiratory rate changes Sweaty hands

Table 2. 1: Components of Emotions

Based on the literature, one basic common section of studies indicate that emotions are real psychological entities, and a finite number of them can universally be found in all human beings (Oatley, 1992:p.162). However, this finite number may not be stable and it depends on several variables taken into account from the researcher's hypotheses and perspectives.

Several debates have taken place when researchers have tried to categorize a specific number of basic emotions. Some of the main problems found here are that emotions can be

deemed as an elaboration or combination of other basic emotions, and the initial criteria that the researcher sets might not agree with the criteria found on primary emotions. For instance, if a researcher accepts that basic emotions should not necessarily be connected with facial expressions, then he/she would not agree with Ekman's six classes of emotions (Oatley, 1992:p.162). Some theorists, in addition, might support that e.g. both happiness and relief are basic emotions while another portion of researchers would argue that relief is a subcategory of the basic "happiness" emotion (Oatley, 1992:p.166; Wirth & Schramm, 2005:p.6).

2.2 Emotional Classes

2.2.1 Emotions & Other Affective Traits

Before attempting to categorize emotions into several classes, one important note when eliciting emotions is the disjunction between emotion and other affective phenomena such as mood, emotional traits, and emotional disorders (Ekman, 1992:p.174), in that they differ in the scope of their duration and physiology (Ekman, 1992; Innes-Ker, 2015). Emotions, for example, are brief in duration lasting between seconds and hours, contrary to moods such as apprehension, dysphoria and euphoria (Ekman, 1992:p.194), which may last hours or days (Ekman, 1992:p.186).

Attention is also required in the difference between emotions and feelings: feelings are perceived as a self-perception of certain emotions and they constitute a subjective expression of them (Bitbrain, 2019). Comparing, lastly, personality with emotions, one should have in mind that "personality is to emotions what climate is to weather" (Favaretto, Musse & Costa, 2019:p.30).

2.2.2 Generation & Categorization of Emotions

In the context of identifying the generation of emotions, there seem to be two perspectives (Wirth & Schramm, 2005:p.6), the evolutionary and the empirical. The former states that some emotions, the so-called "specific", or "evolutionary" emotions cannot be further analyzed to other subtypes of emotions. On the other, the so-called "secondary'", "mixed", or "complex" emotions come from the primary evolutionary emotions. For example, 'pleasantness' could be considered as a secondary emotion after the primary "happiness" emotion. On the other hand, the empirical perspective tries to identify the ratio of similarity between certain emotions. Empirical studies seem to have four common categories of emotions and these are: anger, anxiety, sadness, and joy.

After the study of emotions' generation, and based on the literature, there seem to be two main approaches/philosophies regarding their categorization (Wirth & Schramm, 2005; Mauss & Robinson, 2009; Gross & Levenson, 1995; Nabi, 2010; Mizgajski & Morzy, 2019). One the one hand, the first portion of researchers advocates that there is a discrete number of distinct

basic emotions, or a combination of them (Mizgajski & Morzy, 2019:p.355), while other researchers suggest that emotions cannot be strictly defined by specific and unaltered patterns of people's behaviour, physiology, and experiences. Rather, they conclude that measurements of emotions can be found along dimensions, such as arousal and valence, rather than in discrete emotional states such as happiness, sadness, or anger.

The first approach refers to categorical emotional states which originate from cognitive appraisals or evaluations of situations (Nabi, 2010:p.154), formulating the universal emotions model (Mizgajski & Morzy, 2019:p.355). More specifically, these emotional classes refer to affective states which can reflect changes in facial expression, cognitive (perceptual) activity, subjective experience, action tendency, and psychological response (Dillard & Meijnders, 2012:p.65). Ekman's six emotional classes, can be associated with people's facial expressions and can reflect people's response towards environmental events (Gross & Levenson, 1995:p.88). These categories are (in alphabetical order): anger, disgust, fear, happiness, sadness, and surprise. They comprise the most widely used categories for the fulfilment of emotional classification, and in addition, they can be extended into twelve possible emotions, which are composed of: anger, awe, contempt, disgust, embarrassment, excitement, fear, guilt, interest, sadness, shame, and surprise (Ekman, 1992:p.193). Some theories, finally, add a "neutral emotional state" (Gross & Levenson, 1995) along with Ekman's six basic emotions, reaching the total of seven.

To an extension of this approach, another popular classification is that of Chakraverty et al. Inclinations here have also been distributed to six categories being: love, joy, anger, surprise sadness, and fear (Chakraverty & Saraswat, 2017:p.21). Another set of emotions comes from Plutchik (Plutchik, 2001) who identifies eight basic emotions (anger, anticipation, disgust, fear, joy, trust, sadness, and surprise). Specifically for movies industry, after a five-year research using 250 films, eight emotional states were found, and these are: amusement, surprise, contentment, a relatively neutral state, disgust, anger, sadness, and fear (Gross & Levenson, 1995:p.87).

Proceeding to the second approach, some other theories so-called dimensional theories of emotions have additionally been investigated. These theories conceptualize emotions as general motivational states which can be found in broad affective dimensions (Nabi, 2010; Mizgajski & Morzy, 2019). Several dimensional theories exist, such that of Lazarus et al, of Russell, and of Desmet and Hekkert (Mizgajski & Morzy, 2019:p.356), nonetheless, one of the most popular among them is the study of Osgood et al. (Odić *et al.*, 2013:p.15) which is related to emotion research in the context of Neuropsychology and Physiology. Each emotion here is assigned to three dimensions, and these are: dominance, arousal/intensity, and

pleasantness/valence (Odić *et al.*, 2013; Gross & Levenson, 1995). Other dimensional theories, in addition, consider only two dimensions consisted of arousal and valence (Shapiro, MacInnis & Park, 2002; Wirth & Schramm, 2005). The latter dimensional theory has also been successfully used in movie industry projects, and studies have shown that by using attributes per dimension can give greater insights regarding the effects that each dimension provides. These attributes are: high, neutral, and low for the arousal dimension, and positive, neutral, and negative for the valence level (Shapiro, MacInnis & Park, 2002:p.24).

Figure 2.1 summarizes some of the most popular categorizations of emotions that were also analyzed above:

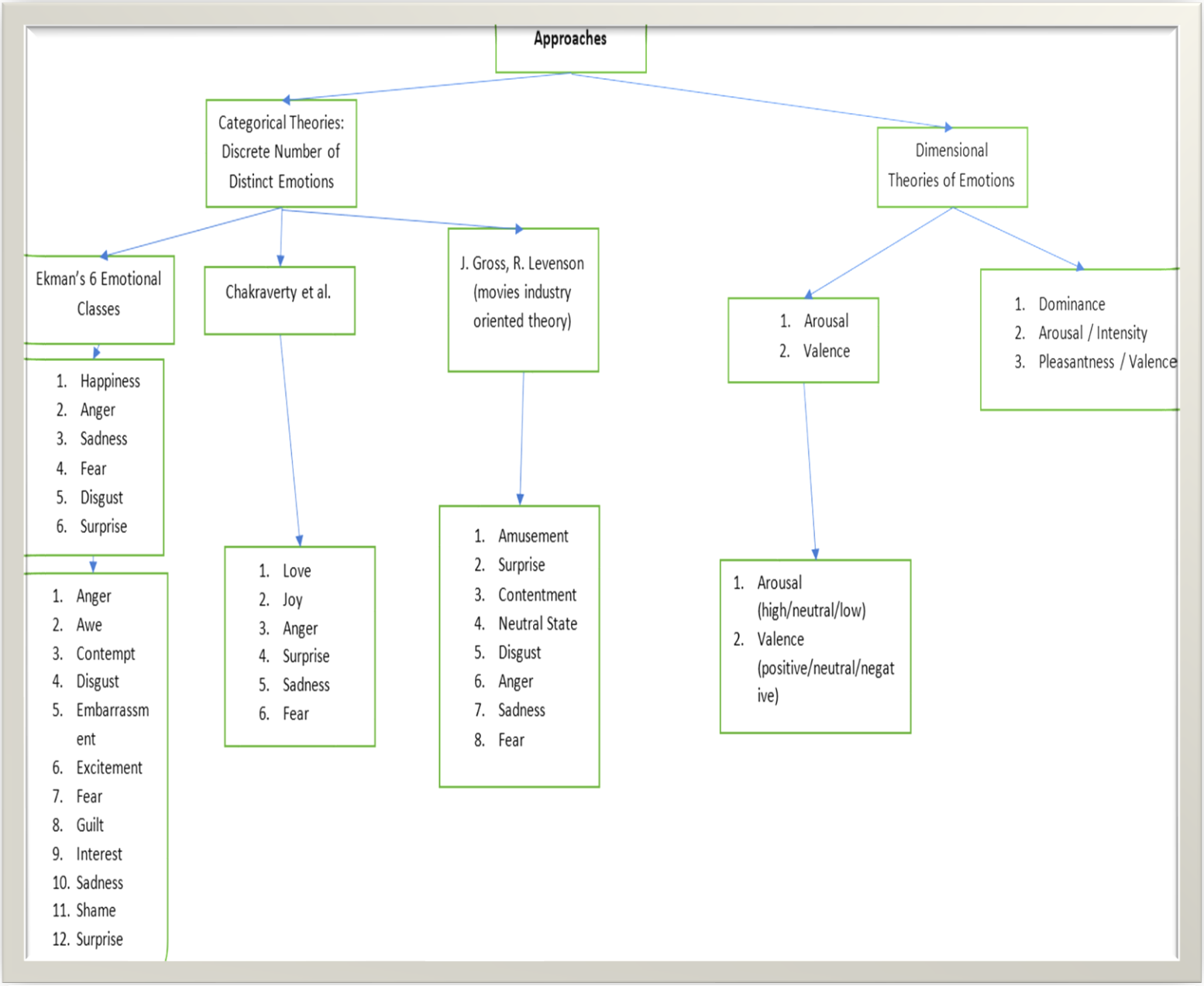

Figure 2. 1: Classes of Emotional States

2.2.3 Comparison & Evaluation of Emotional Theories

In movies industry, both categorical and dimensional theories of emotion have been used for emotional elicitation (Innes-Ker, 2015:p.59). Attempting to conduct a comparison between the two basic emotional theories, it is concluded that the theory of discrete emotions can be more useful and beneficial, at least for the context of this project. This can be supported by the fact that this approach not only embraces the dimensional perspectives of arousal and valence (by assessing them), but it also goes one step further by producing further significant information illustrating additional components of emotional states (Nabi, 2010:p.154).

2.3 Related Work

One related work can be considered the "tag genome" in the "Movie Tuner" application using data from the MovieLens database (Vig, Sen & Riedl, 2012). It is about an advanced tagging model in the context of collaborative filtering which computes the tag relevance using tag applications, text reviews, and ratings in its training model (Vig, Sen & Riedl, 2012:p.4). One of its useful applications is that it can also be applied to items that had no previous tags in the past, alleviating the data sparseness in items (Vig, Sen & Riedl, 2012:p.40) and the new item cold-start problem in movie recommendations (Lops *et al.*, 2019:p.245).

Since the notion of emotion is a core element in this dissertation, related work is, also, considered the construction of "affective recommender systems" which is a relatively new field, and tries to encompass emotions in the area of content recommendations (Mizgajski & Morzy, 2019:pp.372–373). Another closely related work is that of "eDNA" which refers to an emotional DNA[1], a work by the Magid Company[2]. The emotional DNA is based on the primary idea that emotions drive behaviour, and that audience's emotional state can be successfully associated with their TV program choices. This could achieve emotional tonality and emotional alignment in broadcast content (Magid, 2019). The eDNA emotional dimensions identified and linked with the viewers' emotional states are: edge, passion, smarts, adrenaline, originality, heart, relatability, and gravity (Magid, 2020). Disney ABC Television Group, for example, has used the aforementioned eDNA emotional dimensions and improved its content quality, whereas the right connection between viewers' emotions and TV shows produced successfully targeted ads and gave higher profits (Weisler, 2018).

Finally, closely related to this thesis is the research in the context of folksonomy based on tag recommendation for collaborative tagging systems (Font, Serrà & Serra, 2013), whose aim is to solve the problem of tag scarcity and/or ambiguity. One such work is that of Kar et al. which predicts tags using the emotion flow encoded neural network (Kar, Maharian & Solorio, 2018). The rationale of this related work is quite close to that of this project, however the tags produced are pre-defined tags voted by users and not emotional tags.

[1] https://magid.com/emotional-dna/
[2] https://magid.com/

Chapter 3|

3. Research Methodology, Implementation & Analysis

3.1 Research Approach & Methodology

The research approach will be inductive since a new theory (in this case a machine learning model) is aimed to be built, rather than testing an existing one. It will also be an experiment type of dissertation with applied research for developing techniques that can answer the research questions and achieve the research objectives, i.e. the extraction of a set of emotional tags. Moreover, it will be exploratory research (McCombes, 2020) since the researcher will explore various aspects of the research problem, such as which are the best machine learning techniques for the problem, what data should be used and how, and how the final model can be better generalized for any movie review of any dataset. Finally, various methods are going to be applied: implicit qualitative methods when NLP will be used (exploration of content analysis, sentiment analysis) and quantitative methods such as statistical analysis for the exploration of the data and evaluation of the final model. The following figure depicts the structure and workflow for the following sections:

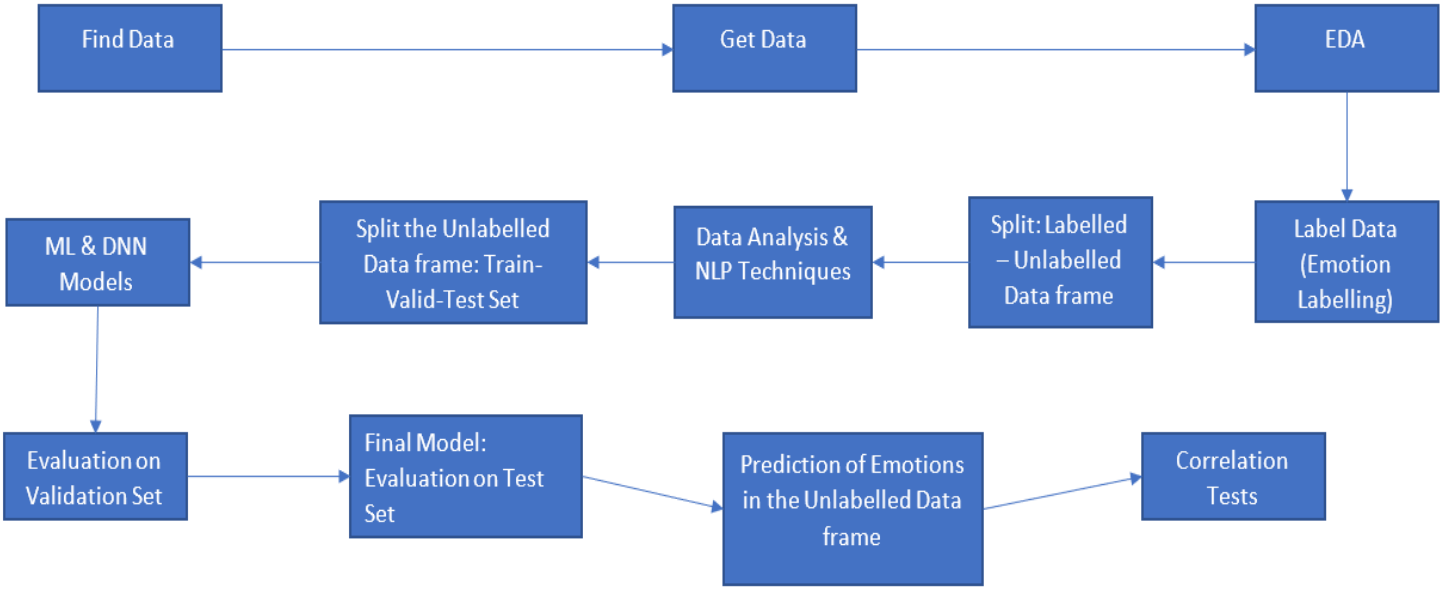

Figure 3. 1: Structure of Tasks & ML Workflow

3.2 What Data Was Used & Why

In order to achieve an efficient result with regard to this project's subject, some data that would encompass movie overviews should be found. Based on various explorations, mainly three data sources were candidate, and these were: the "TMDB 5,000 Movie Dataset" (Kaggle, 2017), which comes from the repository of Kaggle.com[3] and provides movie overviews and metadata[4], secondly via the Application Program Interface (API) of the TMDb database[5], and thirdly, through the GroupLens research lab (GroupLens, 2020) which specializes in RSs and provides its movie database through the MovieLens website (MovieLens, 2020).

Eventually, the second and third were used from the above sources, at the same time and interactively. The reason the first source was not used is that the researcher wanted to gather a greater quantity of unstructured text data than the one provided by that repository (5,000 movies). The fetching of a greater quantity of movie overviews was achieved via the TMDb API. An API request was successfully made and got accepted on 17/06/2020 in that database[6], with an educational type of application request, and stating that the purpose is the analysis of movie overviews for the context of this dissertation. Finally, the movie metadata for these overviews were found through the MovieLens research dataset (Harper & Konstan, 2015b, 2015a; GroupLens/MovieLens, 2020; Vig, Sen & Riedl, 2012), which is provided for free and for new research (MovieLens 25M Movie Ratings)[7]. In terms of research data types, the data used from GroupLens is considered secondary (Surbhi, 2016) because they have already been collected from previous researchers, while that of TMDb is about primary data (Health, 2015), in that the researcher personally collected them for the purpose of this dissertation, and to his best knowledge this unstructured data is not available in any repository or website, at least in this form and/or quantity (number of movie overviews extracted, and a bridge linkage with MovieLens dataset).

3.2.1 Ethical Considerations

There will be no participants in this research since no qualitative research methods are going to be used, such as surveys or questionnaires. The sources of all the data collected from various repositories are referenced both in this paper and in the coding notebooks, and no ethical informed consents were needed.

[3] https://www.kaggle.com/
[4] https://www.kaggle.com/tmdb/tmdb-movie-metadata
[5] https://www.themoviedb.org/
[6] https://developers.themoviedb.org/3/getting-started/introduction
[7] https://grouplens.org/datasets/movielens/

3.3 Gathering the Data & EDA

3.3.1 MovieLens Dataset & TMDb

The MovieLens dataset provides six csv files, and initially the data analysis took place there. These files are with regard to: tags[8], genome tags[9], genome scores[10], links[11], movies[12], and ratings[13]. In the initial data analysis, all csv[14] files turned into dataframes for data exploration. It was noticed that "tags.csv" included an 1:M relationship among users and tags, and this was fixed creating a new dataframe and csv file,[15] and reducing the dataframe's size by 788,019 rows. Genome tags and scores are about relevance scores (in range (0, 1][16]) assigned to tags that have been attributed by users. Later, a new dataframe was created [17] merging the genome tags and scores, and keeping only those tags which had a relevance score more than 0.7[18]. As it can been seen from figure 3.2, the data is skewed to the left and as the relevance score increases, the less the relevant tags we get having a high relevance score. The rationale behind this was to keep tags and metadata which had been voted by users and at the same time had a great relevance score: many users may provide quite irrelevant tags which could feed wrong information to a model, therefore, the genome tag relevance can bring a balance to that.

[8] "tags.csv"

[9] "genome-tags.csv"

[10] "genome-scores.csv"

[11] "links.csv"

[12] "movies.csv"

[13] "ratings.csv"

[14] A wide variety of csv files were created through this project. Some of those may or may not be referenced here depending on their importance or intermediate role, however, they are all provided in the zip file with a README txt notepad describing the content of each file, and referencing the corresponding ipynb notebook to which the imminent files were created.

[15] tags_set.csv.

[16] For a more efficient data explanation, the interval notation followed will be: "[x , y)", where the number in brackets ("x") is inclusive, while that in parentheses exlusive ("y").

[17] genome_score_final_2.csv

[18] The researcher's thought was that tags with score> 0.5 would be an adequate threshold for filtering good tags. Based on studies of the original paper which produced the theory of genome tags, it was decided that 0.7 would both generate relevant tags, and simultaneously it would not lose many tags from the dataset, a fact that would occur if the threshold increased even more, since many tags had a score below that number.

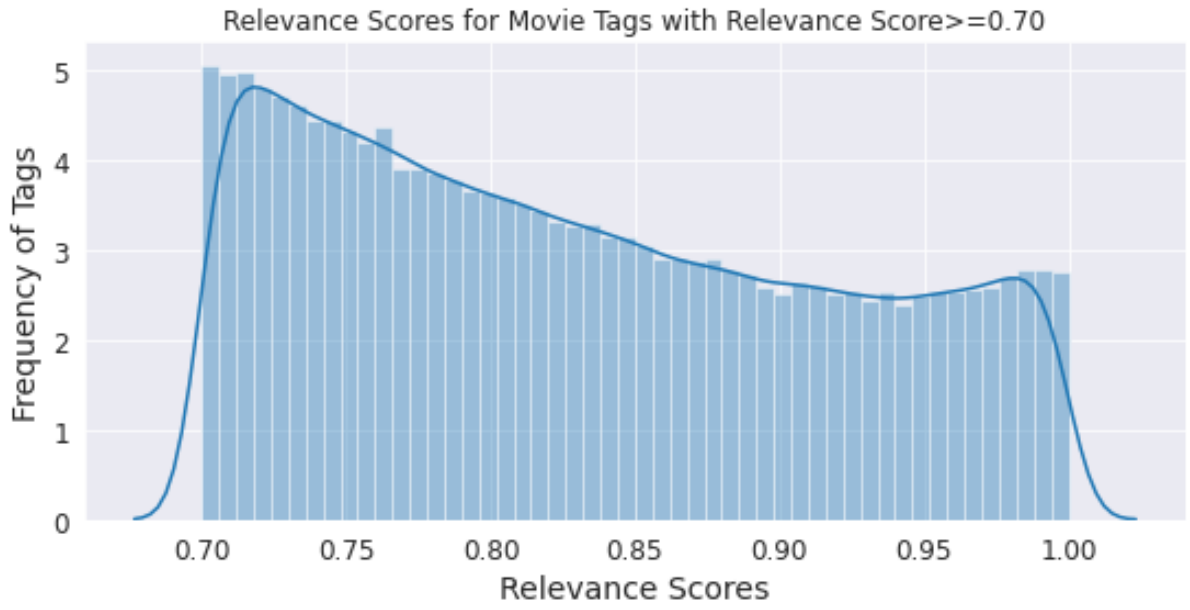

Figure 3. 2: Tag Relevance Scores - MovieLens

For every movie in the dataset at least one movie genre is assigned, whereas every unique movie has a triple id: one linked with the MovieLens database, one with that of IMDb and the third with the TMDb, a fact which was very useful for the actions undertaken subsequently. Figure 3.3 illustrates the allocation of movie genres across the dataset.

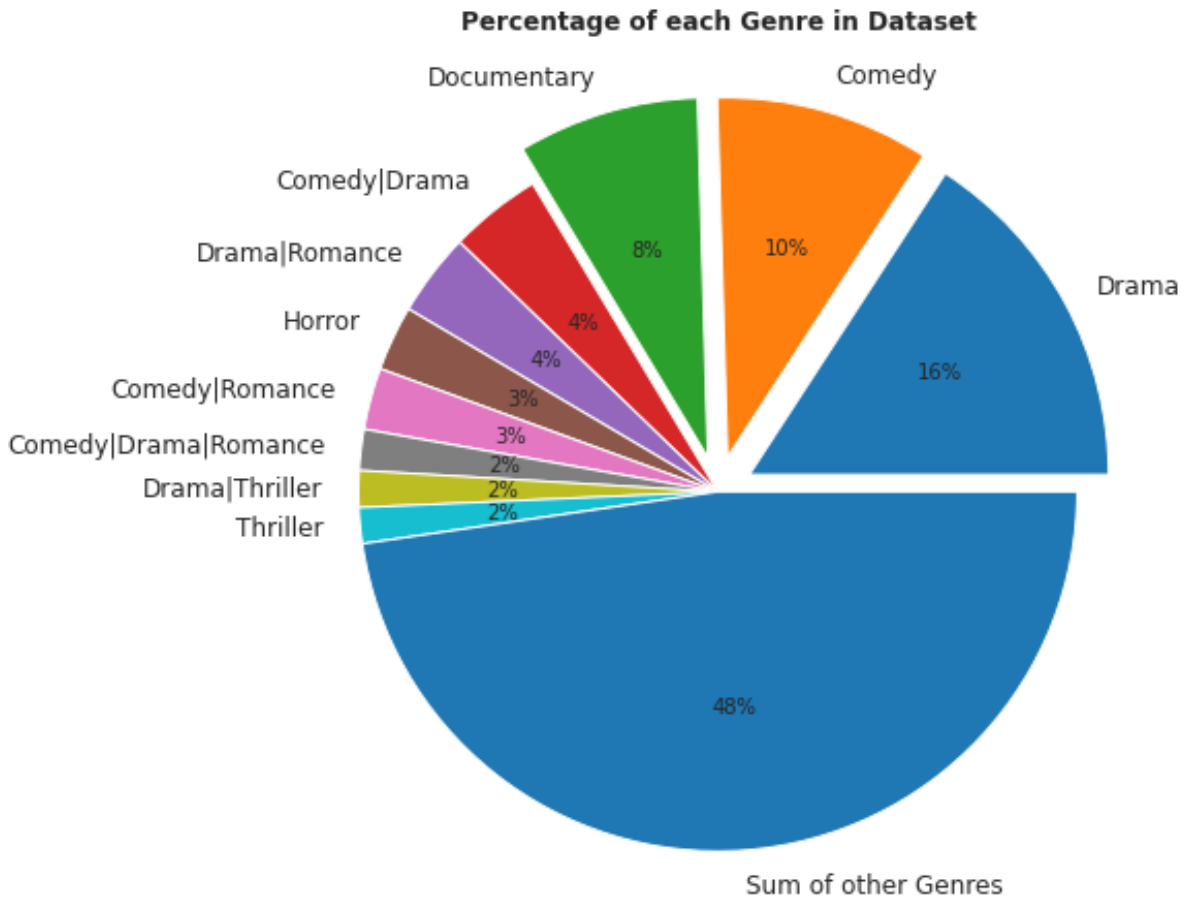

Figure 3. 3: Movie Genres – MovieLens Dataset

At this point, an initial feature engineering took place, by creating three stratums of categories with regard to the tag relevance and the rating votes. These included the creation of the variable "relevance_cat" in range [1 , 3] replacing the scores for tags with relevance score higher than 70%, and the creation of the variable "rating_cat", which stands for rating categories. The original ratings are made on a 5-star scale, with half-star increments, and their overall values are in range [0.5 , 5]. The new introduced variable has values in range [1 , 3], where:

- "1" replaces the values of the original "rating" variable for values in range [0.5 , 2]
- "2" replaces the values of range [2.5 , 3.5], and
- "3", replaces the values of range [4, 5]

The purpose of that was to categorize the movies metadata based on their negative, neutral, and positive rating category, and the benefits from that will be shown in the next sections.

3.3.2 Linkage with the TMBb

Overall, the MovieLens datasets provided a wide variety of useful movies metadata that the researcher has later used for building the necessary models. Nevertheless, there was one

important missing feature, and that was the overview for every movie displayed. For this reason, movie plots were fetched through the TMDb API key for developers mentioned earlier, and were linked with their corresponding movies metadata (title, genres, etc.) via the TMDb link provided by MovieLens for every movie displayed. The CSV file created[19] contains totally 62,324 movie overviews. The other column features of this file come from the merge result of "movies.csv" & "links.csv" provided by MovieLens. However, the final number of movies kept for data analysis was 55,877[20]. The procedure had as followed:

Initially, the movies and movie links provided by GroupLens were merged in a new dataframe based on their distinct movie id. This number was true for 62,423 movies. Nonetheless, some of them had no longer a link to the TMDb and the linkage for those was impossible, hence, a smaller number of 62,324 movies could potentially be linked and extracted through the TMDb database. During the extraction, two main problems occurred: either movies with an available movie id from MovieLens were no longer in TMDb, or some movies existed in TMDb but had no overview displayed available anymore, a fact which caused errors and mismatches during the data extraction, since every movie separately should carefully be downloaded and matched with that of the MovieLens data. The total number of those occurrences was 1,765, and as a consequence, the total number of movies turned to 60,559. After reading the csv file created and assigning it to a dataframe, a deeper data exploration took place. Care for rows containing null values had already taken place from previous steps, however, duplicate values were identified (e.g. repetition of movies/movie ids from MovieLens led to double fetching of overviews from the TMDb). This reduced the number of extracted movie overviews by 4,682 reaching the final of 55,877 distinct movies along with their meta information merged (title, genres, etc.).

3.4 Data Preprocessing & Feature Engineering

Before building any model, an efficient data preprocessing should precede. Some basic preprocessing steps (handling null values, misplaced values, inconsistent shapes when the datasets were merged or concatenated) had already taken place so that the movie overviews' fetching could successfully and correctly be executed.

A useful text data visualization can take the form of word clouds which they cohesively depict the frequency or importance of each word in the dataset (Vu, 2019). Below, figure 3.4 depicts the word cloud of all the dataset:

[19] "final_8.csv": The movie plots were fetched in 8 iterations in 8 parts to avoid RAM crush problems
[20] "movies_final.csv".

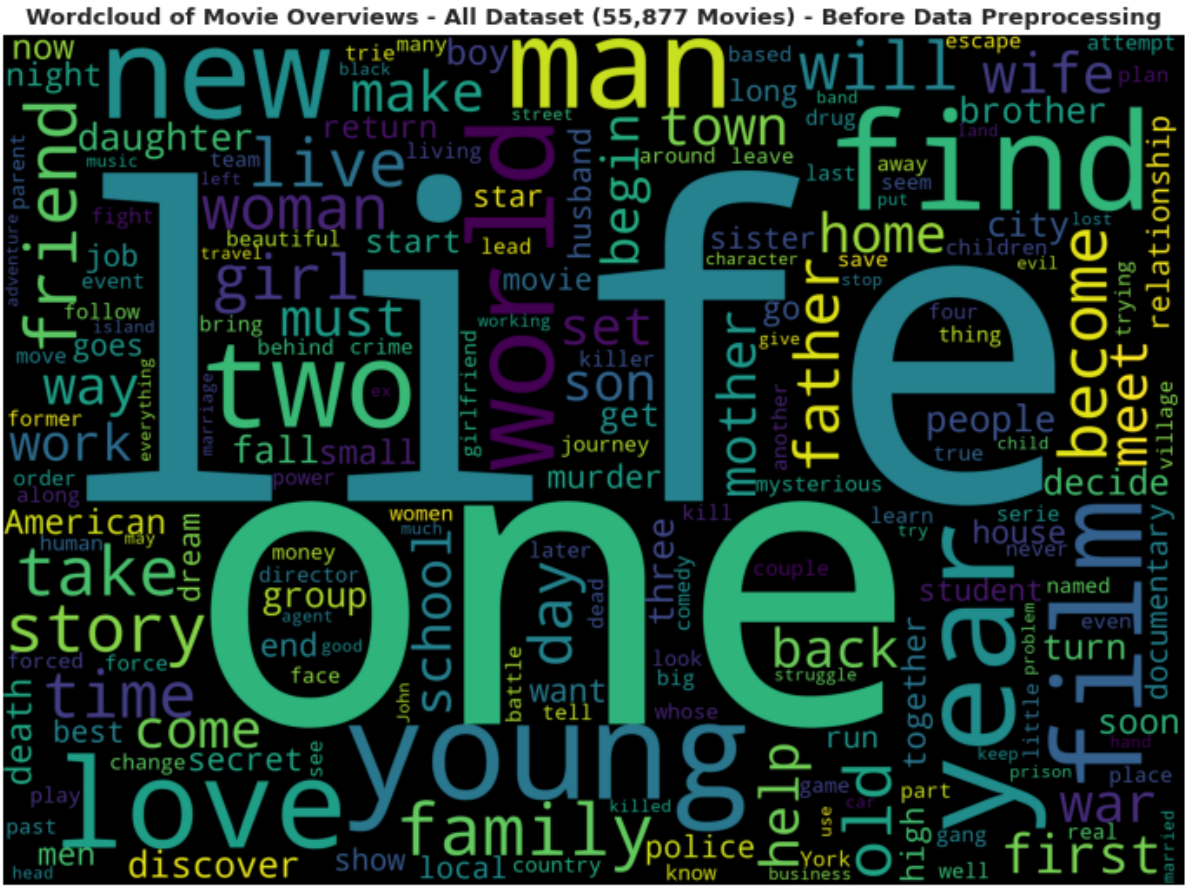

Figure 3. 4: Wordcloud of All the Dataset

However, now that the data has been gathered and before this is fed to a model, the next NLP data preprocessing steps involved:

- Conversion of movie overviews to lower-case
- Removal of punctuation, possible html tags and non-alphabetic characters
- Removal of stop words, customizing the stop words list by adding few extra words, such as “one”, “two”, or “across”, “within”, a fact which gave a better performance
- Fixing some of the informal text in the vocabulary of overviews, e.g. “what’s” was converted to “what is”
- Applying stemming, which is essentially the removal of the suffix from a word and reducing it to its root word (Srivastava, 2019). For example, the stem word of: “amusing”, “amusement”, and “amused” would be “amus”. Applying stemming or lemmatization can help boost the performance of the models, contrary to the occasion of raw text (Riedl & Biemann, 2017:p.7)

After applying the above transformations to the overviews’ text, a feature engineering followed. The data columns at this point can be shown in table 3.1:

	VARIABLES	DESCRIPTION	DTYPE
1	MovieId	The movie id in MovieLens	Numerical (int64)
2	TmdbId	The movie id in TMDb	Numerical (float64)
3	Title	The movie title	Categorical (object)
4	Genres	One or more genres per movie	Categorical (object)
5	Overview	The movie overview	Categorical (object)
6	Vader_score	A dictionary of Vader's scores	Categorical (object)
7	Vader_compound	The Vader's compound score	Numerical (float64)
8	Vader_polarity	Positive/neutral/negative	Categorical (object)
9	NMF_topic	Number in range [0, 6] covering the 7 topics derived from the NMF document	Numerical(int64)
10	NMF_topic_description	Topics' description	Categorical (object)
11	Entities	One or more entities per movie	Categorical (object)

Table 3. 1: Data Columns before Data Preprocessing

Pandas[21] is a software library written for the python programming language for data manipulation and analysis. Using pandas' "astype()"[22] method the variable "Vader_compound", which was displayed as dictionary containing the Vader's negative, positive, neutral and compound scores, was successfully decomposed into 4 new columns, each one expressing the above scores. Accordingly, a feature engineering regarding "entities" occurred so that all entities for every movie are displayed in one and in the respective intersection of the dataframe. Regarding numerical values there is no need for any encoding, however, care should be taken with regard to the categorical types: the variable "overview" which contains narrative text should be treated specially and separately with NLP techniques. Investigating, also, the categorical type of the categorical variables is an essential step before feeding them into a model. For example, treating a nominal categorical variable as an ordinal one, would affect their correct conversion into a numerical data type. LabelEncoder function might lead to that exactly misinterpretation if it is applied to nominal categorical variables since it would generate a rank ordering weight (Pathak, 2020). For this reason, OneHotEncoder[23] was used for the right conversion of the categorical variables into their numerical representation.

3.5 Suitability of Models for the Project's Task

A basic introduction to the different types of classification problems and classifiers is considered essential for seeking out and choosing the correct one for this project. First and foremost, in binary classification, binary classifiers can distinguish between two classes whereas

[21] https://pandas.pydata.org/
[22] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.astype.html
[23] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

in multi-class classification, classifiers are able to distinguish more than two classes (Li, 2018a). However, when the final output is composed of multiple binary tags, then this is called multi-label classification (or multinomial), and an extension of that is the multioutput-multiclass classification where each label itself can be multiclass (Géron, 2019:pp.100–107).

Based on the scikit-multilearn API documentation[24] and respective paper (Szymánski & Kajdanowicz, 2019), Stochastic Gradient Descent (SGD) classifiers, Random Forest classifiers and Naïve Bayes classifiers are able to proceed to multiclass classification. Logistic Regression and Support Vector Machine (SVM) are primarily suitable as binary classifiers. However, there are ways multiclass or multilabel classification can be implemented here with multiple binary classifiers: either with One Versus Rest (OvR), or with One versus One (Ovo) classifiers . In the case of OvR, six binary classifiers should be trained, one for each emotion out of the 6 emotions, combining the six classifiers' outputs as an ensemble method With OvO, a binary classifier should be trained for every pair of emotions, e.g. one classifier distinguishing the emotion "happiness" from "sadness", the other classifier distinguishing "happiness" from "anger" and so on. In every case, Scikit-Learn can implicitly understand when a binary classification algorithm is used for multi-class classification, and automatically it runs OvR or OvO (Géron, 2019:p.101).

In order to gain flexibility and the chance to use a range of classification models, which would not necessarily be restricted in multilabel tasks, the project's classification problem will be treated as a 6-binary multi-class classification task, where the 2 classes will be 1 or 0 for every emotion displayed/not displayed. In other words, the multilabel problem will be decomposed into a 6-set of sub-classification problems, each of which will be a binary classifier, a methodology which is common and efficient across multilabel classification problems (Nag, 2019).

3.6 Multi-label Classification Evaluation Metrics

During the construction of several models and their deployment, mainly three evaluation metrics were used for their comparison, and these were: micro average f1 score, mean subset accuracy score and mean cross validation score. At the end and when the final model was found, then this was further evaluated with the hamming loss metric and with an AUC score in order to gain a complete spectrum of evaluation.

Compared to traditional evaluation metrics for classification problems, here different approach should be followed (Jain, 2017). This happens because a traditional accuracy evaluation score would give a score of 0 for a classifier predicted correctly some of the labels and not all of

[24] http://scikit.ml/api/skmultilearn.html#classifiers-and-tools

them. In other words, in multilabel classification, a single-isolated misclassification should not be considered as a wrong prediction (Nooney, 2018): for example if a movie has 4 emotions, then a classifier predicting correctly the three would be better than a classifier predicting the two out of the four emotions.

3.6.1 Accuracy Metrics

Subset Accuracy (Exact Match Ratio)

Subset Accuracy, as mentioned earlier, is essentially the most strict metric, giving the percentage of samples that have all their labels classified correctly (Nooney, 2018), its equation is in equation 3.1. For this reason, instead of calculating the subset accuracy across all labels (six labels-six emotions), the researcher here decides to calculate the subset accuracy for every label separately (since the multilabel task is treated as a set of 6-binary multiclass labels), and then calculates the mean value of those. This was decided in order not to report the subset accuracy score for every one of the six emotions separately and for all the models built in the next sections.

$$Exact\ Match\ Ratio, MR = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i)$$

Equation 3. 1: Exact Match Ratio - Subset Accuracy

Cross Validation Accuracy Score:

Same rationale as in subset accuracy, but here calculating the accuracy over a k-fold cross validation[25], which presupposes the existence of a validation set. Across the models' deployment, the "CV" parameter of the Scikit-Learn's cross_val_score() method which determines the cross-validation splitting was set to 3 (3 folds). A greater number would have been given if the size of training dataset was larger, but this would not make a lot of sense in the particular task where the labelled dataframe is relatively small. Again, the mean value of cross validation scores over the 6 emotions was calculated and reported.

[25] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

F1-Score

F1 can be interpreted as the weighted average of the precision and recall, representing essentially their harmonic mean (Venkatesan & Joo Er, 2014), and the possible result values range in (0, 1), where F1 score reaches its best value at 1 (Scikit-Learn, 2020b). Micro f1-score (equation 3.2) was decided to be calculated since in multilabel classification tasks it emphasizes more on the common labels in the dataset by providing to each sample the same importance (Peltarion, 2020), as opposed to "macro" f1, which based on the documentation[26], it does not take label imbalance into account.

$$Micro\ F1-score = 2 * \frac{Micro-precision * Micro-recall}{Micro-precision + Micro-recall}$$

Equation 3. 2: Micro F1-score

Hamming Loss

Instead of counting the number of correctly classified data instances, Hamming Loss calculates the loss derived from the bit string of class labels during predictions. It executes XOR operation between the original binary string of class labels and the predicted class labels for a data instance (Nag, 2019). More simply, it is the fraction of labels that are incorrectly predicted, i.e. the fraction of the wrong labels to the total number of labels (Nooney, 2018). Finally, the hamming loss gives the percentage of those wrong labels (to the total number of labels), where contrary to the other accuracy metrics, a perfect classifier has hamming loss with a value of 0 (Venkatesan & Joo Er, 2014). In equation 3.3, "|N|" is the number of data instances, |L| the cardinality of class space, $y_{i,j}$ the actual bit of class label j in data instance I, and ^yi,j the predicted bit of class label j in data instance i. Lastly, hamming loss can be calculated with Scikit-Learn's hamming_loss function[27].

$$\frac{1}{|N|*|L|}\sum_{i=1}^{|N|}*\sum_{j=1}^{|L|}x\ oR\left(y_{ij},\hat{yij}\right)$$

Equation 3. 3: Hamming Loss

[26] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
[27] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html

The first three above metrics were executed and calculated initially using a validation set. This was made mainly for two reasons: test data should only be used when a promising model has been found, and secondly, by using a validation set it can safely be concluded how much a model can be generalized and achieve good results, taking into consideration the potential levels of overfitting or underfitting the training set. After having found the final model, then those metrics were applied again, along in addition with calculating the Hamming loss metric and the ROC-AUC score.

3.7 Natural Language Processing (NLP)

3.7.1 NLP Introduction

NLP, also known as computational linguistics (Hirschberg & Manning, 2015:p.261), is an interdisciplinary field of computer and information science, artificial intelligence and linguistics, which explores the natural language in texts or speeches (Liu, Li & Thomas, 2017:p.1112), and it is used to acquire insights from significant amount of textual data (Quang 2017). Apart from entity extraction and sentiment analysis, NLP field can include: automatic text summarization, machine translation, information search, question answering, machine reading (Hirschberg & Manning, 2015:p.263), text generation, speech recognition and semantic annotation (Liu, Li & Thomas, 2017:p.1115), as well as spoken dialogue systems and conversational agents (Hirschberg & Manning, 2015:p.262).

Sentiment analysis is a field of NLP which focuses on the analysis of people's attitude towards entities (Tkalčič *et al.*, 2016:p.119). These entities might be people, such as individuals, or a group of people (organizations, social media, etc.), products/services, events and topics. Some of the main and famous tasks in this sector are the identification of polarity, which is the task of determining a statement as either positive, negative, or neutral (Géron, 2019:p.534; Maynard, Bontcheva & Augenstein, 2017:p.76), the detection of sentiment's subjectivity (Topal, Koyutürk & Özsoyoğlu, 2017:p.3), as well as opinion identification based on sentiment classification, and the identification of belief states on the basis of lexical and syntactic information (Hirschberg & Manning, 2015:p.265).

In general, NLP can give significant insights in the processing and exploration of text data in an expanding area of fields, from cognitive psychology to artificial intelligence (Crowston, Allen & Heckman, 2012:p.525), filling the gap between human communication and machine understanding (SAS Institute Inc., 2019:p.7). After unravelling and processing the unstructured text via NLP, Natural Language Understanding (NLU) is used for a deeper understanding of the

data making it possible for Natural Language Generation (NLG) to produce human language for specific domains and tasks (SAS Institute Inc., 2019:p.7).

Lastly, a challenge in this field seems to be regarding emotion generation, i.e. the identification of emotions such as depression or cheerfulness (Hirschberg & Manning, 2015), a fact to which this project's work could possibly contribute.

3.7.2 NLP Implementation

For a wide variety of NLP techniques, the researcher used extensively the spaCy[28] environment. SpaCy is NLP library. It can apply many features, such as tokenization, NER, POS Tags, topic modelling and it also includes word vectors while it can easily be integrated with deep learning algorithms. The researcher experimented with various libraries, such that of "Transformers"[29], but the reason he decided to proceed with spaCy library is because its second version released in 2017 offered the fastest syntactic parser worldwide with its accuracy being less than 1% of the best available compared to several systems evaluated in 2015 (Choi, Tetreault & Stent, 2015).

After integrating the spaCy environment, the researcher explores a series of regular expressions, proceeds to tokenization and to a fine-grained part-of-speech (POS) tags. Same words can have different order in sequences of sentences and different meaning. SpaCy creates a "Doc" object which can be used after tokenization, and using linguistic features it can parse and tag a given Doc. Figure 3.5 shows a visualization of POS application in the very first sentence of the first movie overview in the dataset.

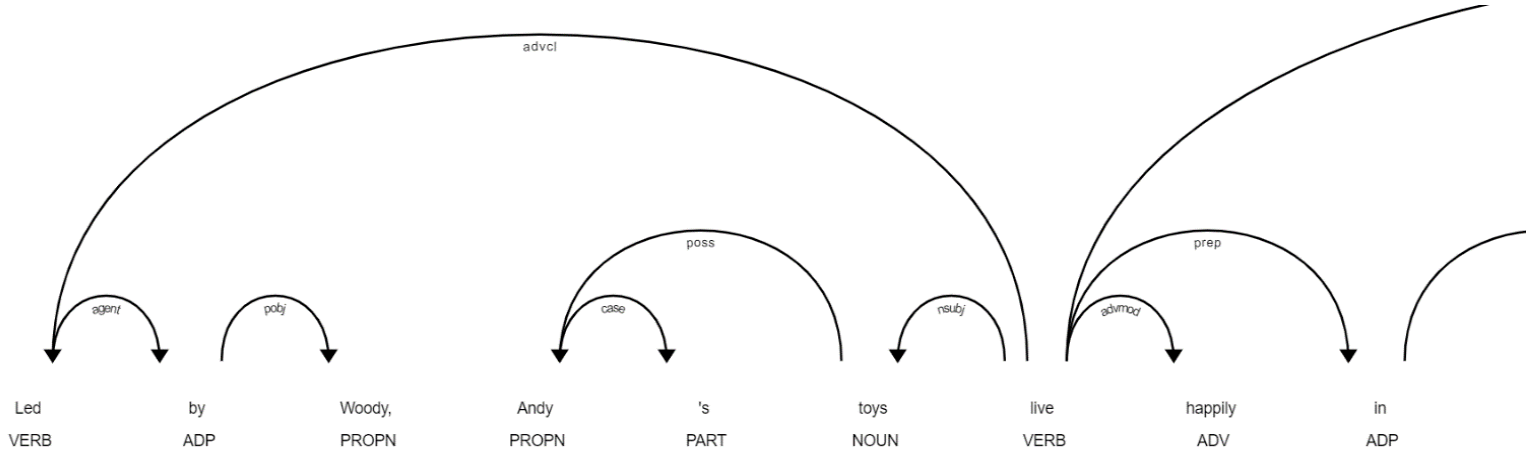

Figure 3. 5: Part of Speech Tags

[28] https://spacy.io/
[29] https://github.com/huggingface/transformers

3.7.3 Sentiment Analysis

Later, the researcher proceeds to sentiment analysis with regard to movie overviews. Although sentiment analysis and polarity identification do not lead to recognition of emotions, the researcher, trying to find useful additional features in the context of feature engineering which could potentially boost any model for the emotions' predictions, he decides to find out the polarity of the overviews. Natural Language Toolkit[30] (NLKT) provides the NLTK's VADER's module[31] (Hutto & Gilbert, 2014) which stands for Valence Aware Dictionary for sEntiment Reasoning. Vader's SentimentIntensityAnalyzer() method can return a dictionary of scores with regard to four categories, i.e. negative polarity [0 , 1], positive [0 , 1], neutral [0, 1] and a compound score [-1 , 1] which is computed by normalizing the above scores, and it means a positive polarity if the score is greater than 0, neutral if equal to 0, otherwise negative. At this stage, the researcher introduces 2 new column features in the dataframe, and these are comprised of the above compound scores (variable "Vader_score") plus the polarity identified for each movie overview ("Vader_polarity"). As it can be seen in the histogram in figure 3.6 and in bar chart of figure 3.7, the majority 49.89% of movie overviews' polarity is negative (negative compound scores) with almost around 28,000 movie overviews, followed by 23,815 positive polarity overviews (42.62%), while a smaller percentage of 7.5% of the total movies was identified with a neutral polarity.

[30] https://www.nltk.org/

[31] https://www.nltk.org/_modules/nltk/sentiment/vader.html

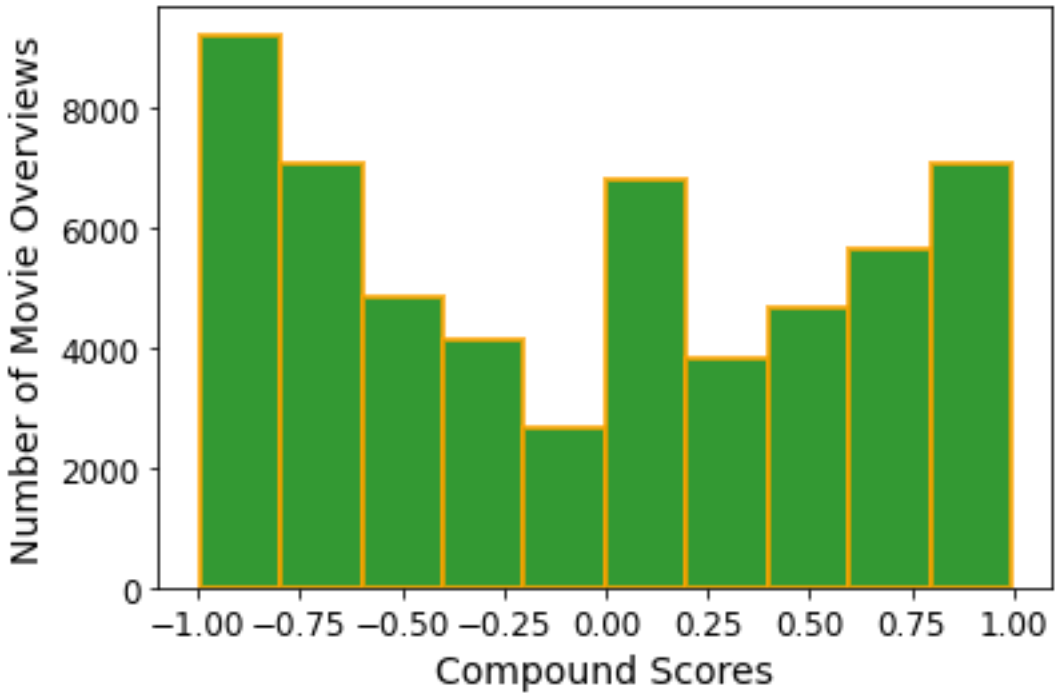

Figure 3. 6: Histogram of Vader's Compound Scores

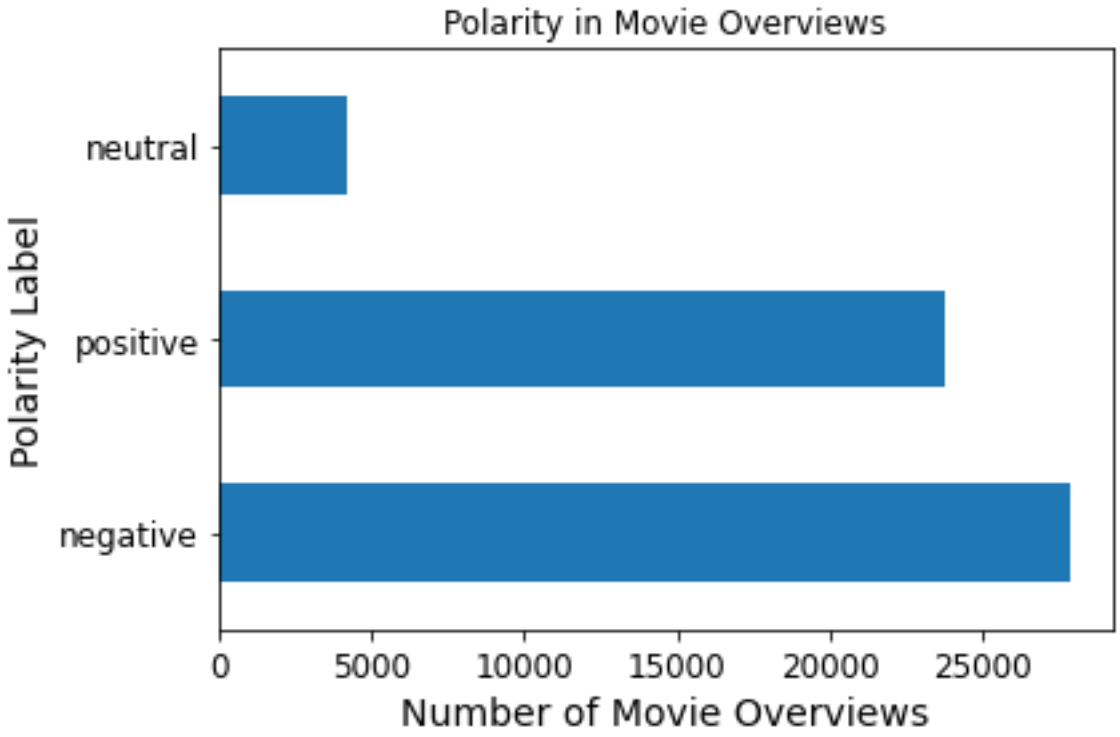

Figure 3. 7: Vader's Polarity in Movie Plots

3.7.4 Topic Modeling

Subsequently, one of the famous tasks in NLP is topic modelling. It is about the discovery of topics given a collection of documents (Jagota, 2020). More broadly, it is an unsupervised

learning technique that simultaneously performs dimensionality reduction and clustering (Portilla, 2019; Chawla, 2017). There are mainly two popular ways to perform topic modelling, either with the Latent Dirichlet Allocation (LDA) or via the Non-negative Matrix Factorization (NMF) (Dwivedi, 2018; Jagota, 2020; Salgado, 2016; Chawla, 2017; Portilla, 2019). The research decides to choose the NMF for two reasons: it is can be deemed as more scalable (Salgado, 2016) and because it can learn more incoherent topics than LDA (Stevens *et al.*, 2012:p.960), and lastly it can be used in conjunction with TF-IDF (Portilla, 2019), which will be referred later on. Creating a document term matrix via NMF, a sparse matrix is created with more than 1.4m stored elements, and every movie overview is assigned with about 40,000 terms. It is necessary to give a predefined number of topics ("n_components") that someone wants to discover, and the researcher experimented with a range of topic numbers. Since there is access to the most common words per topic, the researcher decided that the number 7 can be a good number to represent the range of the different topics regarding all dataset's overviews. Finally, after attributing every topic with a description, the topics were assigned, and the pie chart in the next figure depicts the allocation of those across the dataset.

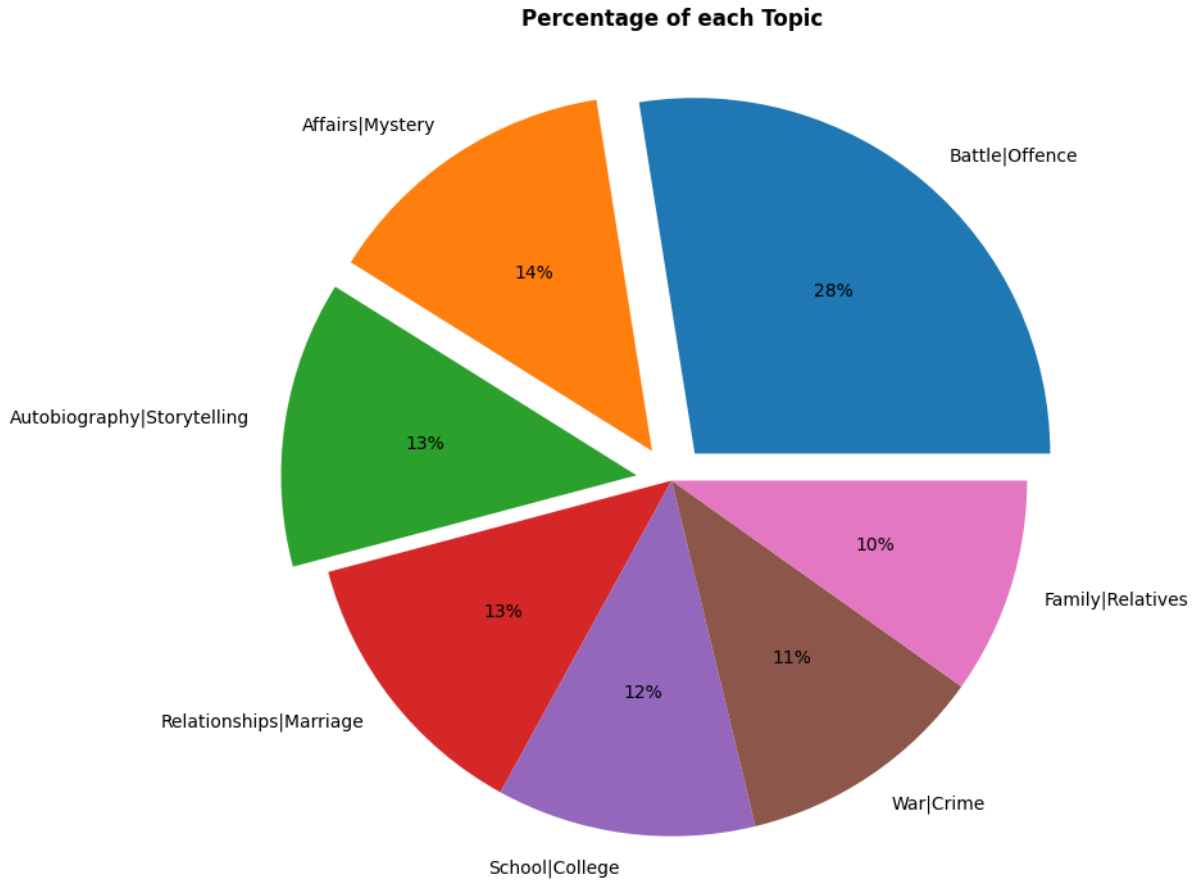

Figure 3. 8: Topic Modeling - Pie Chart

3.7.5 Name Entity Recognition (NER)

Introduced in the beginning of this section, NER can also be referred to as entity chunking, extraction, or identification, is the task of identifying, classifying and extracting key information in text into categories, the so-called entities (Li, 2018b; Marshall, 2019). Again, spaCy can give robust results in this field, outperforming other popular libraries such that of Apache OpenNLP and TensorFlow, when it comes to NER productions. More specifically, based on studies (Shelar *et al.*, 2020), spaCy's process about NERs is combined with an F-score accuracy for every individual tag, while its training loss can be reduced with every training iteration (Shelar *et al.*, 2020:p.9). Exploring the entity '.label_' function, there are totally 18 NER tags[32] (Portilla, 2019) which can be depicted in the following table:

TYPE	DESCRIPTION	EXAMPLE
'PERSON'	People, including fictional	Freddie Mercury
'NORP'	Nationalities / religious or political groups	The Democratic Party
'FAC'	Buildings, airports, bridges, etc.	Eiffel Tower, Charles de Gaulle, the Tower Bridge
'ORG'	Companies, agencies, institutions etc.	Microsoft, MIT
'GPE'	Countries, cities, states	France, Rome
'LOC'	Non-GPE locations, mountain ranges, bodies of water	Europe, Nile River
'PRODUCT'	Objects, vehicles, foods, e.tc. (not services)	Formula 1
'EVENT'	Named hurricanes, battles, wars, sports events	Olympic Games
'WORK_OF_ART'	Titles of books, songs, etc.	The Mona Lisa
'LAW'	Names documents made into laws	Roe v. Wade
'LANGUAGE'	Any named language	English
'DATE'	Absolute or relative dates/periods	20 April 1960
'TIME'	Times smaller than a day	3 hours
'PERCENT'	Percentage, including "%"	Seventy percent
'MONEY'	Monetary values, including unit	Thirty cents
'QUANTITY'	Measurements, in terms of weights or distance	Several miles, 100kg
'CARDINAL'	Numerals that do not fall under another type	2, Fifty-five

Table 3. 2: Name Entity Recognition (NER)

[32] https://spacy.io/api/entityrecognizer

As it can be viewed by figure 3.9, the NER tags of the 1st movie overview, which happens to be the famous animation movie “Toy Story” (1995), is comprised of 9 entities, 7 of which belong to ‘PERSON’ and two labels to ‘ORG’ tags.

Figure 3. 9: Entities in the 1st Movie Overview

The researcher, at this point, after exploring the percentage of each tag in the dataset, decided to proceed to a fine-grained grouping, reducing the number of NER tags from 18 to 13. Since many tags were not popular inside the dataset, and taking into account that some of the entities’ labels have some conceptional similarities, the researcher decides to join the ‘Date’, ‘Cardinal’ and ‘Time’ NER tags creating a custom NER tag ‘entity_time’, also ‘Fac’ and ‘ORG’ are merged together (‘entity_fac_org’), and lastly a new tag ‘entity_measurements’ replaces the ‘Percent’ and ‘Ordinal’ tags. This forms a new allocation of entities in the dataset, which can be viewed in figure 3.10:

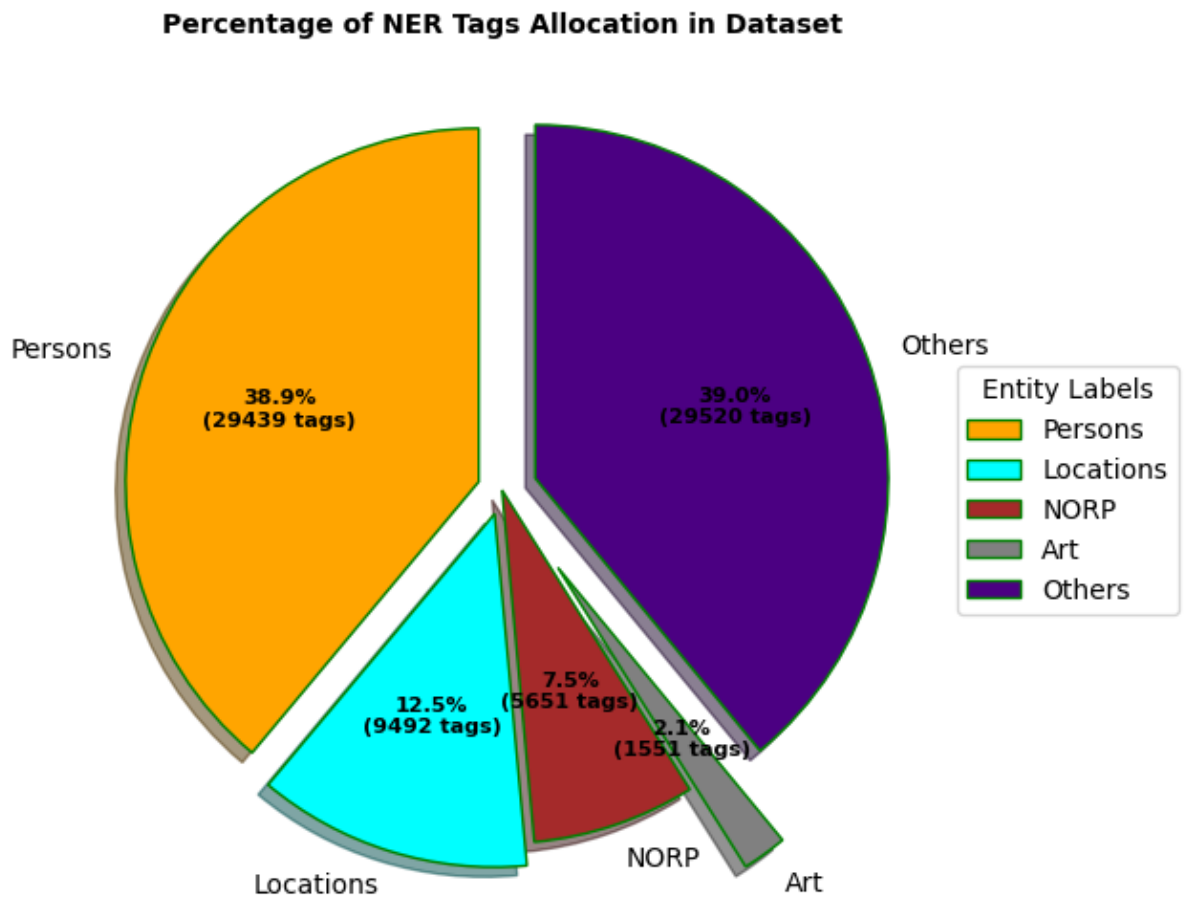

Figure 3. 10: NER Tags Allocation in Dataset

At this phase and after having discovered the entities for every movie overview separately, the researcher decides to add a new feature "entities" into the dataframe. As it can be understood, topic modelling along with NER tags can be successful tools for content classification over large unstructured documents, and these were used as new attributes. All the aforementioned led to the formulation of a new CSV file[33] in which every movie is integrated with its imminent NLP objects.

3.8 Emotion Labelling

Based on the literature review, the researcher decides to proceed with Ekman's 6 emotional classes, and these are:

1) Happiness
2) Sadness
3) Anger
4) Disgust

[33] 'movies_final_2.csv'

5) Fear
6) Surprise

The main goal is here is to build a model which given a corpus of text it can predict and generate the desired emotions. In order for this to be achieved it is necessary the existence of labelled data with regard to those emotions. Based on a wide range of research, no datasets were found with labelled emotions, and especially these undertaken here. Although some relative work can be found (Google Colaboratory, n.d.; Zablocki, 2020a, 2020c, 2020b), these do not match with the imminent emotions used in this project, and therefore it is necessary to proceed to manual labelling of those.

3.8.1 Labelling Process

Labelling will take place based on researcher's best possible judgement, after reading and studying a specific amount of data, in this case movie overviews of the dataset referred in previous sections. Only one person (the researcher) will label the data.

Rules During Labelling

Totally there are 6 emotions. A movie might be tagged with no emotion, or with a combination of emotions with a maximum number of 5 emotions. This happens because the researcher decides that emotion "happiness" cannot concur with emotion "sadness", and vice versa. This does not mean that a movie overview might not contain both emotions, but the rationale behind labelling is that if a movie overview contains both of these emotions, then the dominant one will be labelled. All the rest of the combinations of emotions can be displayed and be accepted. For example, a movie might be labelled with happiness and fear at the same time, but not with happiness, anger, and sadness.

Labelling will have a binary form (scores of 0 or 1), and thus, no gradation/hierarchy of emotions takes place. For this reason, estimating the maximum number of occurrences of emotions that a movie can have, we should talk about combination with no order and no repetition. It is does not belong to permutation because order does not matter, i.e. emotion "fear" might precede the emotion "disgust" -and vice versa, and there is no order because the same emotion should not be displayed twice for a particular movie. The formula (Edgell, 2016; Academy, 2020) of combinations for finding the right number of all possible emotions' occurrences can be given by the equation 3.4. "C" is number of possible combinations, and this

can be found by dividing the factorial number of "n" (number of objects to choose from) with the product coming from the factorial "r" (number of chosen objects) and the factorial of n-r.

$$C_{(n-r)} = n_{C_r = nCr} = \left(\frac{n}{r}\right) = \frac{n!}{r!\,(n-r)!}$$

Equation 3. 4: Calculation of Possible Combinations

In this task, n=5 every time (maximum number of emotions displayed), but "r" alters and depends on the final number of emotions a movie could have (a movie could have from 0 to 5 possible emotions). Executing this equation 6 times, each time with r values in range [0, 5][34] accordingly, it can be found that the maximum possible number of combinations of emotions[35] is 32.

Number of Labelled Data

There is not a right answer on which is the least amount of labelled data required in order to feed it to a model and learn from that, because this depends on the task and the purpose of it. However, in general there are some techniques, such as transfer learning, unsupervised pretraining and data augmentation which can alleviate this problem. (Géron, 2019; Biering, 2020). The former can transmit knowledge from a relative task using pre-trained models already built in a huge amount of data (it will be discussed later on when it will be applied), however for this project's task this could be used only for gaining insights with regard to movie overviews. Unsupervised (or self-supervised) pretraining can be used to train an unsupervised model[36], and by using its lower layers and by adding the output layer for the specific task, the final model can be fine-tuned using supervised learning along with the labelled data (Géron, 2019:p.349). This approach could be close to this task, however no labelled data available has been found from somewhere. Lastly, data augmentation can artificially multiply the already labelled data by generating new data points based on the existing ones. This is quite applicable in the sector of image classification where, for example, data augmentation techniques could rotate an existing image or change its resolution creating in this way a new image artificially.

As it can be understood, it is necessary to manually label movies with regard to the 6 emotions. The researcher initially begins with 200 labelled movies, however considering that

[34] r=0, when no emotion is displayed for a movie, r=1 when the result is 1 emotion, and so on until r=5 gaining all the five maximum possible emotions out of the six

[35] Considering, as mentioned earlier, that two of the emotions (happiness & sadness), should not be displayed concurrently.

[36] For example, autoencoders or Generative Adversarial Networks (GANs)

there are 32 possible combinations and that not all the amount of labelled data should be used for training (i.e. there should be a test and validation set as well), and since after some initial experiments the results were not quite satisfactory, the researcher eventually ends up labelling 300 movies[37]. In the context of trying to reduce bias towards the data feeding of the model, the researcher decides to label 25 movies from 12 movie genres each. This happened because feeding the model with emotions learned only e.g. from comedy or drama movies would not give a uniform distribution of all movies' subjects.

In movies' datasets, as well in movie databases, a movie is usually displayed with two to four genres. After a data analysis, cleaning and feature engineering with regard to movie genres in the dataset, figure 3.11 depicts the top 30 movie genres, and totally 13 distinct genres stand out:

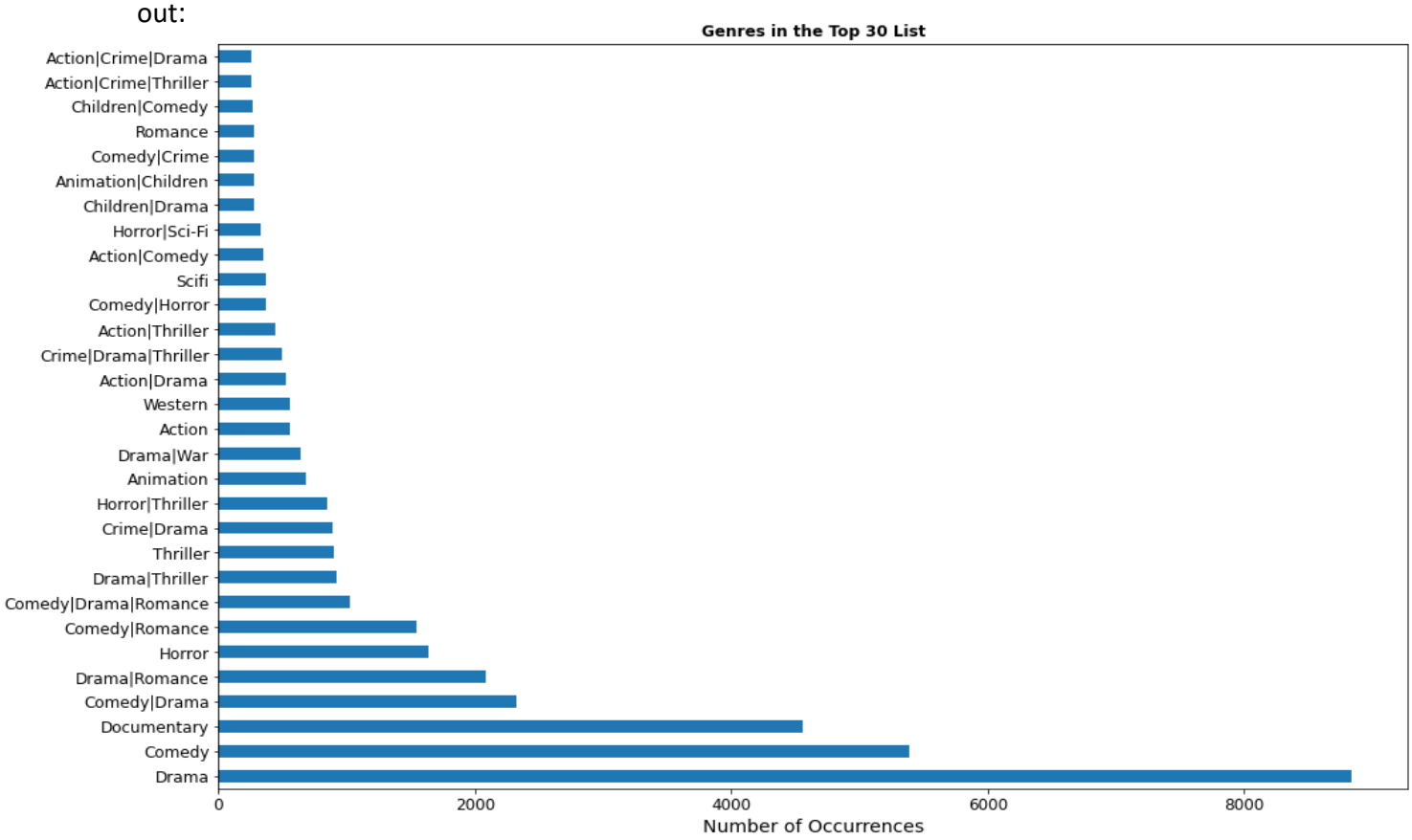

Figure 3. 11: Top 30 Movie Genres in the Dataset

Considering the genres "Animation" and "Children" as one[38], the researcher finally labels 300 movies of the following genres:

[37] Labelling_300_Movies.xlsx

[38] The researcher decides to join these two genres because 1) they most likely reproduce similar emotions, 2) they belong to the bottom list of the most popular genres, 3) In many occasions, in the dataset they

1) Drama
2) Comedy
3) Documentary
4) Romance
5) Horror
6) Thriller
7) Crime
8) War
9) Western
10) Action
11) Sci-Fi
12) Animation & Children

After the successful completion of labelling, figures 3.12 and 3.13 depict the binary emotion assignment as well the percentage of emotional classes allocation throughout the labelling procedure:

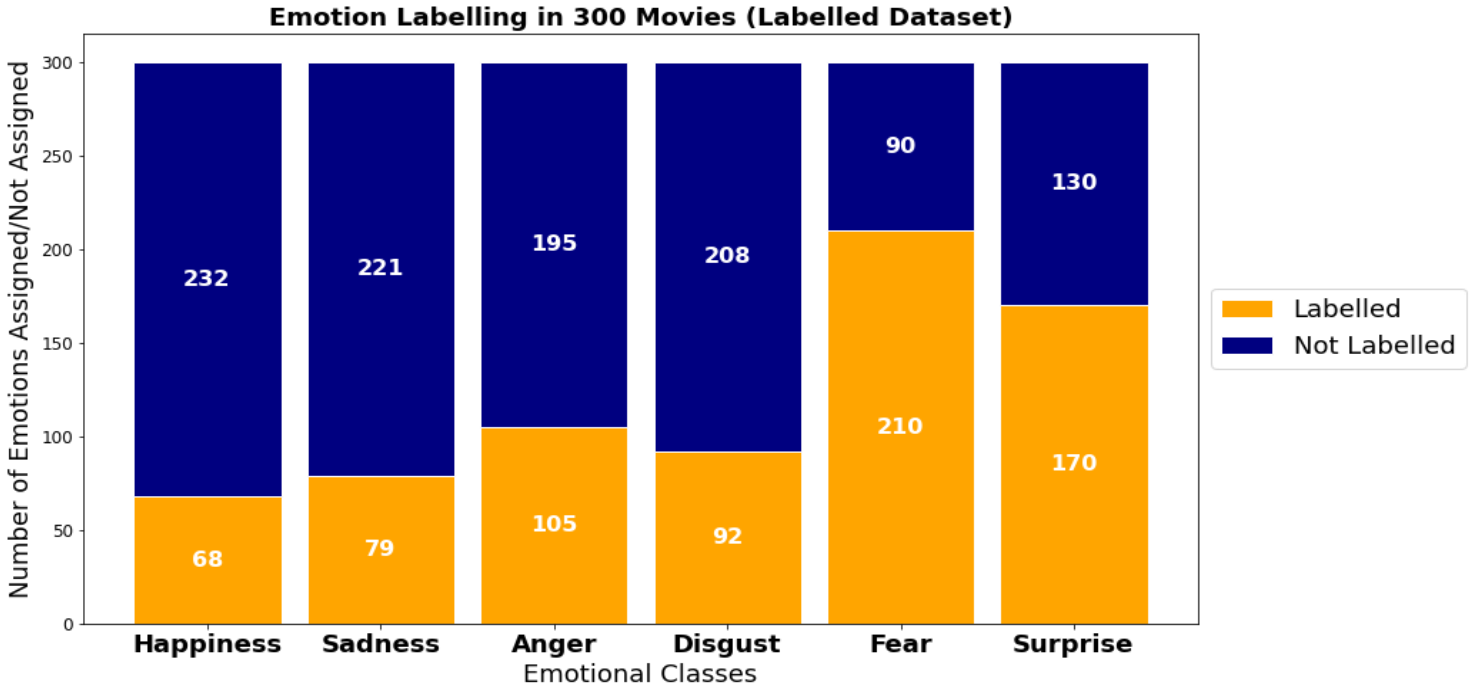

Figure 3. 12: Emotion Labelling - 300 Movies

are displayed together, see rank "25" in figure 3.11, 4) if 25 movies of each of these 2 genres were labelled separately, then this would mean less occurrences of other more popular genres in the dataset.

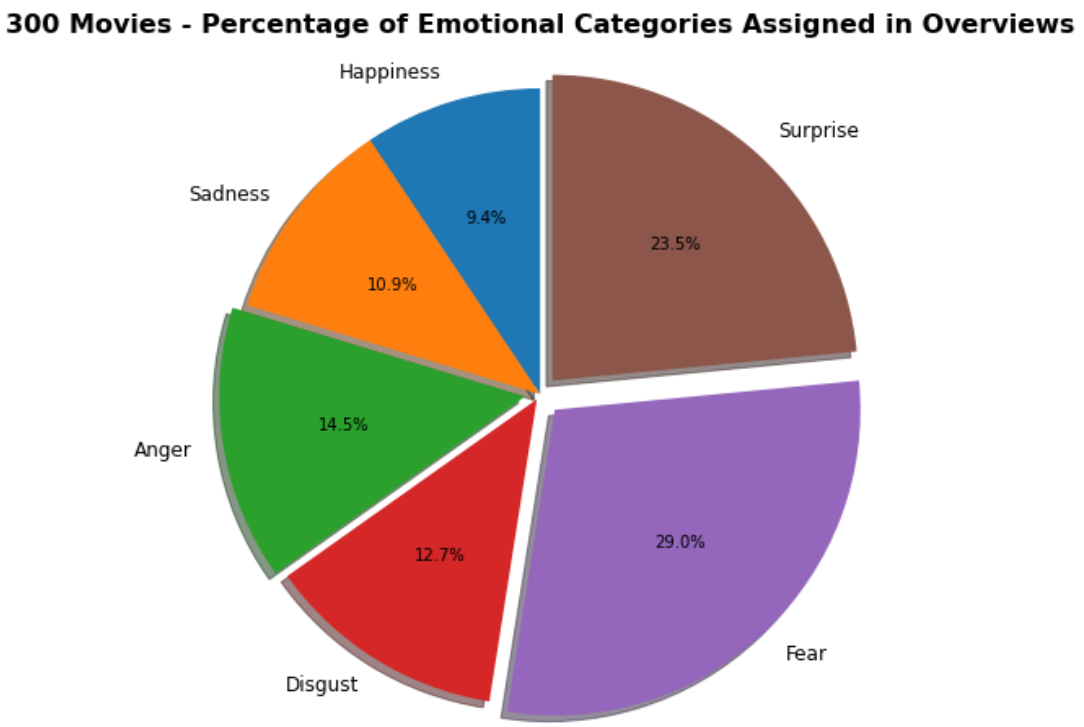

Figure 3. 13: Pie Chart: Allocation of Emotions - Emotion Labelling

Whereas the following figure, figure 3.14, represents the number of combinations of emotions per movie, over the 300 labelled movie overviews. As it can be seen, one of the disadvantages of manual labelling is that e.g. the candidate number of 5 occurrences of emotions over one individual movie took place only once, a fact which might struggle the future model to decide the output of a similar situation, and a fact which justifies why a large amount of labelled data (and the different combination of its occurrences) is considered important for every machine learning model. At this phase, a new dataframe[39] has been constructed, integrating the emotion features which will be the target variables of the model.

[39] movies_final_3.csv

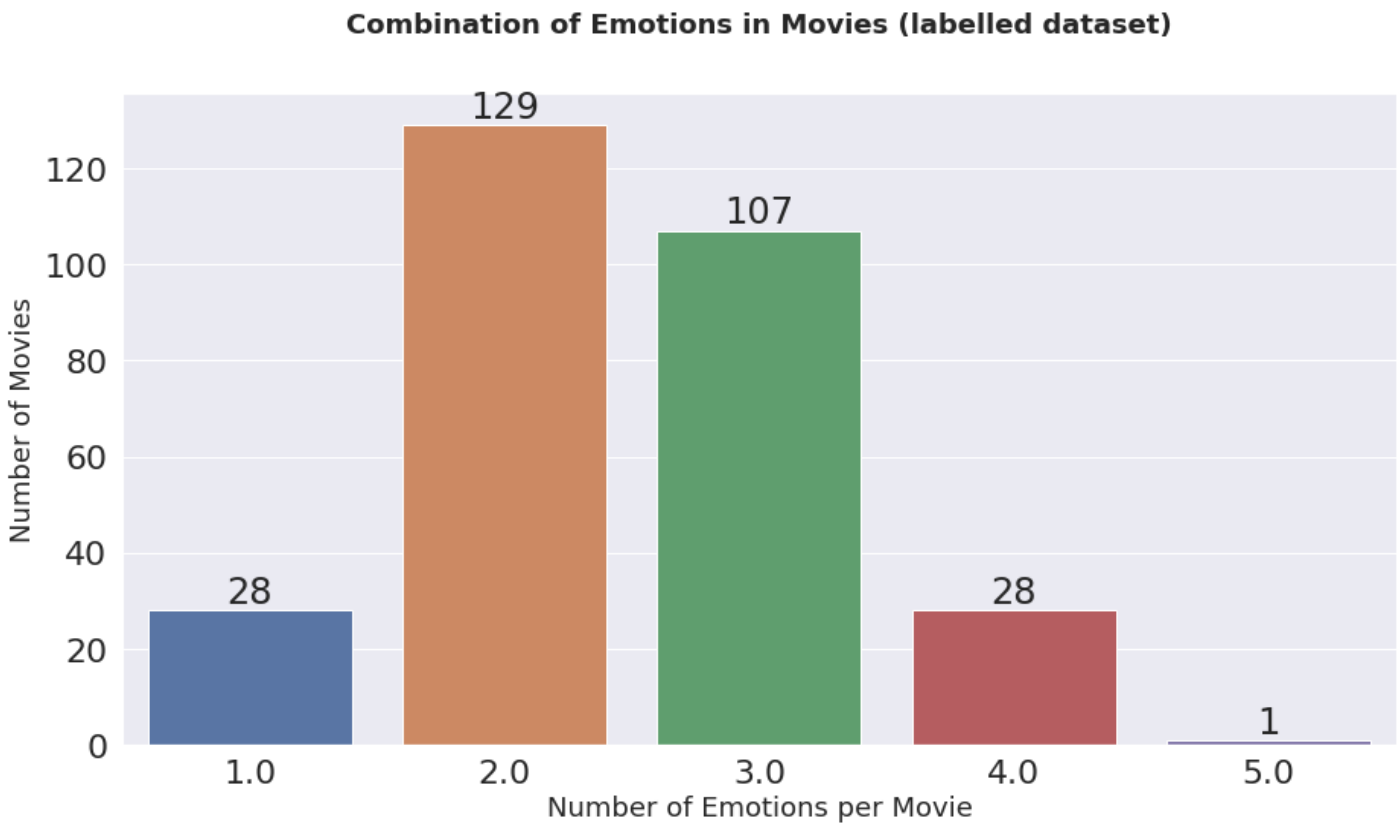

Figure 3. 14: Bar chart: Combination of Emotions after Labeling

Chapter 4|

4. Production of Models, Prediction of Emotions & Findings

After the emotion labelling, the dataset which was consisted of 55,877 rows was split into two parts: the 1st was the labelled dataframe[40] (300 rows since labelling took place in 300 movies), composed of the same variables as before but now carrying the emotions for every movie in a binary form. The 55,577 of the second and bigger dataframe[41] had the same variables across the x axis, but its emotions were null values.

Since it is known that there is No Free Lunch (NFL) theorem and there is no model that can guarantee to perform better (Géron, 2019:p.33), various experiments will take place on several models. In order to better explain which models were built, the researcher will categorize them into machine learning models, and deep learning ones. The former were built in Scikit-Learn environment, while the latter in TensorFlow[42] open-source library with the Keras[43] library, which are mainly used for deep learning. Several experiments were conducted in the context of feature selection in order to achieve the greatest classification accuracy. A good model in terms of predicting the emotions correctly would also be very important and crucial in the next sections when the hypothesis tests are going to be conducted. If for example the unlabeled dataframe is fed with low-accuracy predictions, then the validity of the correlation results would be questioned as well.

The target variables for all models were the six emotions. However, the architecture for the selection of the predictor variables was composed of two kinds. The first type of architecture has as its feature predictors only the movie overviews, while the second type was composed of the movie overviews along with other variables (movie metadata) for a potentially better performance.

4.1 ML Models Using Movie Overviews

This part focuses on machine learning models fed with movie overviews for the prediction of the emotions in the unlabeled dataframe consisted of 55,577 movies. The labelled dataframe, consisted of 300 movies was split into a train, validation and test set in a split of 80%, 10% and

[40] "labelled_df" in the google colaboratory notebook
[41] "unlabelled_df"
[42] https://www.tensorflow.org/
[43] https://keras.io/

10% respectively. After the completion of the data pre-processing, the TfidVectorizer()[44] was applied in the text data, which was later fit and transformed in the training set.

4.1.1 TF-IDF Vectorizer

Term Frequency (TF) – Inverse Document Frequency (IDF) Vectorizer provided by the Scikit-Lean library, can combine the advantages of using both the CountVectorizer()[45] and the TfidfTransformer()[46]. The former converts a collection of text documents to a matrix of token counts, generating a sparse representation of the counts using the spicy.sparse package (Scikit-Learn, 2020a). In addition, it contributes to the preprocessing of text before the production of words' vector representation by tokenizing the collection of text documents, ,constructing a vocabulary of known words, and finally encoding new documents using that vocabulary (Browniee, 2020a). However, this is not enough for building an efficient NLP model, because words appearing often in the overall spectrum of text documents would give a high weight in the encoded vectors. TF-IDF Vectorizer brings a balance to that, by firstly using the CountVectorizer() to compute the word counts, and then by calculating word frequency scores (IDF and TF-IDF scores). In this way, the TfidfVectorizer() highlights the words which seem to be more interesting and meaningful, i.e. words which are frequent in a document but not across a collection of documents (Browniee, 2020a; Maklin, 2019). In this project case, it would mean that it would take into consideration the word frequency with regard to every movie overview separately and not aggregately for all the movie overviews in the dataframe. TF represents the frequency of a given word within a document. In equation 4.1 num_t is the number of times the term occurs in a document set to d, while $total_d$ is the total number of terms in this document d. On the other, IDF "downscales" words which appear across all documents displayed, by computing the logarithm of the N number of documents in the collection divided by the number of document descriptions which contain term t (equation 4.2). Lastly, TF-IDF combines TF and IDF and produces a term weight which represents the importance of a word both in a document d and to the collection of documents (equation 4.3). In our case, "document" would be a specific movie overview, while "collection of documents" would be the corpus of all movie overviews in the dataframe.

[44] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[45] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[46] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

$$tf_{dt} = \frac{num_t}{total_d}$$

Equation 4. 1: Term Frequency

$$i\,df_t = log(N\ /\ n_t)$$

Equation 4. 2: Inverse Document Frequency

$$(TF - IDF)\ weight_{dt} = idf_t * tf_{dt}$$

Equation 4. 3: Term Frequency - Inverse Document Frequency

In order to apply correctly the above transformations to the rest of the data, they were later transformed in the validation and test set. Fitting of the transformations here should be skipped (Géron, 2019; Srinidhi, 2019).

The models which were used in this part are:

1) Logistic Regression as OvR classifier
2) Multinomial Naive Bayes
3) Linear SVClassifier
4) Random Forest
5) SGD Classifier
6) SVC
7) Neighbors Classifier
8) SoftMax Classifier
9) Decision Tree Classifier
10) XGBClassifier
11) Multi-layer (MLP) Perception Classifier (shallow network)
12) Multi-layer (MLP) Perception Classifier (deep network)

Although the MLP classifiers are neural network ones, they are displayed in this section because they can be performed with Scikit-Learn. Figure 4.1 depicts the model's results with regard to micro f1 score, mean cross validation score and the mean subset accuracy score.

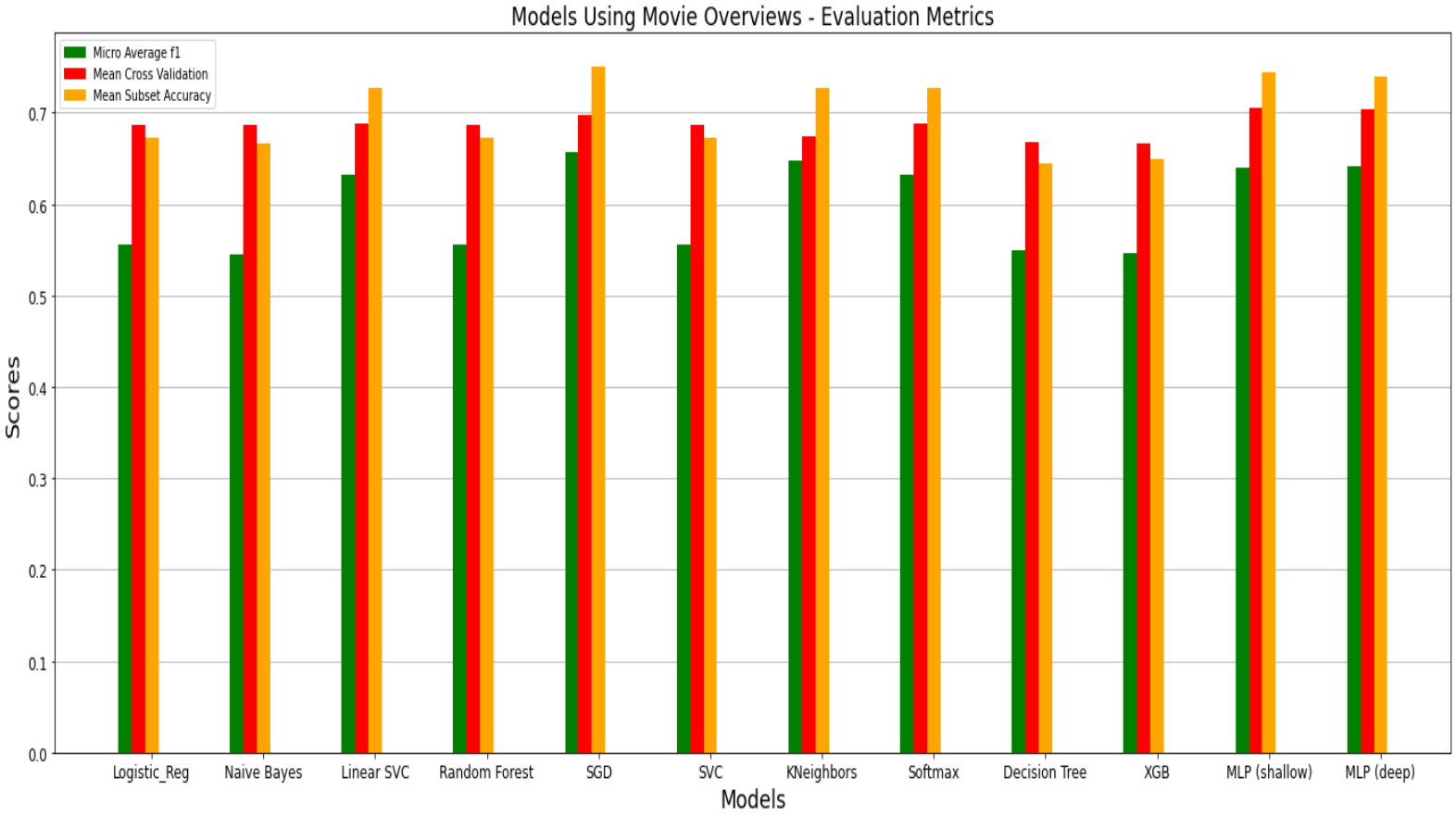

Figure 4. 1: ML Models Using Movie Overviews - Evaluation Metrics

Test data should only be seen when the most promising model has been found, and therefore until then evaluation should take place on a validation set. For this reason all of the above models were evaluated on a 3-fold cross validation set[47]. A voting classifier also was conducted with the SVC, SGD and Kneighbors classifiers which seemed to give promising results, however the overall performance was not greater with great levels of overfitting (micro f1-score in validation set: 65.65%). This happened because if all classifiers are trained on the same data, they are likely to make the same type of errors, hence there will be many majority votes for the wrong class, reducing in this way the ensemble's overall accuracy. Nevertheless, this would be difficult to be implemented here since the test and validation set have already a small amount of data available because of the absence of adequate labelled data.

[47] If labelled data was greater, then the k-fold cross validation could be even greater as well.

Overall, the most promising models of this section were: SGD Classifier (micro f1: 65.65%, mean cross validation score: 69.72%, mean subset accuracy score: 75%), and the linear SVClassifier (micro f1: 63.16%, mean cross validation score: 68.75%, mean subset accuracy 72.78%). Although the MLP Classifier had close results with those of the two promising ones, they are not displayed here because their computational complexity[48] was quite high giving no significantly better results

4.2 ML Models Using Movie Overviews & Metadata

Same train-validation-test split took place here as well, for a valid comparison between models across the different types of architectures constructed, whereas 3-fold cross validation was used again for a complete estimation of models' accuracy. However, here care should be taken for the right conversion of the different variable types belonging to movie meta information, as well to the right integration of both movie overviews and metadata towards the model input. Since different transformations are applied to movie overviews from the rest of the predictors, there is a need for the construction of a machine learning pipeline which would carry all those transformations in a harmonic way.

A feature selection took place for choosing the movies metadata. These were both categorical and numerical variables. The former were consisted of the title of the movie, the movie genres, the Vader's polarity, and the NMF topic description. Whereas Vader's four polarity scores (negative/neutral/positive/compound score) along with the number of the NMF topic constituted the numerical features. StandardScaler() function was used in numerical values while the categorical ones were OneHotEncoded. In parallel, the movie overview variable was vectorized with the TfidVectorizer in the same way as in the previous subsection.

In order the heterogeneous data features can be combined, the Scikit-Learn's ColumnTransformer() estimator[49] was used to apply the corresponding transformations to the dataframe's variables as well as for their integration and transition to the train set (Browniee, 2020b; Honold, 2020). A sparse matrix is created after the ColumnTransformer() and TfidVectorizer transformations, and in order all the predictors to be concatenated as an array without affecting their order, the numpy.hstack[50] was used to stack the arrays in sequence

[48] Their model execution time was triple to quadruplicate.
[49] https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html
[50] https://numpy.org/doc/stable/reference/generated/numpy.hstack.html

horizontally. All above transformations were fit and transformed to train data, and transformed to validation and test set. Figure 4.2 depicts the results:

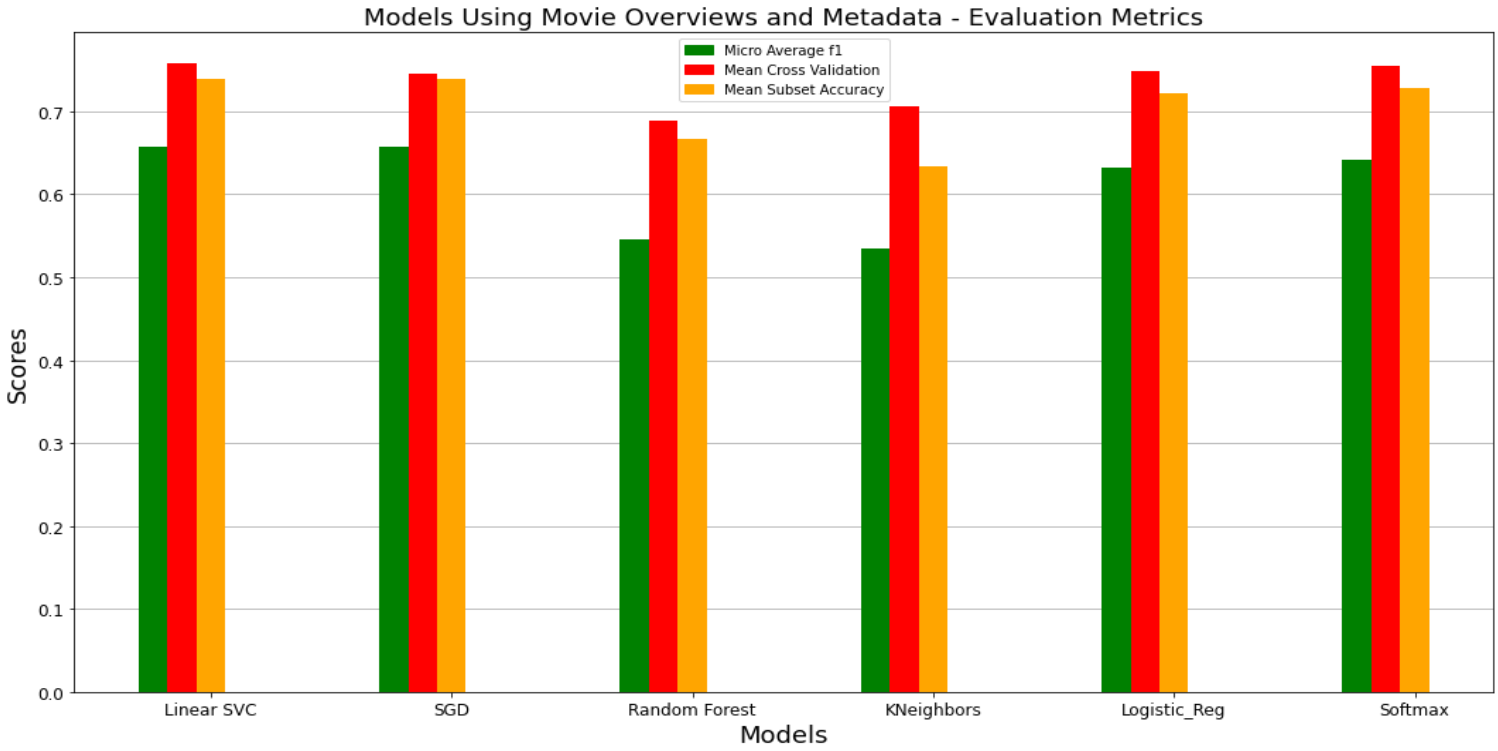

Figure 4. 2: ML Models Using Movie Overviews & Metadata - Evaluation Metrics

The most promising models were again the SGD and the linear Support Vector Classifiers with almost equal results, however these were higher compared to using only the movie overviews from the previous subsection. Linear SVClassifier: (micro average f1 score: 65.69%, mean cross validation: 75.76%, mean subset accuracy: 73.89%), and SGD Classifier: (micro average f1 score: 65.69%, mean cross validation: 74.51%, mean subset accuracy: 73.89%). As it can be understood, both models had the same mean subset accuracy and micro average f1 score, with the linear SVClassifier having 1.25% higher mean cross validation score.

Having found the above most promising models, the researcher proceeds to a fine-tuning of those via RandomizedSearchCV()[51] and GridSearchCV()[52] so that an exhaustive search over a range of specified parameter values and hyper parameters of the estimators can be discovered,

[51] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

[52] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

and consecutively, this could hopefully lead to a performance boost. Different experiments took place for a wide variety of the parameter values for both two models, and Linear SVClassifier seemed to have the best results. At this phase, in the context of trying to upgrade even more the results of the benchmark model, Scikit-Learn's "feature_importances_"[53] were investigated via the RandomForestClassifier() and GridSearchCV() in order to measure the relative importance of each feature in the dataset (Géron, 2019:p.200; Santhanam, 2019; Teng, 2019). This helped the researcher realize that the categorical variables "title" and "entities" did not seem to be important for the model's right predictions, on the contrary this could also possibly make it lead to confusions and to more worse results. Therefore, those variables were removed from the predictors' list.

Running the linear SVClassifier again with GridSearchCV() and with the final predicting features, the Grid Search best estimator gave the best possible parameters of the model which were: "C" regularization parameter: 1, "class_weight': balanced[54], "dual"[55]: True, "multi_class": ovr, "penalty"[56] = l2, "tol"[57]: 0.0001 (1e -4), "fit_intercept": True, "loss": squared_hinge[58]. Applying the above parameters in the linear SVClassifier, micro average f1 score increased by approximately 3.4%, mean cross validation score by 0.35% and the mean subset accuracy by 1.8%: (micro average f1 score: 69.06%, mean cross validation: 76.11%, mean subset accuracy: 75.69%). Since this will be the final model, now predictions can be made on test data, where micro average f1 score was reduced significantly reaching 54.79%, while the mean subset accuracy indicated much lower levels of overfitting the training data with a score standing at 75%.

4.3 Final Model: Evaluation & Predictions in the Unlabeled Dataframe

4.3.1 Final Model

Now that the final model has been selected the researcher can proceed to predictions to the unlabeled data. Validation set can be erased and training data can be increased, since now we are aware of how much this model can be generalize and its validity levels. However, in order to have a rough estimation, not all 300 movies were used as training set, and since there is now no need for a validation part, the final split was 85% - 15% between train and test set.

[53] https://scikit-learn.org/stable/modules/feature_selection.html

[54] The balanced class weight adjusts the target's (emotions) weights inversely proportional to class frequencies in the input data (Scikit-Learn, 2020f).

[55] It solves the dual optimization problem.

[56] The penalization norm.

[57] The tolerance grade for stopping criteria.

[58] The square of the hinge loss.

By training with 85% of the available labelled dataset, i.e. with 5% more training data than the previous times and with 15% of the dataset for the test set, the micro average f1 score was increased by 4.09% reaching 73.15% compared to the corresponding score in validation set of the previous subsection, while the mean subset accuracy reached 78.52% with a rise of 2.83%. The final model's features were composed of the movie overviews, along with the metadata: movie genres (categorical), Vader's polarity (categorical), Vader's compound score (numerical), and NMF topic (numerical).

4.3.2 Final Model's Evaluation

Confusion Matrix & Classification Report

Using the Scikit-Learn's multilabel_confusion_matrix()[59] function a confusion matrix of the predictions in test data was computed (table 4.1), which is a summarized table reflecting the performance of classification models (Terence, 2020). Test data consists of the 15% of the unlabeled dataframe which leads to 45 predictions per emotion (for each of the six emotions) over the movies and movie metadata, therefore n=45 for every one of the six confusion matrices. Let the multilabel confusion matrix be noted as MCM, then the count of true negatives (TN) is $MCM:,_{0,0}$, false negatives (FP) is $MCM:,_{1,0,}$,true positives (TP) is $MCM:,_{1,1}$, and false positives (FP) is $MCM:,_{0,1.}$ (Scikit-Learn, 2020d). The first horizontal row of every emotion's confusion matrix represents the False Positive Rate (FPR): FP/(FP+TN), whereas second row the True Positive Rate (TPR): TP/(TP+FN). Having treated this multilabel problem as a set of multiclass classification problems, now all traditional evaluation metrics can be applied, such as Precision: TP/(TP+FP), Recall: TP/TP+FN, F1 score (already calculated in the previous subsection): 2*Precision*Recall/(Precision+Recall), and Specificity: TN/TN+FP. Overall, the model seemed to confuse instances coming from the emotions "surprise" and "sadness".

For example, in "surprise" emotion, although the model correctly classified 18 movies belonging to that label (TP) and 8 movies which were not (TN), the model wrongly predicted (FP) instances as "surprise" 11 times when they were not, and it gave 8 negative predictions to observations that were truly positive (FN). Accordingly, 3 movies were falsely classified as "sadness", whereas the model rejected the "sadness" emotion in 6 movies when they really had this emotion displayed in the test set.

[59] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.multilabel_confusion_matrix.html

Happiness	TN: 33	FP: 3
	FN: 1	TP: 8
Sadness	TN: 30	FP: 3
	FN: 6	TP: 6
Anger	TN: 27	FP: 3
	FN: 5	TP: 10
Disgust	TN: 31	FP: 2
	FN: 4	TP: 8
Fear	TN: 4	FP: 7
	FN: 5	TP: 29
Surprise	TN: 8	FP: 11
	FN: 8	TP: 18

Table 4. 1: Confusion Matrix of the Final Model

The main classification metrics can also be depicted through a classification report[60]. As was reported in the beginning of this subsection, the micro average f1-score of 73.15% can also be confirmed below. Table 4.2 shows the overall results:

	Precision	**Recall**	**F1-score**	**Support**
Happiness	0.73	0.89	0.80	9
Sadness	0.67	0.50	0.57	12
Anger	0.77	0.67	0.71	15
Disgust	0.80	0.67	0.73	12
Fear	0.81	0.85	0.83	34
Surprise	0.62	0.69	0.65	26
Micro Avg	0.73	0.73	0.73	108
Macro Avg	0.73	0.71	0.72	108
Weighted Avg	0.73	0.73	0.73	108
Samples Avg	0.76	0.74	0.71	108

Table 4. 2: Classification Report - Final Model

[60] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

Hamming Loss

The Hamming loss was calculated to the model conducting the final results, and it considered a valuable evaluation metric suitable for multi-label tasks. In multi-label classification we are interested on the partial true positives, and for example, if a movie truly has the emotions "sadness/fear/surprise" displayed in the test set, then a classifier predicting "sadness/fear" would be more accurate than a classifier predicting only "sadness", where this is not the case for the subset accuracy which would consider those classifiers wrong at the same level. Scikit-Learn's hamming_loss() function returns the average Hamming loss between the elements of "y_true" and "y_pred", i.e. between the 45 movies with 6 emotions for each movie in the test set, and respectively the 45 predictions to those. Conversely to the traditional metrics, since here the loss corresponding to the Hamming distance between "y_true" and "y_pred" is calculated, this means that lower results reflect better classifiers (Scikit-Learn, 2020c). Finally, the Hamming loss of the final model was 0.21.

AUROC

Another useful way to visualize and evaluate a classifier is via the Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC), which can also be described as the Area Under the Receiver Operating Characteristics (AUROC) (Narkhede, 2018). The ROC curve is a plot where the TPR (Sensitivity) stands on the y-axis, whereas the FPR (1 - Specificity) on the x-axis (Toshniwal, 2020; Loukas, 2020).

The diagonal line in the ROC curve represents a random model (random classifier), and in this line the predicted probabilities of the respective classes overlap, in other words in this random model it is true that TPR = FPR for all instances across this line. This line, therefore, represents the default threshold of 0.5 probability that an observation belongs to the right class. As a consequence, a classifier is considered better when (the bigger part of) its line belongs to the up and left side of the random model's line, i.e. when TPR>FPR and where misclassifications are fewer compared to the opposite case (Toshniwal, 2020; Géron, 2019:pp.95–96). For the above reasons, the ROC curve is deemed to identify and demonstrate the trade-off between the false positives and the false negatives observations, with a good classifier achieving a high TPR at a low FPR level.

The Area Under the Curve (AUC) ranges in value from 0 to 1, and it reflects an aggregate measure of a model's performance across all possible classification thresholds (Narkhede, 2018). A perfect classifier has a ROC AUC equal to 1, while the ROC AUC score of the random model-

classifier in the diagonal line is 0.5. As it can be understood, the more higher and left the ROC curve is for a classifier, the bigger the AUC that gets formulates and therefore its AUC score. For this reason in the particular project, the emotion "happiness" which seems to have the best

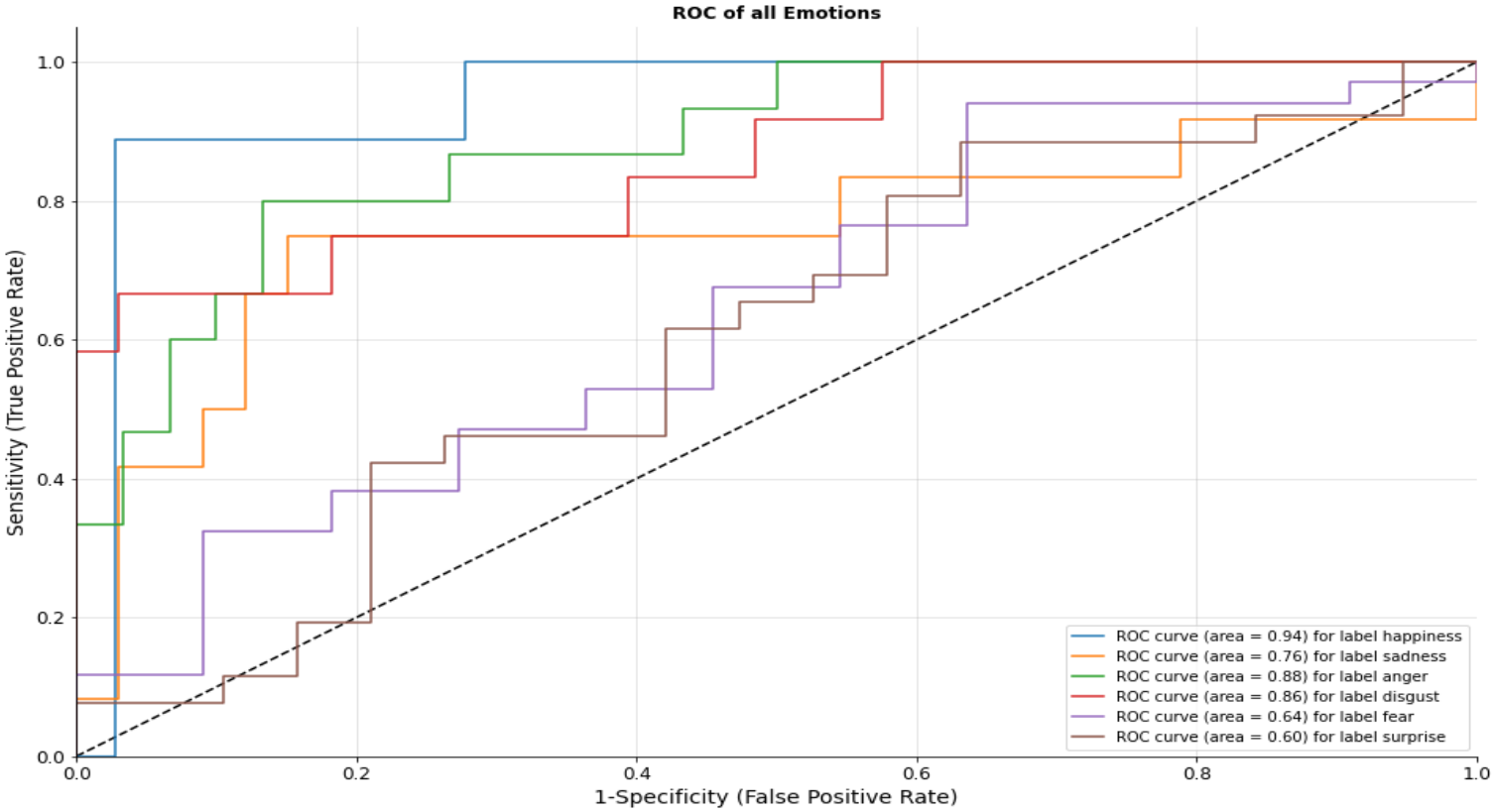

Figure 4. 3: Receiver Operating Characteristic Curve (ROC) - Emotions

formulated ROC curve, has AUC score=0.88 and occupies by far the biggest Area Under the Curve. Whereas the emotion "surprise" comes last with AUC score = 0.63. Overall, each emotion's AUC score was greater than the threshold value of 0.5 respectively, whereas the micro[61] ROC AUC[62] score stood at 0.75. For these reasons, it can be supported that this model now can satisfactorily predict the emotions for the rest of the dataset (i.e. unlabelled dataset consisting of 55,577 movies and movie metadata).

4.3.3 Final Predictions

Finally, now that the benchmark model has been found and evaluated, and being aware of its levels of generalization and validity, the final predictions were made on the unlabeled dataset. Training data now consists of all instances of the labelled dataframe (300 movies and

[61] In Scikit-Learn, the "micro" parameter calculates metrics globally by considering each element of the label indicator matrix as a label, in contrast to the "macro" parameter which based on the documentation (Scikit-Learn, 2020e) and paper (Hand & Till, 2001) it calculates the metrics for each label and finds their unweighted mean, a fact which means that label imbalance is not taken into account.

[62] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

metadata) and the predictions concern the rest 55,577 movies. Two dataframes were created after the above procedure: the first carries the aforementioned predictions in their binary form[63], whereas the second one contains the confidence scores derived from the decision function[64] of the linear Support Vector Classifier model[65].

Figure 4.4 depicts the number of emotions assigned by the model in the unlabeled dataframe, in their binary form. The sum of every bar in the plot for every emotion separately agrees with the total number of movies to which the predictions were made, i.e. 55,577. For example, with regard to emotion "happiness", the model predicted that emotion in 9,876 movies, whereas 45,701 movies had at their emotion label "happiness" the number of "0", meaning not assigned.

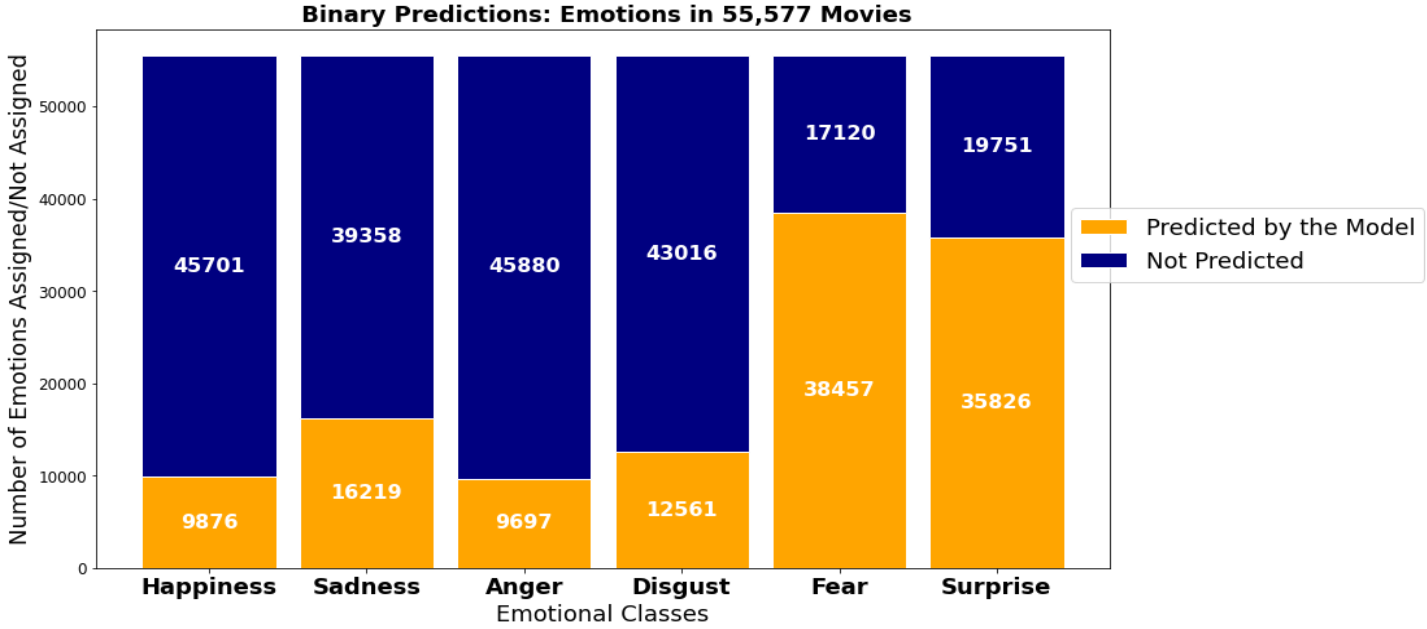

Figure 4. 4: Binary Predictions of Emotions in 55,577 Movies (Unlabeled Dataframe)

The above results can also be viewed aggregately and from another look by the following figure 4.5. As it can be seen, emotions "fear" and "surprise" possess the first and second place respectively, occupying approximately the total of 60% of emotions predicted. 13.2% of predictions concerned the "sadness" emotion (3rd place), followed by "disgust" (10.2%), and "happiness" (8.1%). Lastly, emotion "anger" was displayed the least holding 7.9% of the predicted emotions.

[63] "model_predictions_df"

[64] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.decision_function

[65] predictions_decision_scores_df

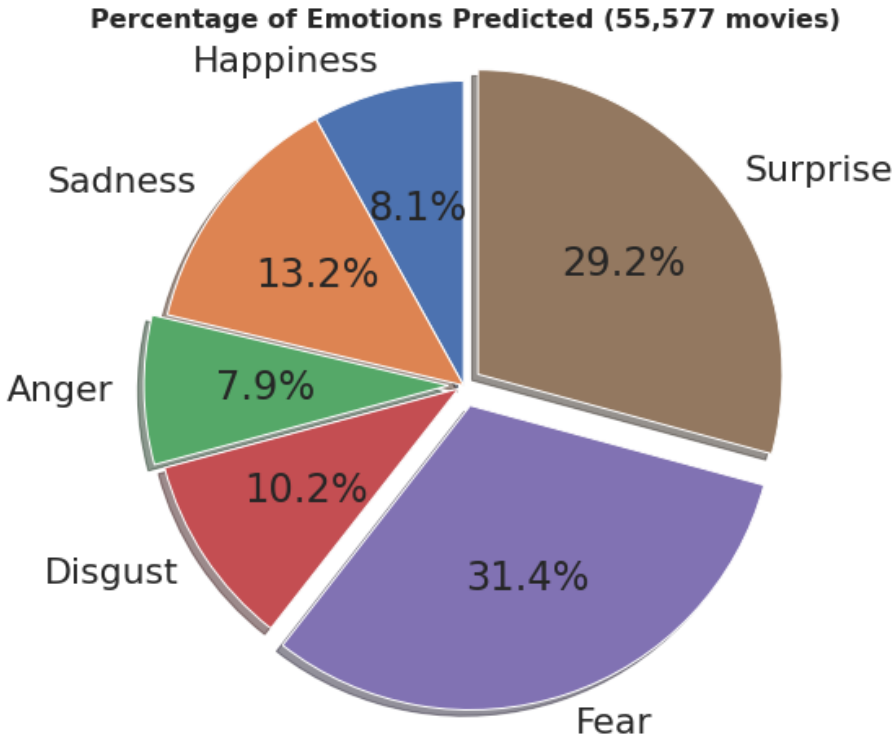

Figure 4. 5: Percentage of Emotions Predicted in 55,577 Movies (Unlabeled Dataframe)

Compared to the emotions labelled by the researcher in 300 movies, the order of emotions has not changed with regard to the 1st place (fear), 2nd place (surprise) as well to the 4th place (disgust). However, emotion "anger" moved from the 3rd to the last place, while the remaining "happiness" and "sadness" emotions rose their position by one and two places respectively. Observing the above comparisons, it can be supported that the fact the model did not give the exactly same order of emotions that it saw during the labelling process can be a good sign: it make sense that different kind of movies (and hence different emotions) were found to the unlabeled dataframe consisting of 55,577 movies, compared to the much smaller size of the labelled dataframe comprised of 300 movies.

An interesting finding would be to investigate the number of predicted emotions per movie. After seeking the above, figure 4.6 shows the results. The sum of all numbers displayed above of every bar is equal with the total number of movies predicted (55,577). As it can be viewed, the majority of movies (23,466) had 2 emotions displayed. The second most common case was movies having 3 emotions (15,663 movies), while 9,369 movies were displayed by only one emotion. The fourth most frequent combination of emotions was found in 4,229 movies which possessed 4 emotions in their emotion labels each, whereas 2,364 movies were emotion free (no emotion assigned). Lastly, the extreme occurrence of movies having 5 emotions was found in 486 movies.

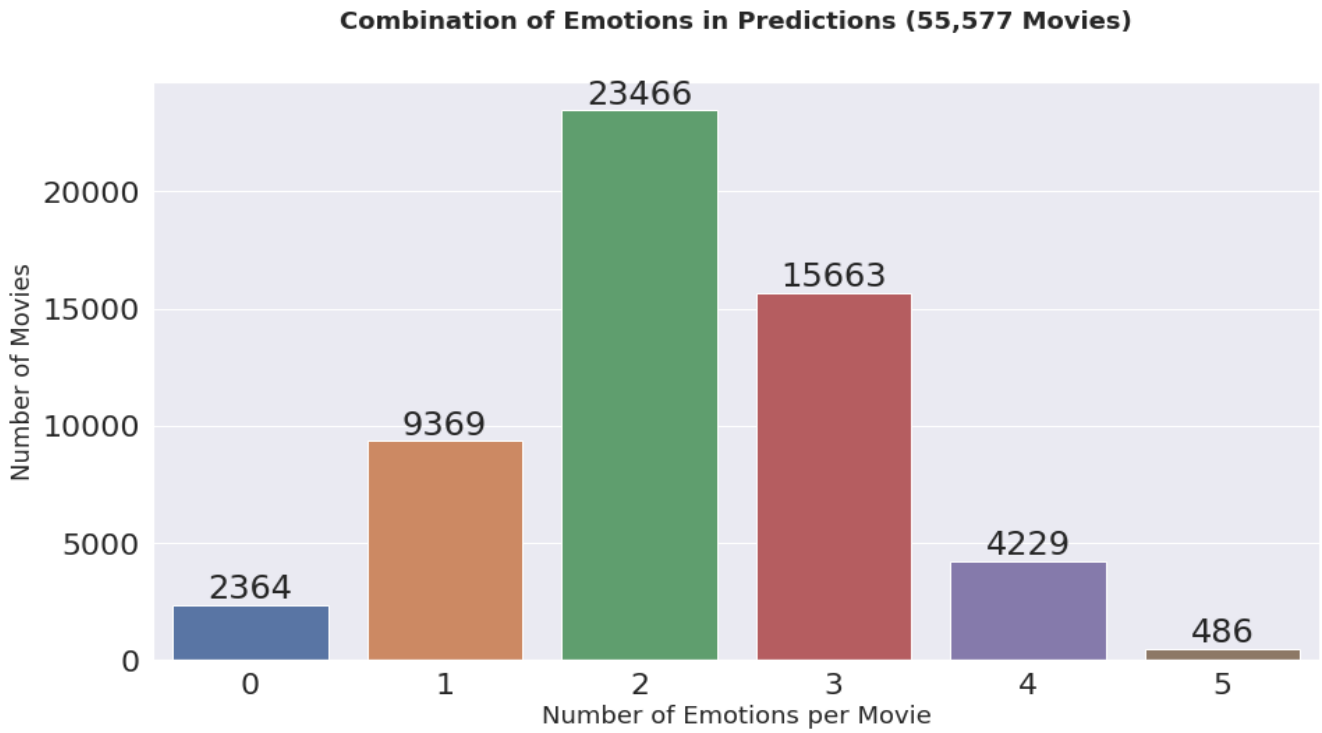

Figure 4. 6: Combination of Emotions in Predictions - 55,577 Movies

Now that the emotions have been displayed for every movie in the unlabeled dataframe, figure 4.7 depicts the word cloud[66] which has been created based on every emotion:

[66] Note that some words might be found in their stemming form since the above word clouds are generated from a preprocessed dataframe.

Figure 4. 7: Wordcloud of Emotions

4.3.4 Emotions' Intensity Magnitude

Closing this subsection, the researcher proceeds to another initiative: as it can be seen from the above figures, different movies have different emotions and a different number/combination of emotions. But what about the intensity levels with regard to every emotion assigned? In the context of finding out how this could be calculated, the researcher uses the decision scores of the final model, which they represent the confidence scores of every emotion assigned or not assigned. Based on the documentation[67] with regard to the functionality decision scores of the linear SVClassifier, all emotions that were displayed as "1" had a confidence score greater than 0, and respectively, all emotions that failed to be assigned (i.e. movies that did not have those emotions) had a confidence score less than 0 (a negative number). However, as it can be seen from the below histogram and density plot the range of values of confidence scores

[67] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.decision_function

across the six emotions was not the same, and this can make sense since we are talking about separate variables and the model was not "confident" at the same level for every one of them.

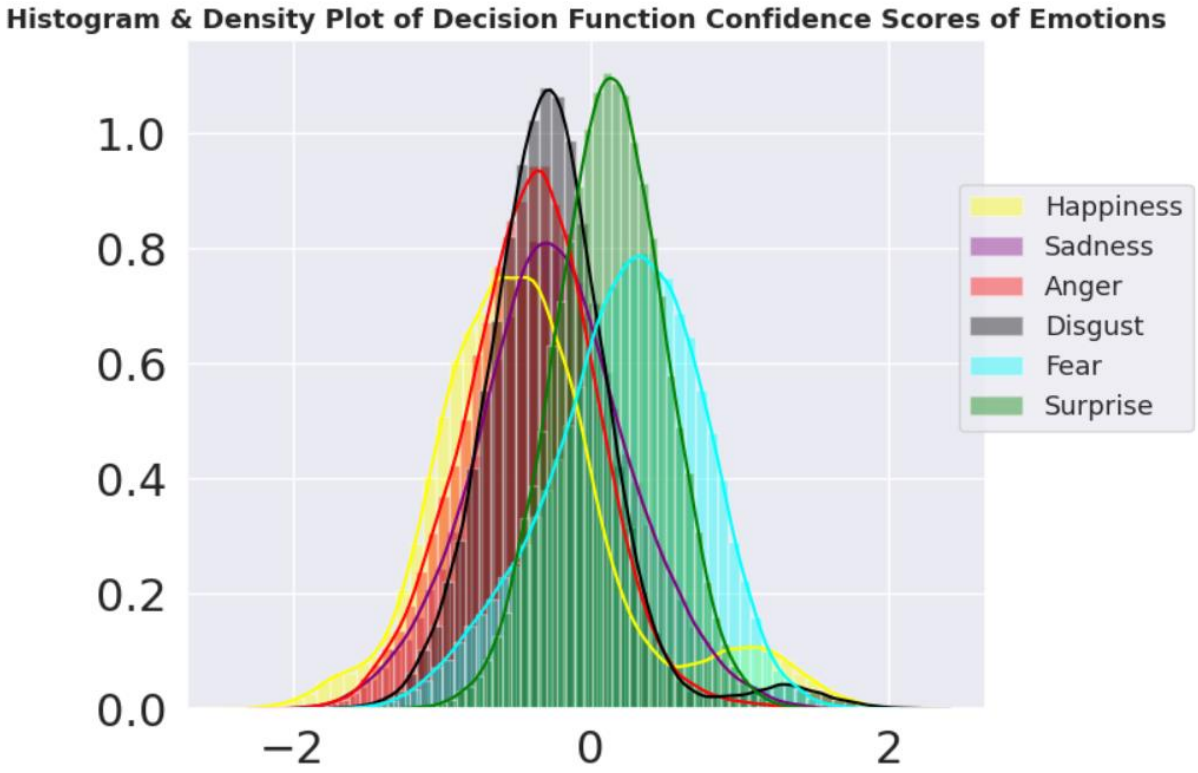

Figure 4. 8: Histogram & Density Plot of the Decision Function Confidence Scores of Emotions

An example could illustrate the above in a more clear way: let us suppose 3 movies, movie with id=1, movie with id=2, and movie with id=3, and let us take only the "happiness" emotion for this example. Let us suppose that the decision function scores of happiness for the above movies were 0.5, 0.3, and -0.2 respectively. This means that: The 3rd movie did not have the "happiness" emotion displayed. Movies with id=1 and id=2 contained "happiness" and both had their binary prediction assigned as "1", however the model was more confident about this for movie with id=1 since its decision function score was higher compared to that of the movie with id=2.

For the above reasons, the researcher proceeds to a normalization of the decision function scores, but only for those scores with a value greater than 0 (so that to pick up only the emotions that were assigned). The normalization chosen was the Scikit-Learn's MinMaxScaler()[68] function which rescales the variables received into the range [0, 1]. After the rescaling process, the researcher decides to attribute three intensity magnitude levels, being: "low" if the rescaled

[68] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

confidence score is in range [0 , 0.25), “moderate” for values in range [0.25 , 0.5], and “high” for values in range [0.5 , 1]. The results of the above actions can be depicted in table 4.3 and figure 4.9:

	Happiness	Sadness	Anger	Disgust	Fear	Surprise
Low	51.91%	70.36%	80.58%	83.89%	50.84%	60.09%
Moderate	24.88%	24.46%	15.72%	7.01%	39.67%	32.87%
High	23.21%	5.18%	3.70%	9.10%	9.49%	7.04%

Table 4. 3: Percentages of the Intensity Magnitude in Predicted Emotions

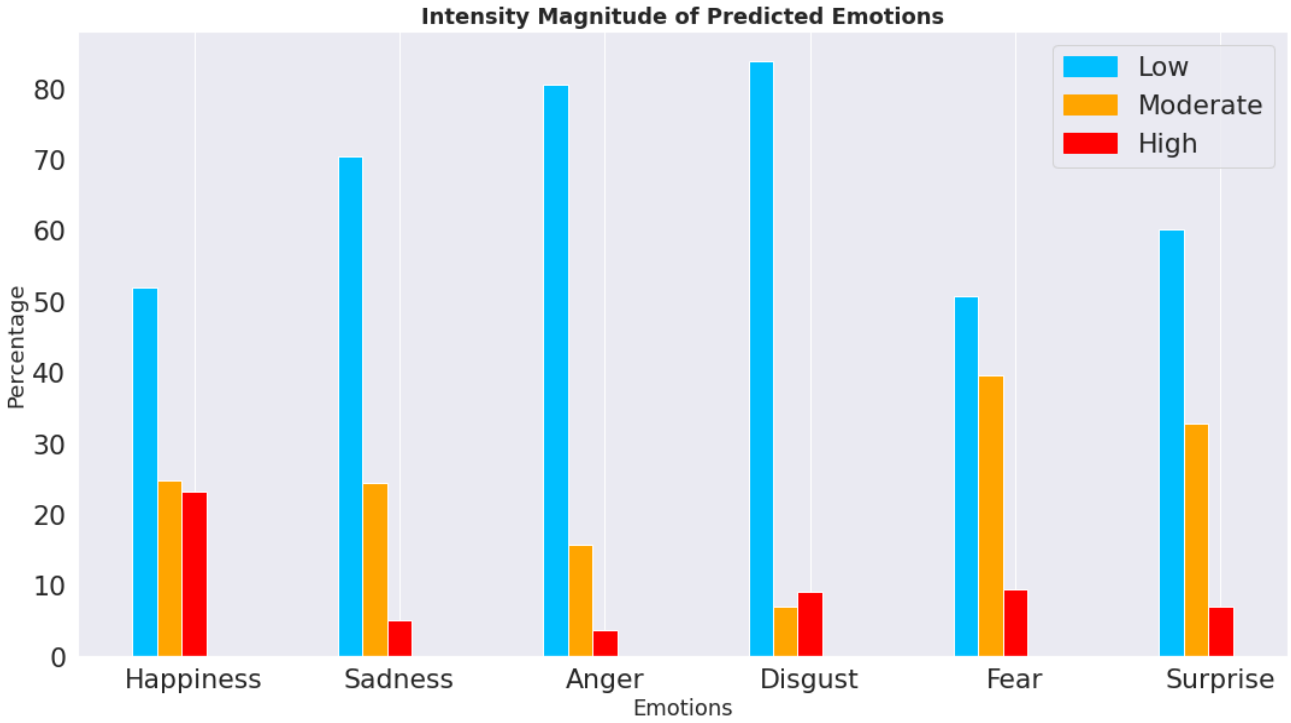

Figure 4. 9: Intensity Magnitude of Predicted Emotions

Some of the basic conclusions inferred from the above are: the majority of each of the emotional categories displayed in the movies had low intensity magnitude, with emotion “disgust” being displayed 83.89% of the occurrences with low intensity magnitude. On the other hand, emotion “happiness” had the greatest percentage of high intensity magnitude compared to the rest of the emotions, with approximately 23% of times being displayed with high intensity levels. Furthermore, another observation that can be made is that “disgust”, although it possesses the greatest percentage with regard to low intensity, it is the only emotion whose second place is the high intensity and not the moderate intensity magnitude.

4.3.5 Examples of Emotions Displayed in Famous Movies

Now that the predictions are made in all movies, table 4.4 illustrates an example of 35 movies displayed along with their emotions. These movies belong to the official top 250 rated

IMDb movies[69]. Where a dash line ("-") exists this means that the emotion is assigned (its binary predicted value was "0"). Intersections at which there is "low", "moderate" or "high" means that these emotions came from a "1" binary prediction result, and their intensity magnitude level was defined as described in the previous subsections.

	Movie (year of production)	**Happiness**	**Sadness**	**Anger**	**Disgust**	**Fear**	**Surprise**
1	Heat (1995)	-	Moderate	-	-	High	Low
2	Taxi Driver (1976)	-	Low	-	High	High	-
3	Pulp Fiction (1994)	-	-	-	-	Low	Moderate
4	The Shawshank Redemption (1994)	-	-	-	-	Moderate	-
5	Forrest Gump (1994)	-	-	-	-	Moderate	High
6	Schindler's List (1993)	-	Low	Low	-	High	-
7	The Silence of the Lambs (1991)	-	-	-	-	High	-
8	The Godfather (1972)	-	-	Moderate	-	High	-
9	Casablanca (1942)	-	-	-	-	-	Moderate
10	The Good, the Bad and the Ugly (1966)	-	-	-	-	Moderate	Low
11	12 Angry Men (1957)	-	Moderate	-	-	High	Moderate
12	GoodFellas (1990)	-	-	-	Low	Moderate	-
13	Amadeus (1984)	-	High	-	-	Moderate	Low
14	The Terminator (1984)	-	-	-	-	Moderate	Low
15	Rocky (1976)	Moderate	-	-	-	Moderate	Moderate
16	Seven Samurai (1954)	-	-	-	Moderate	High	-
17	Life Is Beautiful (1997)	-	Moderate	-	-	Moderate	Moderate
18	American History X (1998)	-	-	Moderate	-	High	-
19	The Matrix (1999)	-		-	-	High	-
20	The Sixth Sense (1999)	-	-	-	-	High	Low
21	Fight Club (1999)	-	-	Moderate	-	Moderate	-
22	Scarface (1983)	-	-	-	Low	High	Low
23	Amélie (2001)	Moderate	-	-	-	-	-
24	The Lord of the Rings: The Fellowship of the Ring (2001)	-	-	Moderate	Moderate	High	-
25	A Beautiful Mind (2001)	-	-	-	-	-	Moderate
26	The Pianist (2002)	-	Low	Moderate	High	High	-
27	The Lord of the Rings: The Return of the King (2003)	-	-	-	-	High	Moderate

[69] https://www.imdb.com/chart/top/

28	V for Vendetta (2006)	-	-	-	-	Moderate	-
29	The Departed (2006)	-	-	-	-	High	-
30	The Dark Knight (2008)	-	-	High	-	High	-
31	Gran Torino (2008)	-	-	Low	-	High	-
32	Inglourious Basterds (2009)	-	Low	Moderate	Moderate	High	-
33	Inception (2010)	-	-	-	-	Low	Moderate
34	Interstellar (2014)	-	-	-	-	Moderate	High
35	Avengers: Endgame (2019)	-	-	Moderate	-	High	-

Table 4. 4: Emotions & Intensity Magnitude - An Example with Popular Movies

4.4 DL Models

In the same way, the dataset is split into two major parts: the labelled dataset, in terms of emotions, consisting of 300 movies and movie metadata, while the rest of the dataset (unlabeled) carries 55,577 movies. The ultimate purpose is to build a model in the unlabeled dataframe so that eventually this could make the predictions for the 6 emotions with regard to the rest of the 55,577 movies.

The main data preprocessing is applied here as well, and a train – validation – test set takes place with the same allocation percentage (80% - 10% - 10%) as in previous model architectures. Proceeding to the building of a deep learning model for this multilabel classification problem, it should be noted that there are mainly two ways this task can be implemented: either with a single dense output layer or with multiple dense output layers (Malik, 2019a). In the first approach, a single dense layer with the number of outputs equal with that of the number of the targets (in this task six, for the six emotions), with a sigmoid (logistic) activation function (Géron, 2019:p.291). In this way, each neuron in the output dense layer would represent one of the six output labels-emotions. The sigmoid activation function returns a value in range (0 , 1) for each neuron, where based on the documentation[70], a number greater than 0.5 would assume that the particular movie is assigned with the emotion represented by that corresponding neuron. In the second approach, each target label (i.e. emotion) is represented by one dense output layer. This leads to a total of six dense layers in the output, where each layer will be activated by a sigmoid function.

4.4.1 Word Embeddings

In the way TF-IDF approach was used via machine learning for the numerical transformation of text data, here word embeddings will be used through which every word can be represented as n-dimensional trainable dense vector (Géron, 2019:p.434; Malik, 2019b;

[70] https://keras.io/api/layers/activations/

Karani, 2018; Cam-Stei, 2019) and can be efficiently applied with deep learning (Malik, 2019b; Analytics, 2017). One of the advantages of using word embeddings over TF-IDF lies in

that the former eliminates the curse of dimensionality of high dimensional data (Yiu, 2019). In other words, via TF-IDF the feature vector for each document can be significantly large, and especially in the case of big datasets this leads to various errors or problems such as computational complexity. It should also be mentioned that word embeddings can efficiently capture both syntactic and semantic relationships in NLP tasks (Tan, 2019).

Although custom word embeddings can be implemented, it is common and more efficient to use pre-trained word embeddings trained by huge datasets. Two popular pre-trained word vectors are Global Vectors for Word Representation[71] (GloVe) (Pennington, Socher & Manning, 2014) and the Word2Vec technique (Mikolov *et al.*, 2013). One main difference between the models using word embeddings via TF-IDF over GloVe and Word2Vec is that the former is based on the frequency method (counting the occurrence of words in corpus) turning it into a sparse count based model, GloVe uses a dense count method, while the Word2Vec generates prediction based word vectors (production of probabilities of word distributions), turning it into a dense prediction-based model (Karani, 2018; Riedl & Biemann, 2017:p.7). Both GloVe and Word2Vec give significant performance, and mostly their difference is found on their building architecture (Gouws, 2017). In this project GloVe was chosen with 100 dimensional word vectors.

Essentially, word embeddings make it possible it for the model to predict words close to any given word (Géron, 2019:p.434), and they fluctuate between 100 and 300 in length[72]. Word embeddings expect the words to be in their corresponding numeric indexes, and this was implemented through the "Tokenizer" class provided by the Keras.preprocessing.text[73] (Malik, 2020). Different values were experimented, and the final maximum length was set to 200. Finally, embedding layers were used, which can be provided by Keras and they can be used to carry, learn, and implement the word embeddings (Malik, 2019b).

[71] https://nlp.stanford.edu/projects/glove/

[72] https://datascience.stackexchange.com/questions/31109/ratio-between-embedded-vector-dimensions-and-vocabulary-size

[73] https://keras.io/api/preprocessing/text/

4.4.2 Architecture of DL Models

- Activation function: The reason the sigmoid is chosen as the activation function lies in the fact that the logistic function can handle and is suitable for multilabel outputs. This is not the case for other activation functions, e.g. in the case of Softmax[74] function the probability of occurrence of one emotion would depend on the occurrence of other emotions. However this would lead to undesirable predictions and results since the final aim is the predictions of each emotion separately and independently from the probability of occurrence of the other emotions.

- Loss function: The purpose of loss functions[75] is to compute the quantity that a model should seek to minimize during training. Loss here will be computed with BinaryCrossentropy because based on the Keras documentation[76] it is suitable and recommended to be used when there are only two label classes, assumed to be 0 and 1, exactly as in this particular task (unlike, e.g. with the categoricalCrossentropy[77] loss function which is suggested to be used when there are two or more label classes).

- Optimizer: Optimizers are necessary arguments for compiling a Keras model. Here Adam[78] optimizer will be used, which is a stochastic gradient descent method, making this an adaptive learning rate algorithm with less need for the "η" learning rate hyperparameter to be tuned (Géron, 2019:p.356).

- Number of Neurons: There is not a pre-fixed right answer about how many hidden layers should be used or how many neurons per layer (Géron, 2019:p.325; Willems, 2019). However, the number of neurons in the output layer should be determined based on the task, in this case it should either be one or six neurons depending on the approach conducted. As with regard to the number of neurons in the input layer, this can also be pre-determined based on the specific task problem, here this number will be equal with the maximum length of the vocabulary that was set during the word embeddings implementation (i.e. 200).

Several experiments took place in this section as well, building DNN models with predictors either with movie overviews plus metadata, or only with the movie plots. Contrary to the ML models, here it was noticed a better performance using only the movie overviews, however and overall, the best possible DNN model built achieved lower accuracy (validation accuracy score: 62.49%, test accuracy: 59.8) than the benchmark ML model of the previous section, and with much higher levels of overfitting the training data. In addition, it was noticed that the models using multiple output layers performed poorly compared to the single output layer models. This can possibly be explained by the fact that having six output layers, each one

[74] https://keras.io/api/layers/activations/#softmax-function
[75] https://keras.io/api/losses/
[76] https://keras.io/api/losses/probabilistic_losses/#binarycrossentropy-class
[77] https://keras.io/api/losses/probabilistic_losses/#categoricalcrossentropy-class
[78] https://keras.io/api/optimizers/adam/

representing one emotion increases the complexity of the model and gives poor performance with high levels of overfitting, especially when this happens to a relatively small trainable dataset.

4.4.3 Part1| Movie Plots with Multiple Output Layers

Below is displayed the figure 4.10, which implements the first approach with six output layers, each one having a dedicated dense layer for each one of the six targets-emotions. Predictors were consisted of the movie plots, and If "y_train" represents the target variables in the train set, then this means here we will have y1_train, y2_train and so on until y6_train for the six emotional classes. The model starts with the textual input (input layer) whose shape is equal with the maximum sentence length which is set to 200. After that, an embedding layers comes forward carrying the vocabulary size, the weights of the embedding matrix created from previous steps, while the number 100 represents the number of dimensions of each word (there are various word dimensions possible to be downloaded[79] and implemented, here the 100 was used). This is followed by a Long Short-Term Memory (LSTM) layer which, in general, can alleviate the limited short-term memory

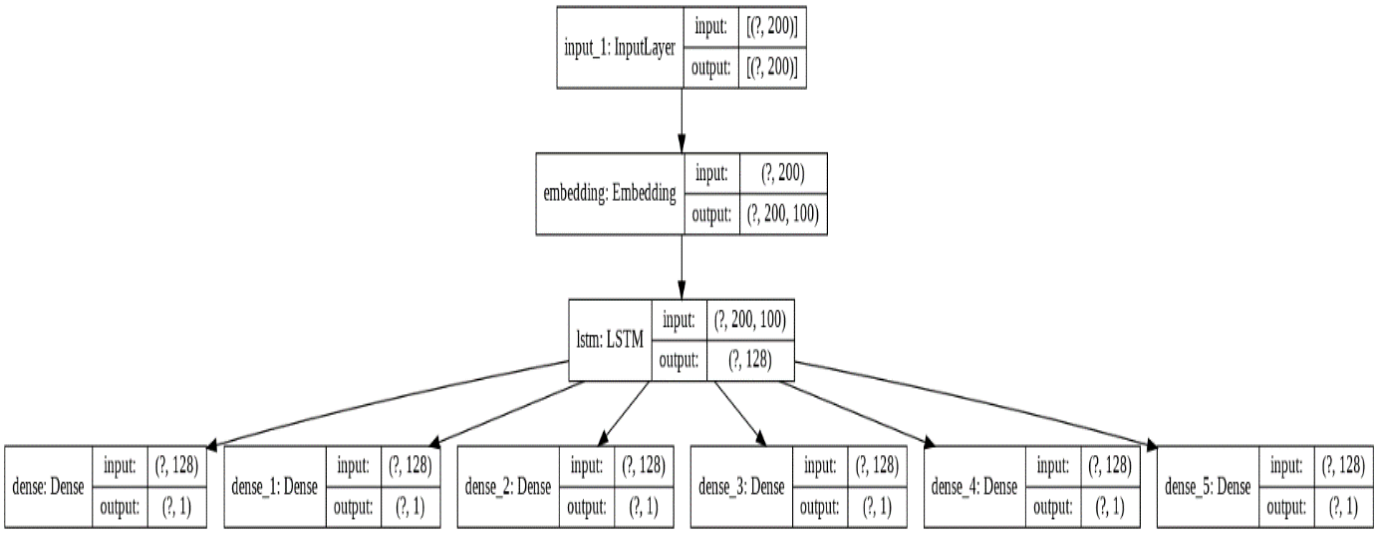

Figure 4. 10: Model Summary: Multiple Output Layers Using Movie Plots

in Recurrent (RNNs) and Convolutional Neural Networks (CNNs) (Géron, 2019:p.497; Bajpai, 2019; Sawarn, 2019). In the end, the six dense output layers carry the predictions for each

[79] https://nlp.stanford.edu/projects/glove/

one of the six emotions. After compiling, the model is fit and run with batch size 300 and for 20 epochs. As mentioned earlier, the performance was poor with a test accuracy of 50.26%.

4.4.4 Part2| Movie Plots with Single Output Layer

The same architecture of the right previous model was run, but this time using a single dense output layer, as it can be viewed in the next figure of the plot model summary:

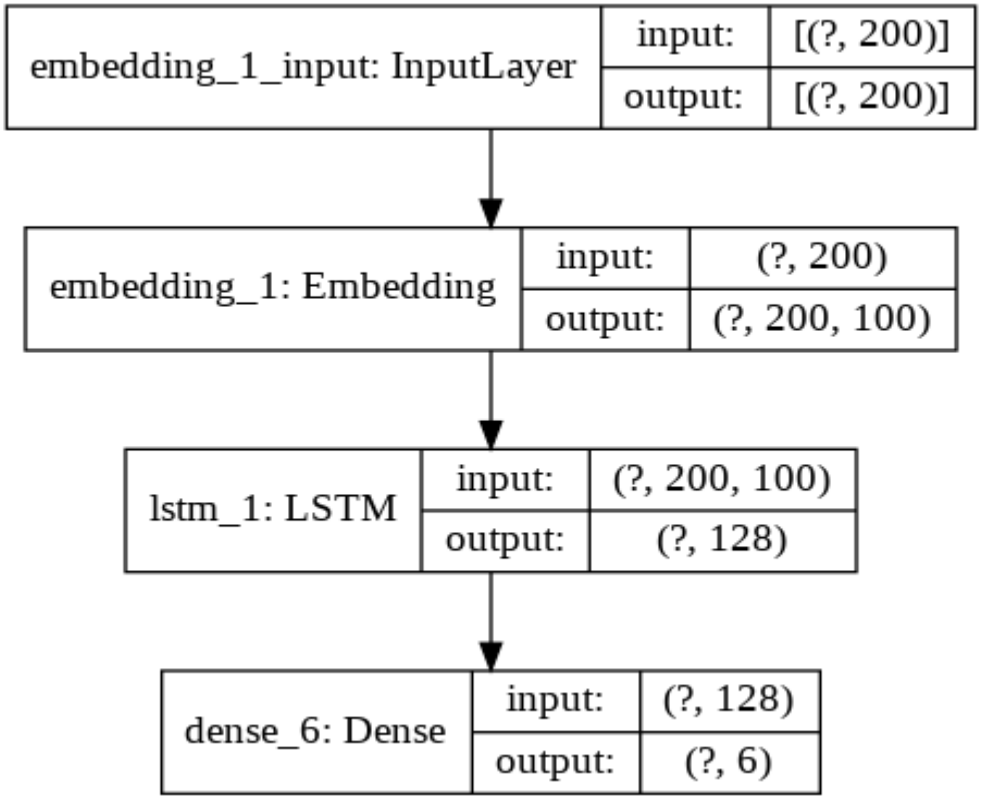

Figure 4. 11: Model Summary: Single Output Layer Using Movie Plots

This yielded greater results, with a validation accuracy score 62.93% and test accuracy 60%, and much less overfitting which can be understood both by the previous scores and from the following figure 4.12. As it can be seen below, the training and validation learning curves are close between them, a fact which implies that not too much overfitting is taking place.

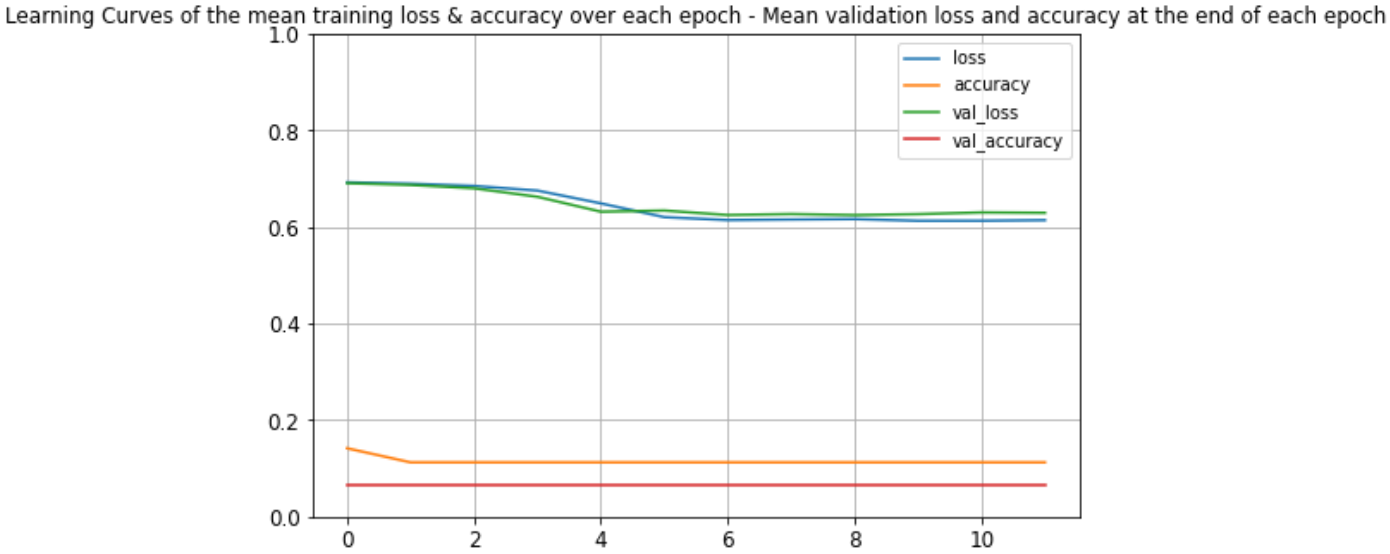

Figure 4. 12: Learning Curves of the Mean Training Loss & Accuracy Over Each Epoch - Mean Validation Loss & Accuracy at the end of each Epoch

The benchmark model in the area of DNNs came by an effort to improve the right above model. The respective model summary can be depicted in figure 4.13. The changes here were: the addition of a global max pooling layer (GlobalMaxPool1D())[80] and dropout regularization. CNNs are also capable of working well with text data (Géron, 2019:pp.497–498), and although pooling layers do not consist of weights (Géron, 2019:p.457), they aggregate the inputs with an aggregation function, e.g. the max or mean function. Here using the max function, the max pooling layer is located after the embedding layer and before the dense output layer, playing, moreover, the role of flattening the data before heading to the dense output layers (Schulz, 2018). The next change was consisted of the inclusion of 15% dropout regularization parameter for avoiding overfitting through regularization (Géron, 2019:pp.364–365).

[80] https://keras.io/api/layers/pooling_layers/global_max_pooling1d/

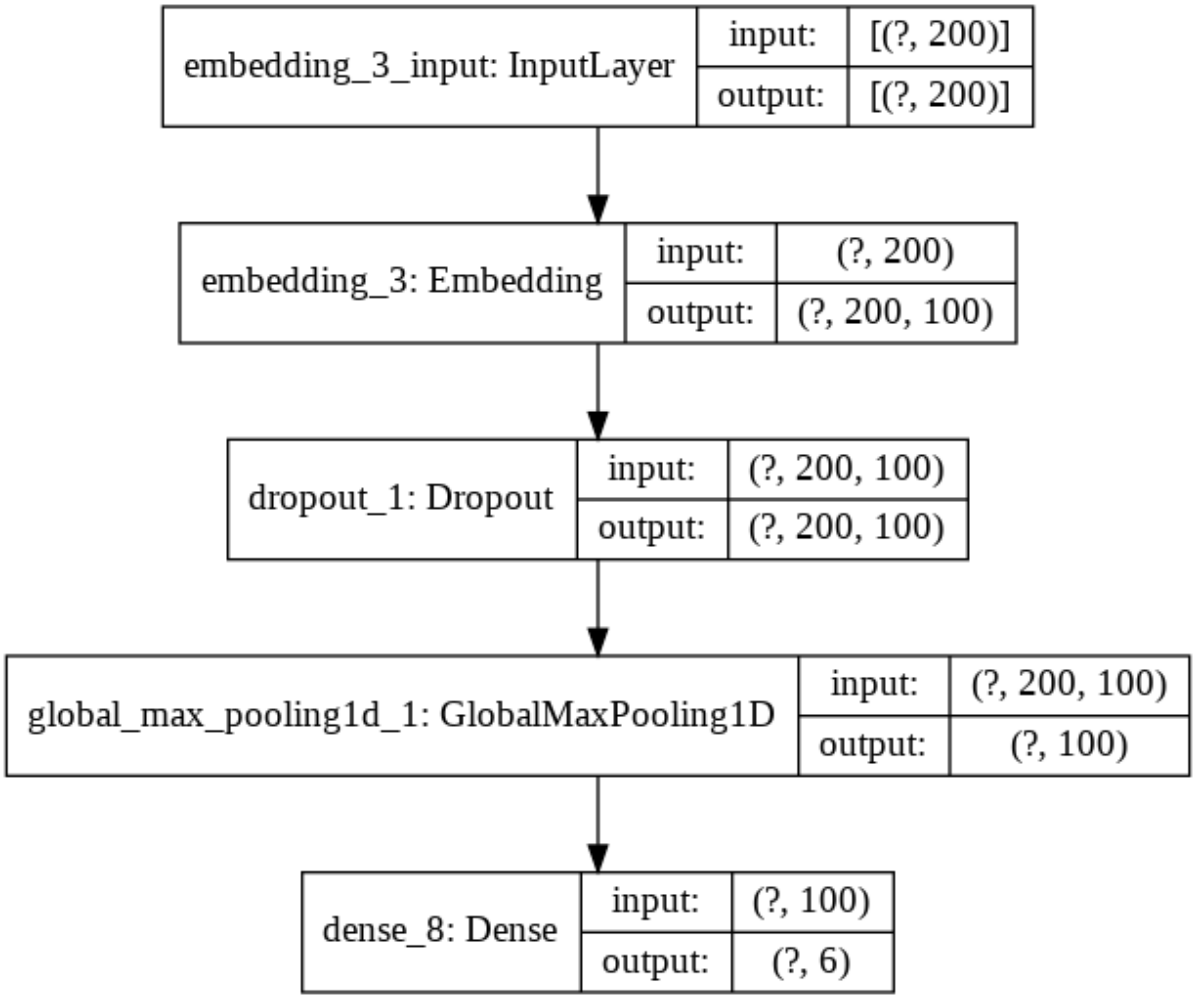

Figure 4. 13: Model Summary: Multiple Output Layers Using Movie Plots

This model gave approximately similar results in the validation accuracy score but here a higher test accuracy was obtained, a fact which implies less overfitting and higher levels of model's generalization.

4.4.5 Part 3| Movie Plots & Metadata - Multiple Output Layers

Contrary to the ML models built in previous sections, the DNN model here when using both movie overviews and metadata seem to underperform, with a test accuracy score of 59.51%. Its architecture has as following: Firstly, a functional and not a sequential API (Géron, 2019:p.307; Li, 2017) was used in order to gain the flexibility of creating two sub-models, since now there would be a need for a harmonic concatenation among the movie plots (1st set of predictors) and the movie metadata (2nd set). As figure 4.14 shows, the left part of the model's architecture is the same as previously, however this part now has turned into a sub-model and it should be concatenated with the second sub-model in the right part. The OneHotEndoder() preprocessing method applied in the categorical variables (movie genres, Vader's polarity) of the second sub-model, with regard to movie metadata, has created a sparse matrix with 16 columns, which

explains this number at that point. In the end, the two sub-models are concatenated through a Keras concatenate layer[81] which carries the outputs of the first and second sub-model, and finally the dense output layer with six neurons accepts the ensemble of those outputs for the final predictions.

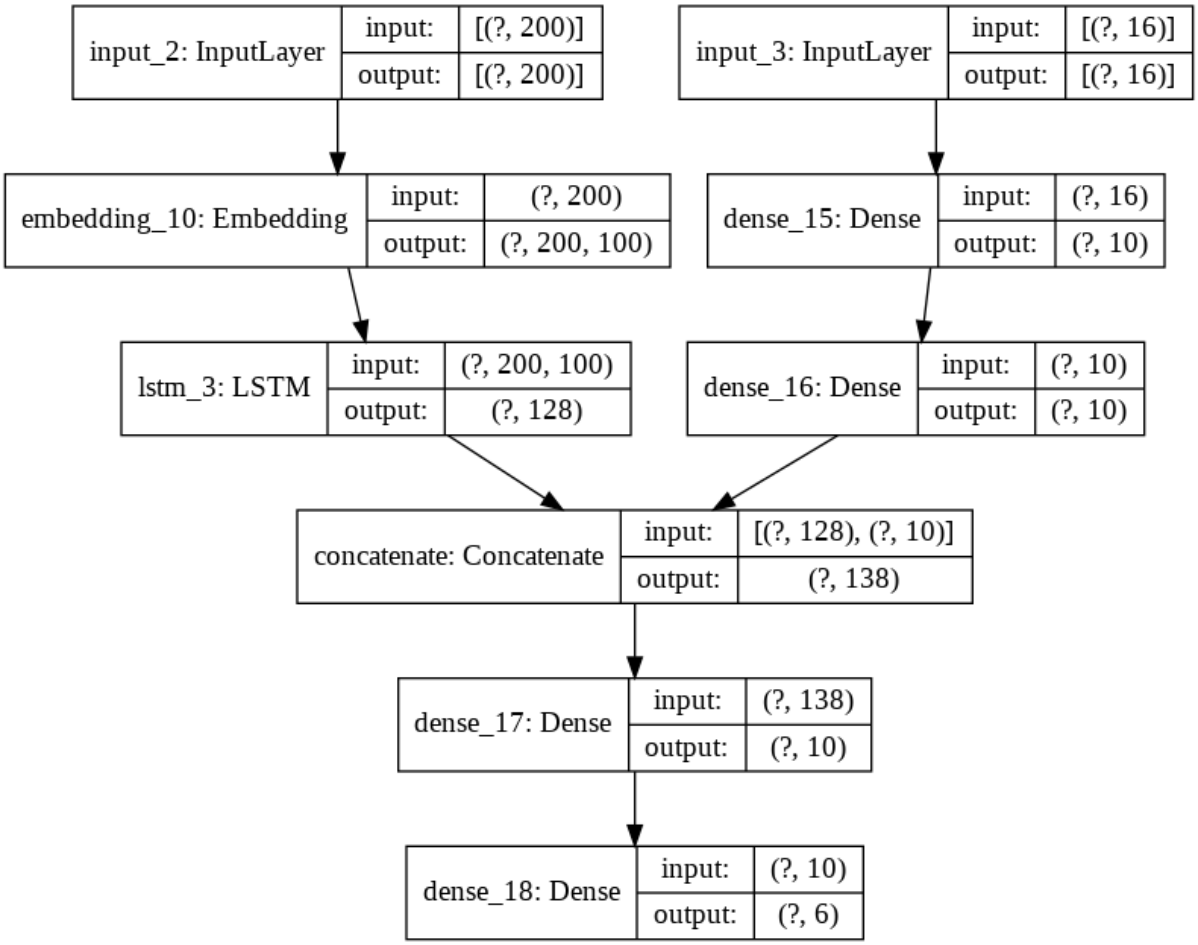

Figure 4. 14: Model Summary: Multiple Output Layers Using Movie Plots & Metadata

[81] https://keras.io/api/layers/merging_layers/concatenate/#:~:text=Concatenate%20class&text=Layer%20that%20concatenates%20a%20list,the%20concatenation%20of%20all%20inputs.

Chapter 5|

5. Correlation Tests

The researcher decides to investigate RQ3 and RQ4 by conducting two broad categories of correlation tests and by using two null hypotheses. It is reminded that RQ3 is: "Is there any correlation among user preferences and emotions expressed in the tv show/movie plots? Is, in the end, the notion of emotion a useful feature in the context of recommender systems and advertising companies?". The assumption considered here is that that there is a positive correlation between user preferences and ratings: if users vote a movie with a great rating then this means they liked it, and vice versa. By "user preferences" here is meant the ratings users vote, and by "emotions" is meant the emotions generated by the model constructed in the project. It is also reminded that RQ4 is: "Is there any correlation in the series of movies in users' watchlist with regard to the underlying emotions that these movies elicit? In other words and for example, if a viewer has watched 100 movies, are the depicted emotions of the first 50 movies correlated with those of the subsequent 50 movies?"

Let the null hypothesis be H_0:

- *H_0 for RQ3: User preferences are not linked with the emotions evoked by the movies they watch, and there is not a correlation among them.*
- *H_0 for RQ4: There is not a correlation among emotions in viewers' watchlists.*

5.1 Normality Tests

The researcher proceeds to normality tests in order to investigate if the variables participating in the hypothesis tests follow a normal (or not) distribution (Browniee, 2019a; Korstanje, 2019). It is noted that both "rating" and "emotion scores" are ordinal categorical variables (Donges, 2018). At first, when emotions were produced by the model in a binary form they were nominal categorical because they expressed the existence or not of the particular emotion in movies (in a form of yes/no). However, the researcher proceeds to the correlation tests using the emotional scores generated by the decision function scores of the final model, which they are in the form of confidence scores. This means that they now reflect an order in their production, and e.g. the "1" binary results are not the same across all emotions since some of those have less or more confidence score when they are displayed in movies.

An overall visualization of the variables can be depicted in the correlogram[82] of the following figure, where the scatter plots as well as the histograms of the rating and emotions variables in the diagonal line can be viewed.

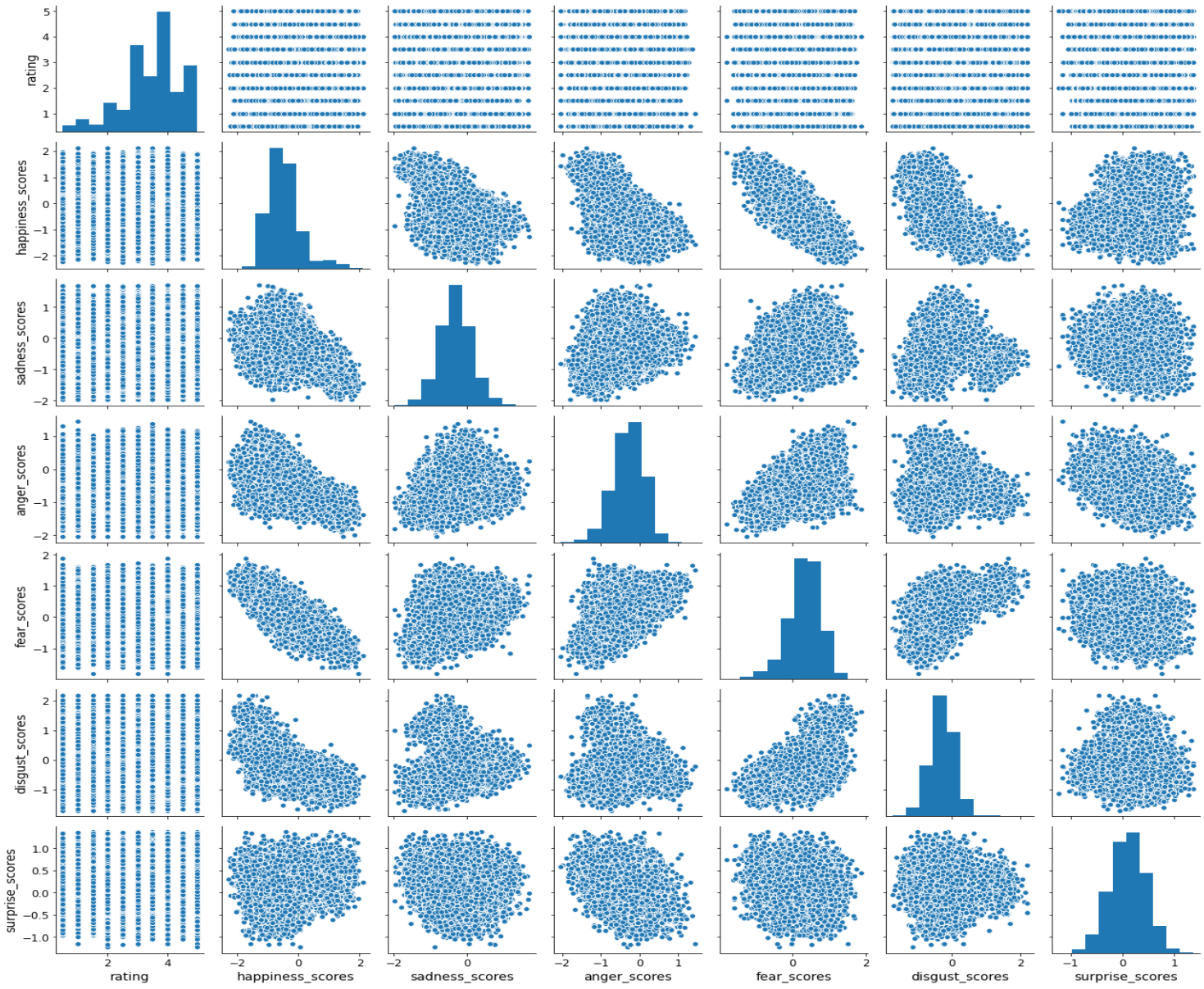

Figure 5. 1: Correlogram of Sample's Variables

The boxplot in figure 5.2 can be useful for visualizing the distribution of numerical data and skewness through displaying the data quartiles, as well for outliers detections (Galarnyk, 2018). The violin plot in figure 5.3 can extend this visualization by also showing the probability density of the data at different values and the complete variables' distributions (Lewinson, 2019). From the previous figure there can be no doubt that the "rating" variable does not follow a normal

[82] https://python-graph-gallery.com/correlogram/

distribution. Regarding the rest of the variables and overall, the different positions of boxplots seem to suggest areas of different group of values among the emotions variables. It can be noticed a different distribution across the 6 emotions with various outliers from the both axis sides, with emotions “fear” and “surprise” having more than 75% of their data towards the positive binary “1”, i.e. these emotions were displayed a lot across movies, contrary to the other emotions the majority of which carry negative confidence scores, especially that of “happiness”. Furthermore, more consistency in the scores are met in “disgust” scores, as opposed to “happiness” at which the inter-quartile range extends quite long. On the other hand, emotions “sadness”, “anger”, and “disgust” seem to present almost both equal median values and distributions. Nevertheless for all variables it can be noticed that although the median values seem to be around the middle of the corresponding boxes, the whiskers do not contain the same size between the lower and upper end of the boxes respectively.

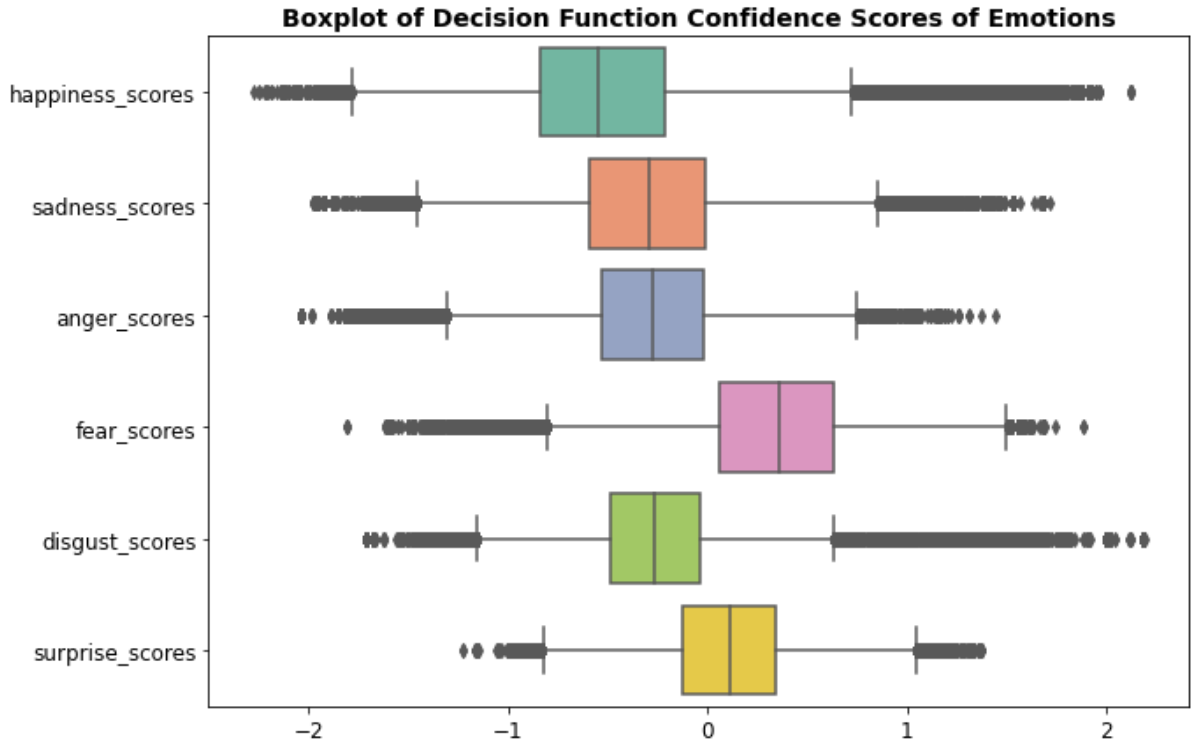

Figure 5. 2: Boxplot of Sample’s Variables

The aforementioned can further be investigated by the violin plot depicted below. The different data variation spread across the emotional scores can also be viewed, as well as the different positions of median values (white dots) between them. It can also be reconfirmed that emotions “fear” and “surprise” present higher density towards the positive values of the y axis, contrary to rest of the emotions and especially that of “happiness”, the latter of which however presents some outliers that extend higher than any other. This in other words would mean that

"happiness" does not appear often in movies, but in some restricted cases the model is pretty confident about its displacement.

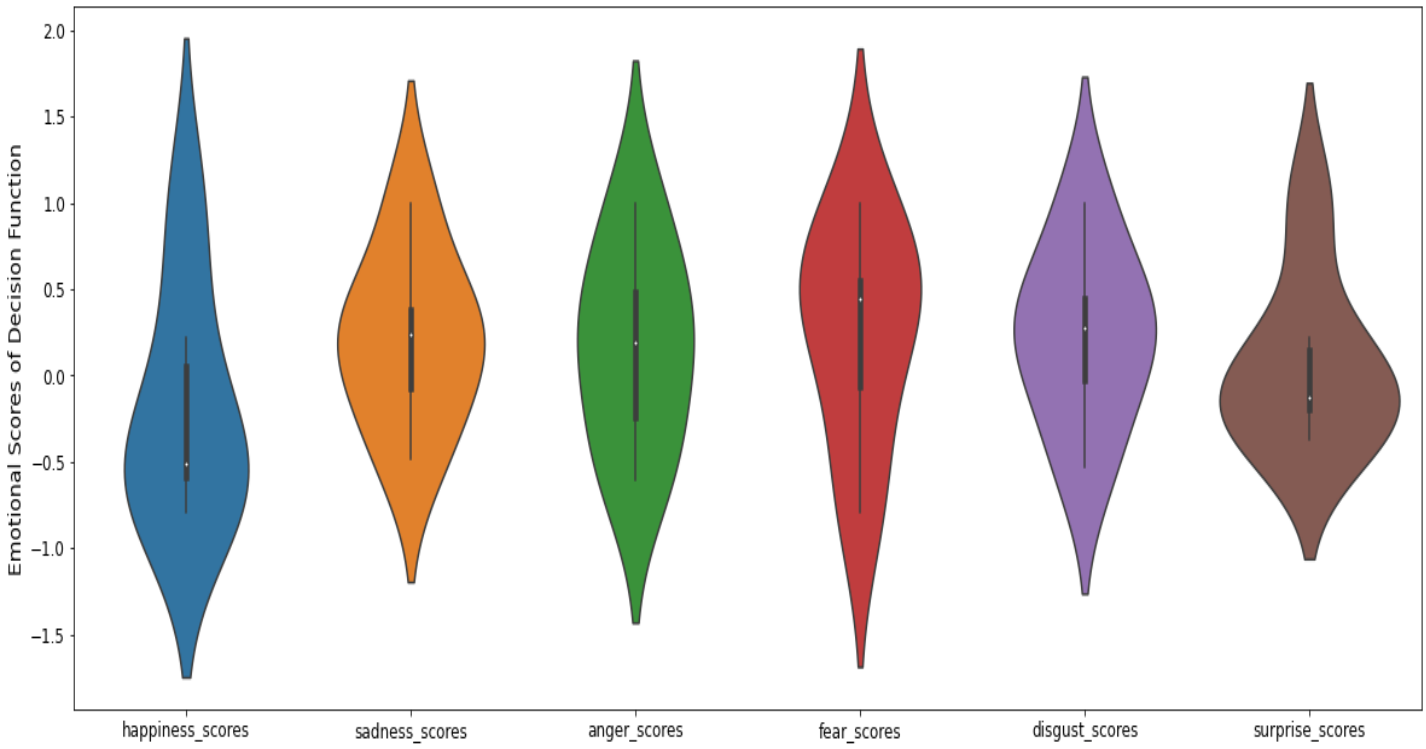

Figure 5. 3: Violin Plots of Sample's Variables

Overall different shapes and positions of violin plots might indicate different distributions across emotions and for each one of the emotions. Each emotion will be examined individually in the correlation tests, and specifically: The first broad category of tests will examine the correlation between rating scores and emotions (for each one of the six emotions respectively), and the first part of the second broad set of tests will compare the same but from a user-centric perspective. Lastly, the second part of the second broad category will investigate the relationship among the emotional scores: here rating values do not enter in the correlation tests, but what is of particular interest is that those emotional scores come from specific group of rating votes respectively, as will be explained later on.

A more strict identification of variable's normality (or not) would be desirable and despite the above indications, totally three normality tests were conducted via the statistical functions of scipy.stats[83], which is an open-source Python library, and these were: The Shapiro-Wilk[84] test[85],

[83] https://docs.scipy.org/doc/scipy/reference/stats.html

[84] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html

[85] Baded on the spicy documentation, this test might not be suitable for sample N>5000, and this is why two more tests were undertaken.

The D'Agostino's K-squared test[86], and the Anderson-Darling test[87]. In the below tables, "statistic" is the quantity value returned be the tests which is compared with the critical values from the distribution of the statistic tests (Browniee, 2019b). P-value will be used to interpret the results, i.e. if p< alpha=0.05 then the null hypothesis H0 gets rejected and means that samples were drawn from a Gaussian distribution, otherwise the results would fail to reject the H0. It should be noted than the p-value here is not the probability of the data fitting a Gaussian distribution, but rather it is used to interpret the respective statistical tests (Browniee, 2019a). As can be seen from the below tables, both Shapiro-Wilk[88] and D'Agnostino's K-squared tests find that the samples significantly deviate from normal and that the H_0 should be rejected.

	Statistics	**p-value**	**Result**
1st set of Variables: Emotions			
Happiness	0.952	<0.01	Rejection of H_0
Sadness	0.999	<0.01	Rejection of H_0
Anger	0.998	<0.01	Rejection of H_0
Disgust	0.939	<0.01	Rejection of H_0
Fear	0.991	<0.01	Rejection of H_0
Surprise	1	0.042	Rejection of H_0
2nd Variable:			
Rating	0.922	<0.01	Rejection of H_0

Table 5. 1: Normality Test: Shapiro-Wilk

	Statistics	**p-value**	**Result**
1st set of Variables: Emotions			
Happiness	6014.354	<0.01	Rejection of H_0
Sadness	76.374	<0.01	Rejection of H_0
Anger	125.383	<0.01	Rejection of H_0
Disgust	11566.956	<0.01	Rejection of H_0
Fear	1175.575	<0.01	Rejection of H_0
Surprise	11.387	<0.01	Rejection of H_0
2nd Variable:			
Rating	102945.202	<0.01	Rejection of H_0

Table 5. 2: Normality Test: D'Agnostino's K-squared

[86] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html
[87] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.anderson.html
[88] This test might not be accurate for sample N> 5,000, however this is why the two other tests are conducted.

Anderson-Darling test returns a list of critical values rather than a single p-value. The parameter “dist” in the scipy.stats.anderson() function is set to “norm” to check if there is a normal distribution. Each emotion’s statistic is compared with the critical values: If the statistic is greater than the respective critical values, the latter of which are with reference to the significance levels of 15%, 10%, 5%, 2.5%, 1%, then the H0 that the data comes from a normal distribution can be rejected.

	Statistic	**Critical Value**	**Result**
1st set of Variables: Emotions			
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Happiness	818.503	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Rejection of H_0
		1%: 1.092	Rejection of H_0
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Sadness	15.946	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Rejection of H_0
		1%: 1.092	Rejection of H_0
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Anger	19.923	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Rejection of H_0
		1%: 1.092	Rejection of H_0
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Disgust	526.555	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Rejection of H_0
		1%: 1.092	Rejection of H_0
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Fear	149.868	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Rejection of H_0
		1%: 1.092	Rejection of H_0
		15%: 0.576	Rejection of H_0
		10%: 0.656	Rejection of H_0
Surprise	0.869	5%: 0.787	Rejection of H_0
		2.5%: 0.918	Fail to Reject H_0
		1%: 1.092	Fail to Reject H_0

2nd Variable:			
		15%: 0.576	Fail to Reject H_0
		10%: 0.656	Fail to Reject H_0
Rating	32820.415	5%: 0.787	Fail to Reject H_0
		2.5%: 0.918	Fail to Reject H_0
		1%: 1.092	Fail to Reject H_0

Table 5. 3: Normality Test: Anderson-Darling

Overall, apart from the Anderson-Darling test which indicated for emotion "surprise" that the H_0 fails to be rejected regarding the significance levels of 2.5% and 1%, it can arguably be supported that the normality tests indicated that there is not a Gaussian distribution, and that ratings as well as the emotional scores for each one of the emotions do not look normal. As a consequence non parametric statistical methods should be used for the hypothesis tests.

5.2 Hypothesis Tests – Spearman's Rank Correlation Coefficient

After confirming what type of correlation tests should be constructed, a range of non-parametric tests were investigated. The researcher eventually proceeds with Spearman's rank-order correlation, which is the non-parametric version of the Pearson's product-moment correlation, for the following reasons (Browniee, 2019c; Statistics, 2020; Shantikumar, 2016; Okada, 2019; Statstutor, 2020; Geographyfieldwork, 2020; Laerd, 2020):

i. Spearman's rank order correlation presupposes that the two variables measured should be ordinal, interval or ratio scale. In this task, both variables are ordinal categorical and meet this criteria.

ii. Spearman's test also requires that the two variables represent paired observations. This, indeed, is met here. For example if a user has watched 50 movies, then the rating from the 1st movie will be investigated with regard to the second variable (emotional score) for that particular movie, next this will take place again regarding the 2nd movie, and so on until the 50th movie.

iii. The assumption that a monotonic relationship between the variables exists: A monotonic relationship exists when either the variables increase in value together (positive correlation), or as one variable value increases, the other variable value decreases leading to an inverse monotonic relationship (negative correlation). The degree and nature of this correlation remains to be investigated throughout the tests' conduction.

iv. Spearman's correlation can be used when the variables are not normally distributed and it is not very sensitive to outliers: from the three normality checks it was confirmed that the

variables do not follow a normal distribution, and the outliers spotted through the boxplots would not influence the validity of Spearman's test results.

For the reasons explained, the researcher proceeds with Spearman's correlation coefficient to test the null hypotheses (H_0) that there is no monotonic correlation in the population, against the alternative (H_1) that there is monotonic relationship. The researcher, therefore, chooses the significance level (alpha) be the p-value of 5%, and lets rho be the Spearman's population correlation coefficient. A Spearman's rank correlation coefficient will be calculated in the sample of every test conducted. Rho values range in [-1 , 1], and correlation with rho +1 or -1 indicate an exact monotonic relationship, a perfect negative or positive relationship respectively. The strength of the correlations based on the rho values will be grouped and interpreted as followed:

- [-0.5 , -1]: strong negative relationship
- [-0.3 , -0.5]: moderate negative relationship
- (0 , -0.3]: weak negative relationship
- (0 , 0.3]: weak positive relationship
- [0.3 , 0.5]: moderate positive relationship
- [0.5 , 1]: strong positive relationship

Both p-value and Spearman's rho will be calculated via the spicy.stats.spearmanr[89] statistical function using Python. P-value will be the probability of 5% being the threshold for detecting if there is or not a correlation among the variables: above this level (p>0.05), the null hypothesis is considered correct. At or below this level (p <= 5) will mean that there is at least a 95% probability that the null hypothesis is wrong and that the data are statistically significant showing a true relationship, and having enough evidence to reject the null hypothesis, and lastly, that the samples were likely drawn from populations with different distributions.

Figure 5.4 depicts the construction of all correlation tests. As it can be seen, two broad categories of correlations tests were conducted.

[89] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

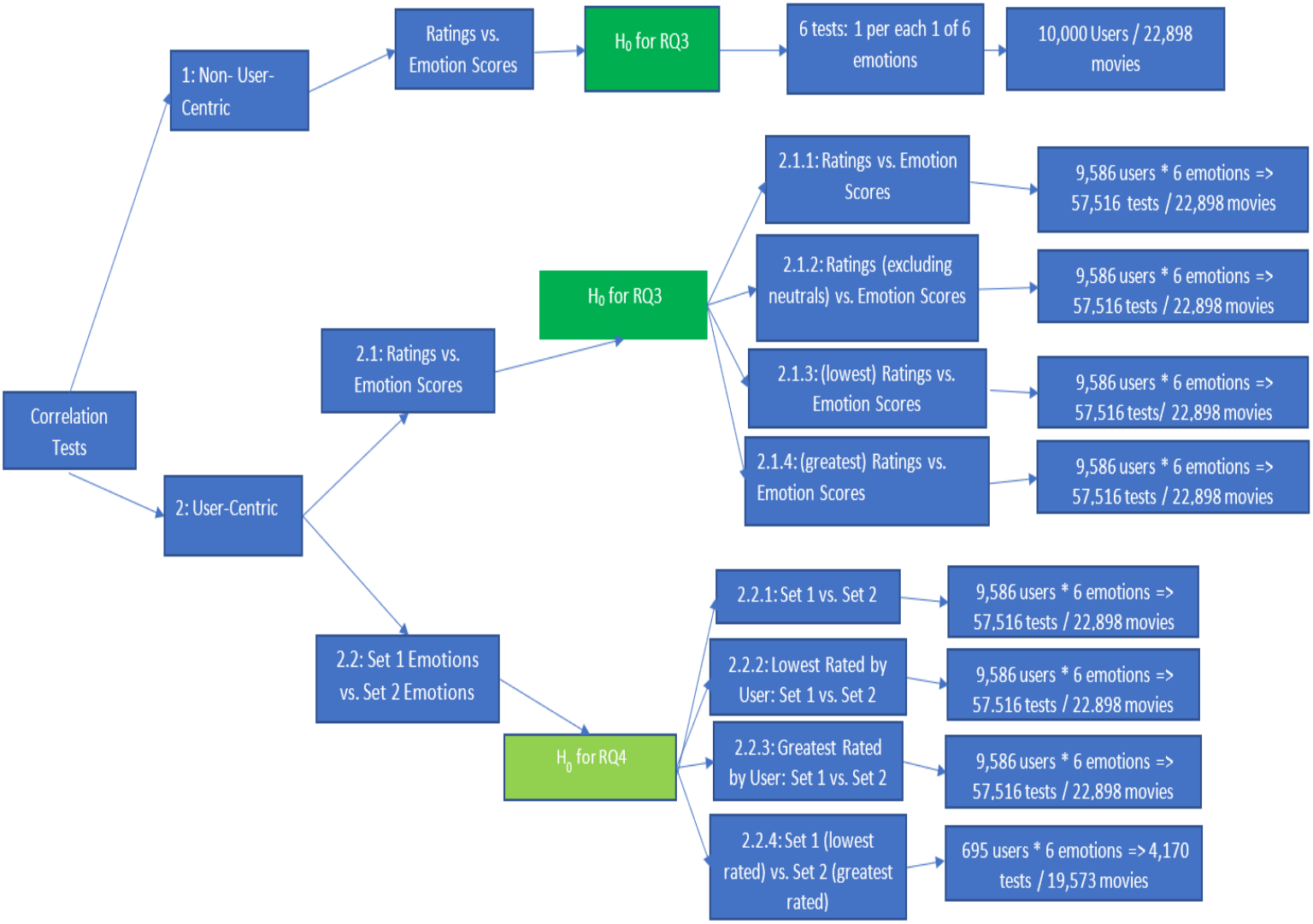

Figure 5. 4: Construction of Correlation Tests

5.3 Hypothesis Tests: Category 1

The first category is non user-centric, the rating scores comprise the independent variable whereas the emotional scores the dependent ones. By "non user-centric" is meant that all data is taken into account without taking into consideration which users have voted which movies. Here, the null hypothesis answering the RQ3 will be investigated, and since each emotion will be examined separately a total number of 6 tests were conducted. All data was taken from 10,000[90]

[90] 10,000 is considered by the researcher a quite sufficient number of users and this is why this number was chosen. These users come from the "ratings_10k.csv" file. This csv is a reduced version of the MovieLens "ratings.csv" which contains 162,541 users in total. 10,000 was picked up as followed: first 3000 users with userId in range [1 , 3000], 3000 users with userId in range [100000 , 103000], and the last 3000 users with userId in range [158543 , 162541]. The rationale behind that was to grab users across all years, since based on MovieLens' documentation those users were created between 1995 and 2019.

users, each one of them has watched 150 movies in average, and all users aggregately have watched 22,898 movies.

5.4 Hypothesis Tests: Category 2

The second broad category is user-centric in that all tests were conducted per unique user (and per emotion). Apart from test "2.2.4", for all tests here the number of sample is 9,586 users (414 less) because only users having watched at least 20[91] movies were chosen, and these users have all together watched 22,892 movies.

The first part of tests (tests 2.1.1 – 2.1.4) is similar with that of the non-user-centric approach in the first category of tests: rating scores are compared with emotional scores, but here per unique user (and per emotion), and the null hypothesis with regard to RQ3 is investigated. The second part explores the null hypothesis regarding RQ4 and users' watchlists. Here ratings are not examined in the test, but they play an important indirect role in that they define which emotional scores will be used each time per test.

5.4.1 Tests: Category 2.1

Six hypothesis tests were conducted with the Spearman's rho correlation coefficient:

1) The results indicated a weak negative correlation between "happiness" emotion and rating scores, ($rho = -0.0243$, $N=10,000$, $p < 0.01$)
2) The results indicated a weak positive correlation between "sadness" emotion and rating scores, ($rho = 0.0459$, $N=10,000$, $p < 0.01$)
3) The results indicated a weak positive correlation between "anger" emotion and rating scores, ($rho = 0.0119$, $N=10,000$, $p < 0.01$)
4) The results indicated a weak positive correlation between "disgust" emotion and rating scores, ($rho = 0.0130$, $N=10,000$, $p < 0.01$)
5) The results indicated a weak positive correlation between "fear" emotion and rating scores, ($rho = 0.0341$, $N=10,000$, $p < 0.01$)
6) The results indicated a weak negative correlation between "surprise" emotion and rating scores, ($rho = -0.0113$, $N=10,000$, $p < 0.01$)

[91] This is so that the sets of the tests which consider two sets of movies by the unique user have at least 10 movies each.

***Test 2.1.1 – Ratings vs. Emotion Scores**:*

Let us suppose that the user with id=1 has watched 100 movies. Then, Spearman's test will correlate the rating scores voted by the userId=1 for those 100 movies, with the corresponding emotional scores assigned to those movies (originated from the model). This procedure takes places for every user and per emotion, and leads to the total number of 57,516 tests per subcategory of tests, a total of 402,612 tests plus 4,170 tests of the "2.2.4" test, for the second category of tests. The total number of tests conducted both in the two broad categories is 406,788[92].

Results can be depicted in the below figure. 90.54% of the sample (52,076 tests) showed no correlation with regard to RQ3, whereas the rest 9.46% (5,440 tests) rejected the null hypothesis. From the sample which rejected the null hypothesis (p<=0.05), the blue bars in the grouped bar plot indicate the number of tests which showed correlation per emotional class.

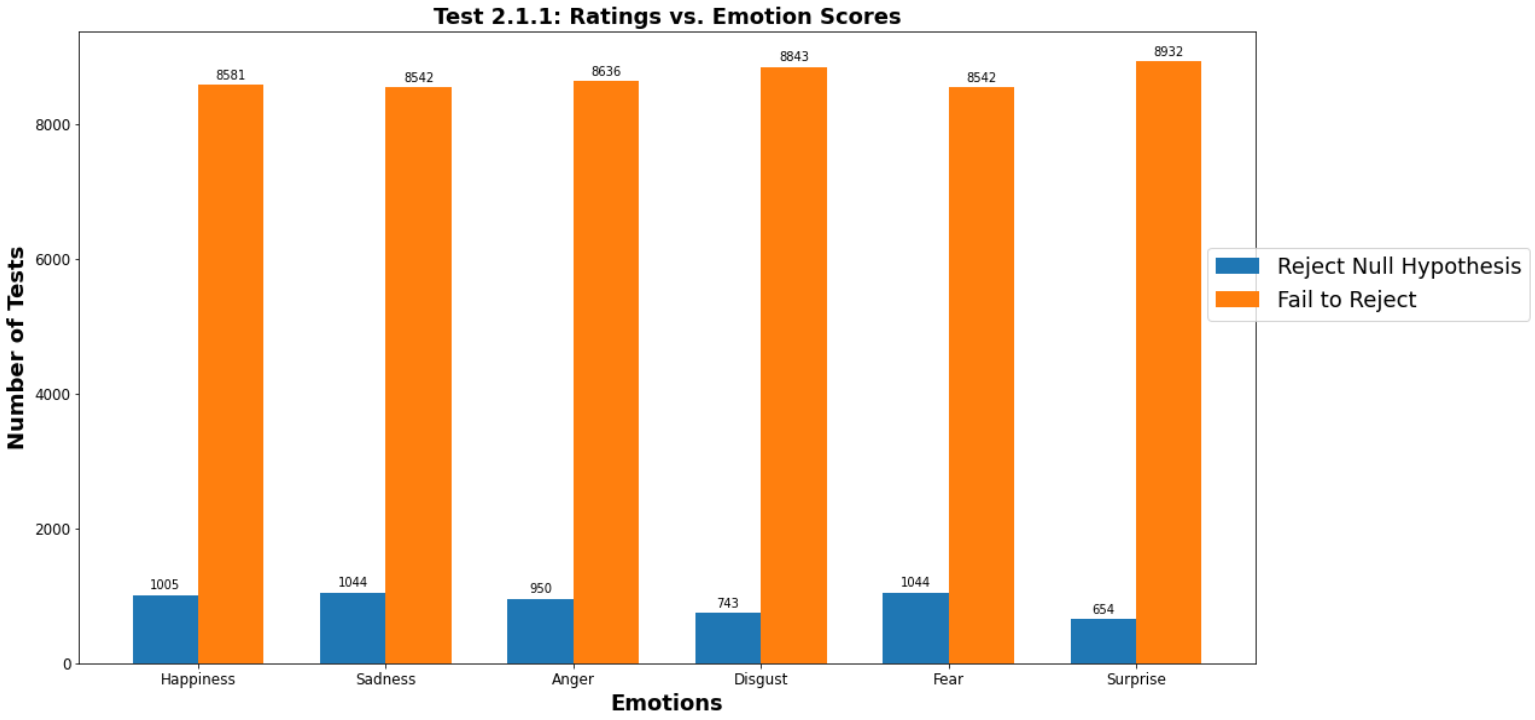

Figure 5. 5: Correlation Test (2.1.1): Ratings vs. Emotion Scores

[92] 1st category: 6 tests | 2nd category: 402,612 tests (tests 2.1.1 – 2.2.3) + 4,170 tests (test 2.2.4).

Test 2.1.2 – Ratings (excluding neutrals) vs. Emotion Scores:

Same rationale as the previous 2.1.1 test, but here without taking any data which is linked with neutral ratings. The researcher defines the categories of ratings to be as followed:

- Ratings in range [0.5 , 2]: negative ratings
- Ratings in range (2 , 3.5]: neutral ratings
- Ratings in range [4 , 5]: positive ratings

Example: A user has watched 60 movies but he has voted 20 movies with neutral ratings. Thus data from those 20 movies will not be included in the test. This is executed per emotion, per user. The results are:

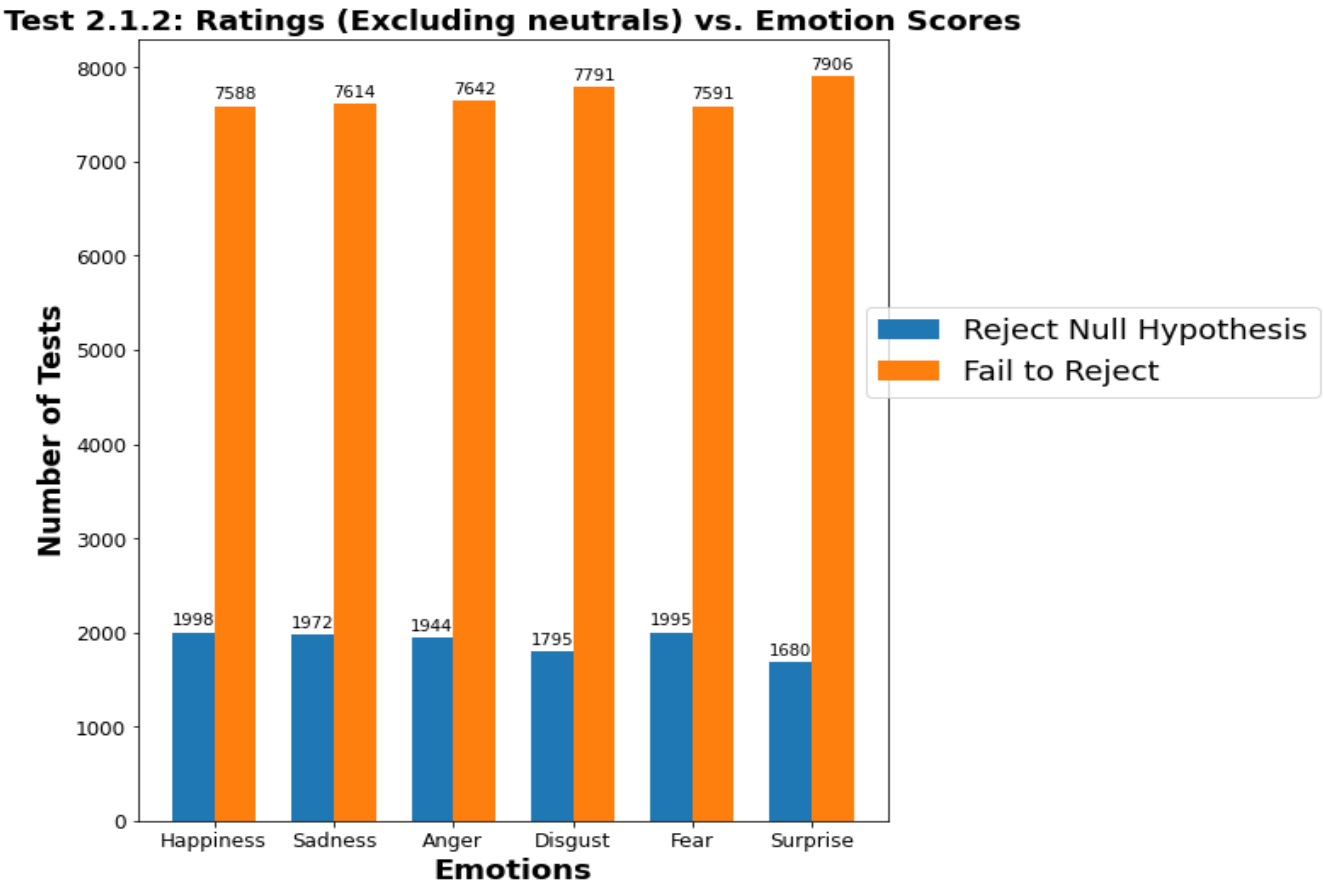

Figure 5. 6: Test 2.1.2: Ratings (Excluding neutrals) vs. Emotion Scores

Here, 80.21% of the sample failed to reject the null hypothesis, whereas 11,384 tests (19.79%) rejected the null hypothesis for RQ3.

Test 2.1.3 – Lowest Ratings vs. Emotion Scores:

Same as test 2.1.2, but here taking only the negative ratings. The results are shown below:

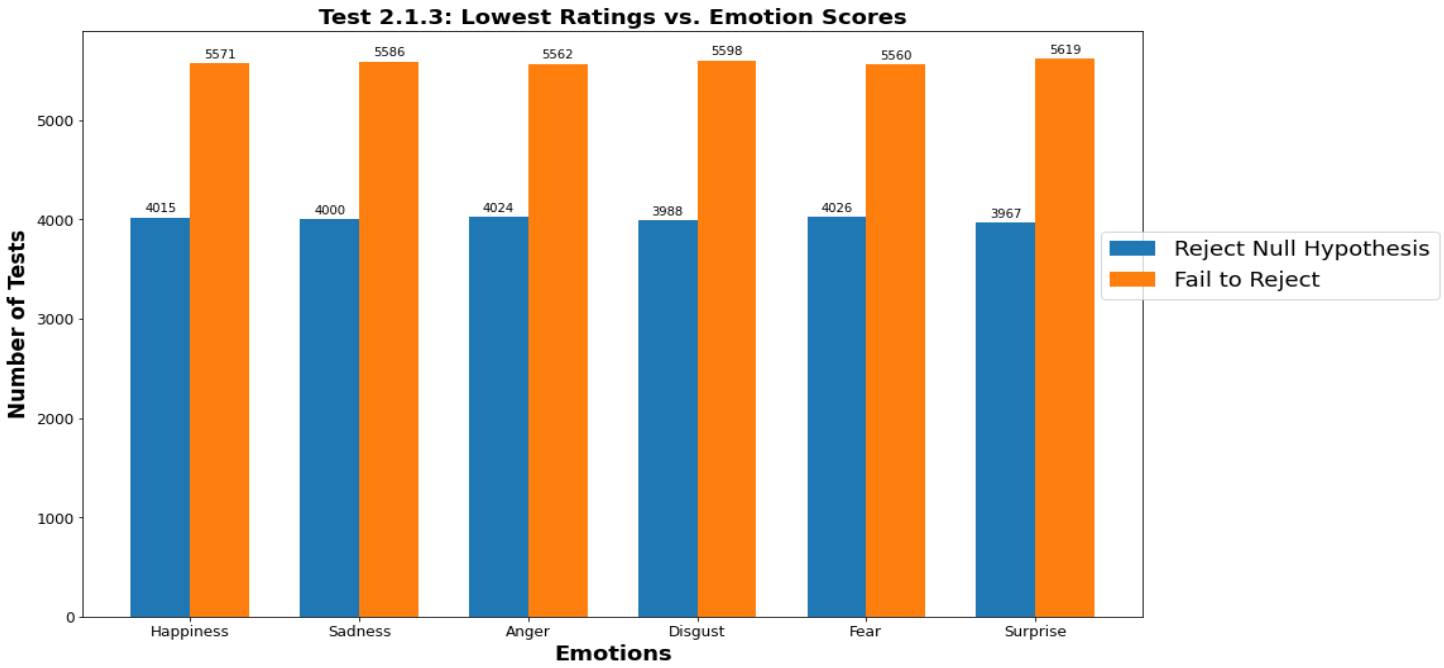

Figure 5. 7: Test 2.1.3: Lowest Ratings vs. Emotion Scores

One of the interesting findings take place in this test: while 58.24% of the sample fails to reject the null hypothesis, however in 41.76% (24,020 tests) the null hypothesis for RQ3 gets rejected.

Test 2.1.4 – Greatest Ratings vs. Emotion Scores:

The rationale here is the right opposite of that of 2.1.3, and only data derived from those movies that each time the user has voted in range [4 , 5] are taken. The results:

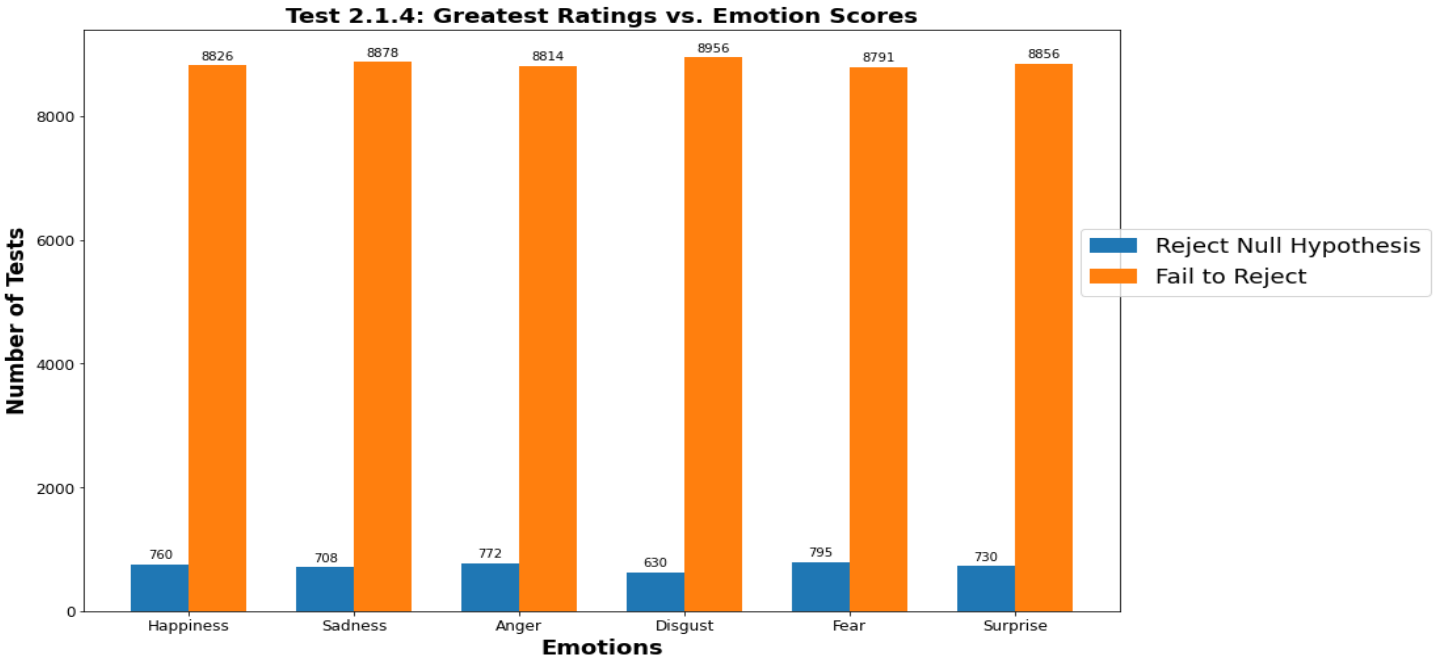

Figure 5. 8: Test 2.1.4: Greatest Ratings vs. Emotion Scores

5.4.2 Tests: Category 2.2

As mentioned earlier, this subcategory of tests (2.2.1 – 2.2.4) investigates the H0 with reference to the RQ4. The researcher here is interested in investigating potential relationships between the movies that the same user has watched, regarding the evoked emotions behind those movies. For example, when someone has watched 30 movies, do the first set of 15 movies correlate with the 15 movies of the second set with regard to the emotions elicited from those movies?

Test 2.2.1: Set 1 vs. Set 2

In this test all movies that the unique user has watched are split by half creating 2 equal sets. If the total number of movies watched is an odd number, then the last movie watched will be subtracted. Same rationale as tests 2.1.1 – 2.1.4, this procedure takes place for every unique user in the sample, and per emotion. Figure 5.9 depicts the results:

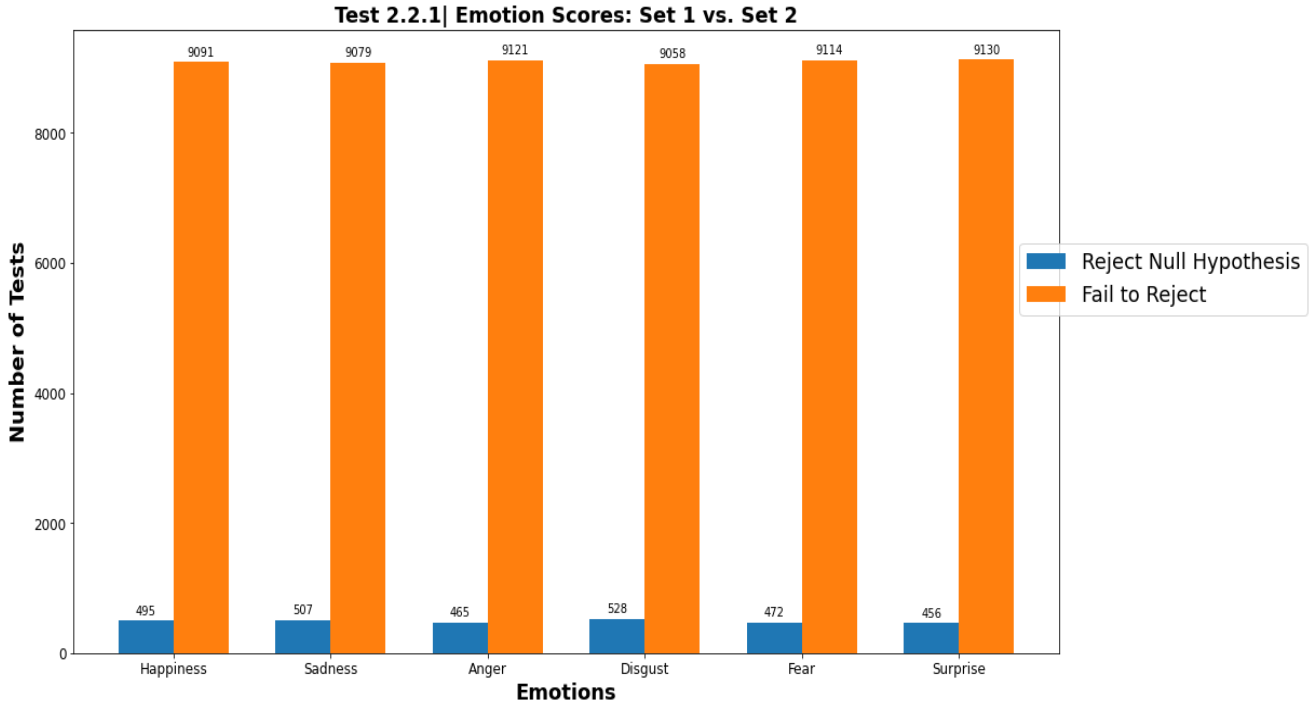

Figure 5. 9: Test 2.2.1 - Emotion Scores: Set 1 vs. Set 2

Overall, 54,593 tests showed no correlation and failed to reject the H0 relating to RQ4. 5.08% of the sample rejected the null hypothesis.

Test 2.2.2: Lowest Rated by the User: Set 1 vs. Set 2

Same as test 2.2.1, but here taking those sets of movies per unique user that the user has voted the least (i.e. ratings in range [0, 2]). The results:

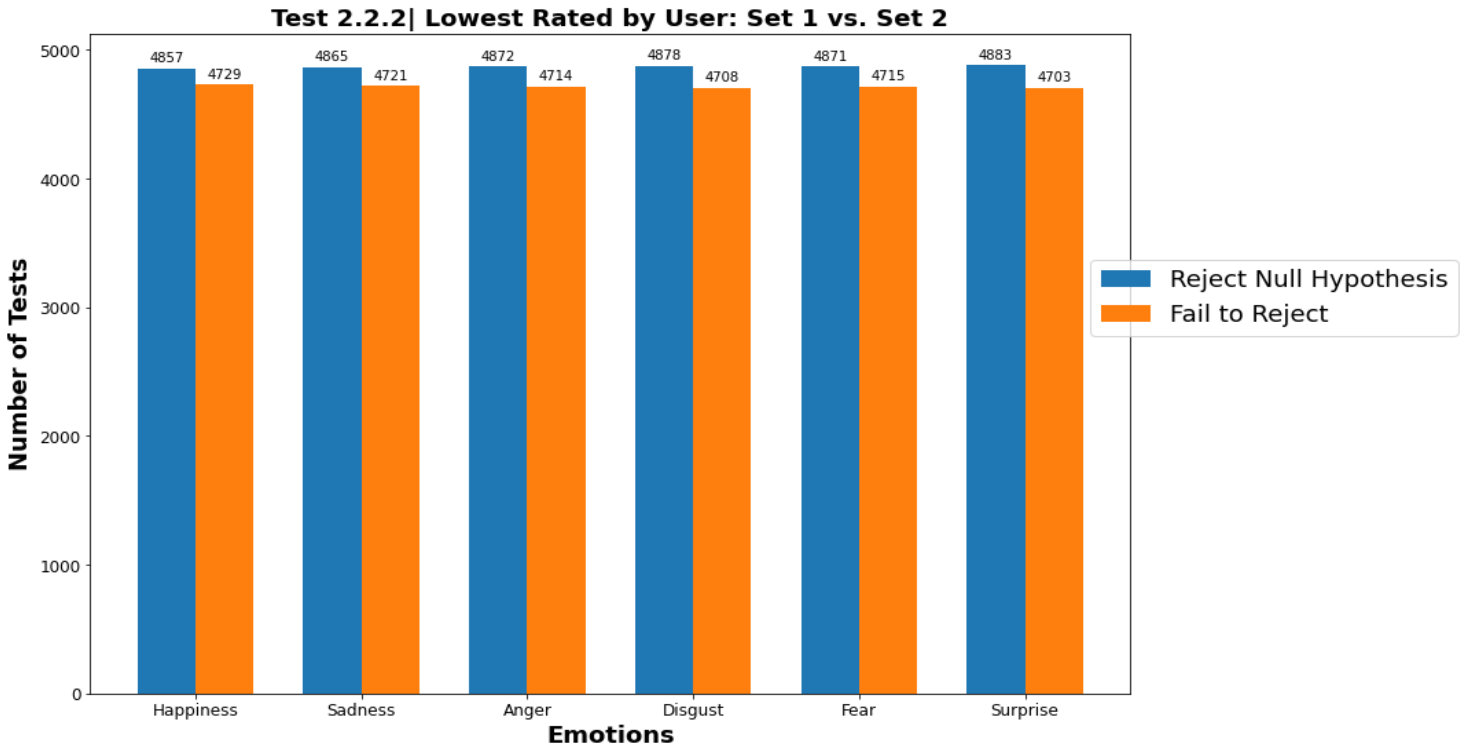

Figure 5. 10: Test 2.2.2 - Lowest Rated by User: Set 1 vs. Set 2

This is the test with the most significant results in terms of correlation found. While 49.19% of the sample failed to deny the H_0, 50.81% of the sample with a total of 29,226 tests rejected the H_0 in regard to RQ4. In average, this holds true at the same level for every one of the six emotions, where the majority of each one of the emotional classes rejects the H_0.

Test 2.2.3: Greatest Rated by the User: Set 1 vs. Set 2

The opposite of test 2.2.2, by taking the data derived from movies voted with ratings in range [4 , 5] by the unique user. The results are:

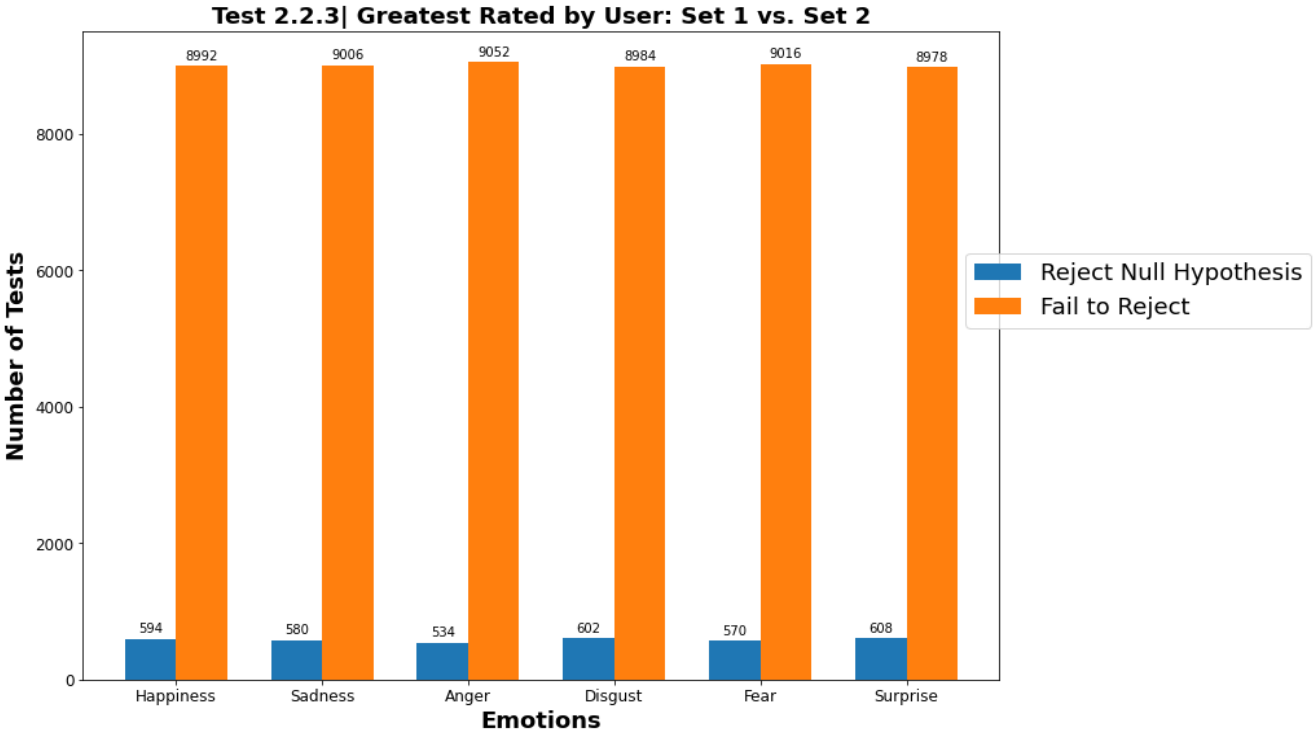

Figure 5. 11: Test 2.2.3 - Greatest Rated by User: Set 1 vs. Set 2

As can been from the above figure, by conducting in a way the opposite of the previous test the vast majority here (93.94%) fails to reject the null hypothesis with reference to RQ4, with only 6.06% of the sample rejecting it.

Test 2.2.4: Set 1 (lowest rated) vs. Set2 (greatest rated)

Lastly, this is a combination of tests 2.2.2 & 2.2.3. This means that in this test the first set of movies consists of negative ratings, while the second set is comprised of those movies voted with scores in range [4 , 5] by the unique user. The reason why this test has a different sample is justified by the fact that not all users who have watched at least 20 movies can fulfil the criteria put in place for this test. For this reason from the 9,586 users, totally 695 participated in this test, and each one of those have watched 50 movies voted with negative ratings by the unique user (set 1), and 50 movies voted with positive ratings (set 2). The results for these tests are illustrated below:

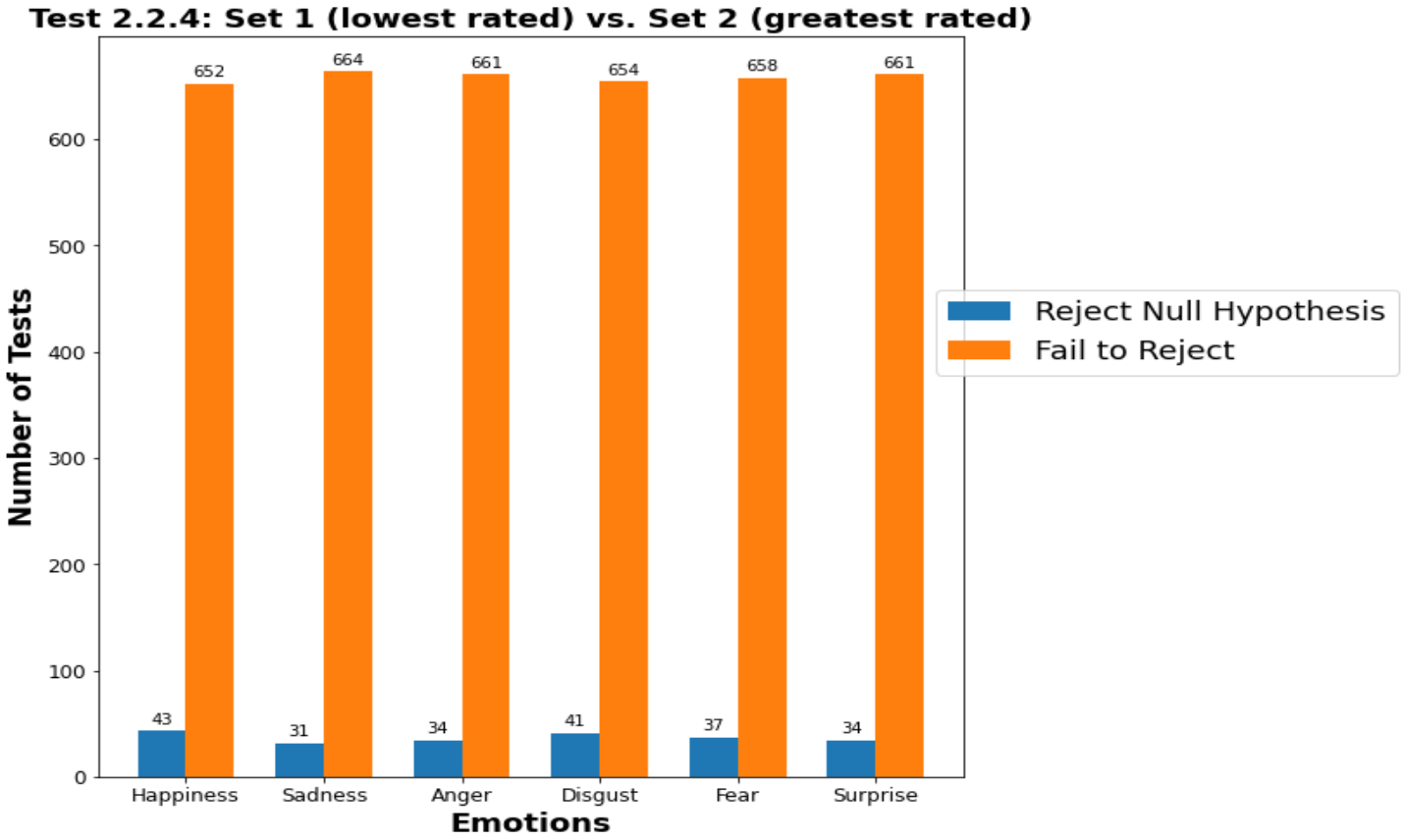

Figure 5. 12: Test 2.2.4: Set 1 (Lowest Rated) vs. Set 2 (Greatest Rated)

And here, the vast majority of the sample with 3,850 tests failed to reject the H_0 with respect to RQ4. The rest of 5.28% of the sample (220 tests) rejected the null hypothesis.

5.4.3 Additional Comments

Closing this chapter, some comparative observations are worth mentioning. A similarity in the logic behind the construction between the two subcategories (2.1 & 2.2) of the user-centric broad category of tests can be detected: the data used in correlation tests for both tests 2.1.1 and 2.2.1 are with regard to all movies watched by the unique user. Similarly, test 2.1.2 and test 2.2.4 ignore movies which have been voted by neutral rating scores by the unique users. Subsequently, a similar pattern is followed between tests 2.1.3 and 2.2.2: test 2.1.3 takes into account only the lowest ratings (in the variable “rating”), and similarly, in test 2.2.2 the emotional scores are linked to those movies which have received the lowest rating scores by the respective unique user. Lastly, test 2.1.4 encompasses the best ratings in the “rating” variable, and accordingly, test 2.2.3 correlates those emotional scores that originate from movies voted only with the greatest rating votes by the corresponding unique users.

Chapter 6|

6. Conclusions & Discussion

6.1 Conclusions

During the search of a suitable model for the prediction of emotions in the unlabeled dataframe consisting of 55,577 movies, the linear SVClassifier with the TF-IDF vectorizer seemed to clearly outperform the rest of the models. Specifically, the generic TF-IDF Vectorizer obtained better results by a factor of about 3% in terms of accuracy in the validation set, compared to the benchmark DNN model using word embeddings and CNN pooling layers. After finding the most promising model, this accuracy score was boosted via the GridSearchCV model selection in the context of finding the best hyperparameters (micro average f1 score: 69.06%), and consequently this score rose by approximately 4% (micro average f1: 73.15%) when the model was trained in 85% of the unlabeled dataframe, and when the final feature selection was decided through Random Forest's feature importances. The hamming loss score for the final model was 0.21 and the micro ROC AUC score stood at 0.75.

The final model was then used to make the final predictions with reference to emotions of totally 55,577 movies. The total number of 402,612 correlation tests was conducted with Spearman's rank-order correlation coefficient in order to address RQ3 and RQ4. In the first major category of tests and with regard to RQ3, all samples rejected the H_0, with emotions "happiness" and "surprise" having a weak negative correlation between ratings and emotional scores, while a weak positive association was found in emotions "sadness", "anger", "disgust", and "fear". This means that the respective H_0 was wrong, that the statistical data among emotions and ratings are statistically significant and that they show a true relationship. For this reason, here the researcher accepts the alternative hypothesis H1, which for RQ3 means that there is a correlation (its grade described above per emotion) between user preferences and emotions. Extensively, this means that the above results can be important pieces of information for the RSs and advertising companies to further investigate, exploit and integrate into their systems.

In the 2nd phase (user-centric), tests 2.1.1 – 2.2.4 addressed the H_0 respecting RQ3, and tests 2.2.1 – 2.2.4 dealt with the H_0 with reference to RQ4. The majority of users in the 1st subcategory of this section failed to reject the H_0 across each one of the six emotions. However, in test 2.1.3 the population of disagreement was quite close, with 41.76% of users (24,020 tests

/ 4,003 users) rejecting the H_0 and 58.24% (5,583 users) failing to reject it. In the second subcategory, the majority of users in three out of the four tests conducted failed to reject the H_0 with reference to the RQ4, contrary to the test 2.2.2 which accepted the alternative H_1 hypothesis from 50.81% of the sample (29,226 tests / 4,871 users).

6.2 Discussion

In relation to the RQ1 and to what is a sufficient number of emotional tags and which there are, based on the literature review and the work conducted, there is not a right answer than can be given for that. Rather, the answer depends on the purpose of the task, and on the nature and category of emotions aimed to be extracted, and the answer to that would subsequently estimate the right type and number of emotional tags respectively. It is worth mentioning, however, that the greater the range of emotional tags aimed to be produced automatically by an ML model, the greater the amount of labelling is required for all the occurrences of all emotional instances. Only then would any model be able to successfully learn the patterns which discriminate the one from the other emotions.

Respecting which models can be suitable for the production of emotional tags and if an automatic generation of those can be implemented (RQ2), the researcher here answers that both neural and non-neural network ML models can be appropriate. This would be accompanied with several NLP techniques as those conducted in the project, and the general ML tools would depend from the task and the size of the data available. The surprisingly better performance of the TF-IDF vectorizer compared to the rest of the models for the project's task might be justified due to the fact that word embeddings and neural networks could possibly overfit the data: Contrary to the TF-IDF, word embeddings contain a more "noisy" signal, having a more complex word representation and a lot of more hidden information for capturing the relationship between words among documents (Cam-Stei, 2019). The small amount of training data possibly gave not enough space for word embeddings to assign the rules for capturing those patterns in the sequences of words.

Lastly, respecting RQ3 & RQ4, and concerning the 1st broad category, the tests there indicated a correlation among ratings and the elicited emotions derived from the respective rated movies. This correlation was for emotions "happiness" and "surprise" weak negative, whereas a weak positive correlation was found with emotions "sadness", "anger", "disgust", and "fear". Overall, from all the categories of the 2nd broad section of tests a significant pattern was shown: the greatest number of tests reflecting a correlation among the variables tested were found when

lowest ratings were taken into account (tests 2.1.3 & 2.2.2). Hence, the user-centric section of tests showed that when a correlation among user preferences and emotions exists, then this comes from when users dislike what they watch. In other words, this probably could mean that when users enjoy a series of movies they watch, then the emotions evoked by the respective movies might not indicate a relationship among them. But when users do not like a set of movies, then the underlined emotions seem to present a correlation. As a consequence, RSs could use this information as followed: by understanding which movies users dislike (low ratings) and by investigating the respective emotions underneath those movies, then this would mean that those users do not enjoy watching movies with that set of particular emotions. This might not give a direct answer on what emotions users seek from movies or from which emotions users are attracted to, but indirectly it could: if users have voted a set of movies of e.g. the genre "action" with low ratings and if the underlying emotions of those particular set of action movies evoke "happiness", then does this mean that users do not enjoy action movies? Now that some emotional categories have been investigated as an extra feature in this project, probably no. Probably the user might indeed be in favour of action movies, but not for those action movies at where emotion "happiness" is evoked. For example, the user might prefer action movies in which emotions such as "fear" or "surprise" stand out.

6.3 Limitations

One of the limitations of this project lies in the fact that one of the researcher's rules during emotion labelling and emotion production was not strictly applied by the model. Specifically, there was a 2.5% error in terms of predictions sample's size (1,401 prediction cases) with regard to emotions "happiness" and "sadness", i.e. there were cases when the model predicted both those emotions for particular movies. The way the researcher fixed that error was by maintaining as a binary result of "1" only that emotion whose model's decision function confidence score was greater compared to the other emotion. This, at the same time, can be a limitation to be addressed as a future work, investigating the ways the above can be avoided before the model even starts predicting the emotions.

Another limitation was found when the researcher tried to implement[93] the FastBert library[94] (Trivedi, 2019), trying to exploit the potential advantages of deploying the BERT model, which is considered a benchmark in the field of DL for NLP tasks and is a work from Google's

[93] "4d_Model Bert.ipynb"
[94] https://github.com/kaushaltrivedi/fast-bert

research. Although unfamiliar with the Pytorch library, the researcher learned to work with it and finally run the BERT model for this project, but the results were unsatisfactory and the researcher proceeded with the rest of the models described in the paper.

6.4 Future Work

The project's work can be useful in movies industry in the context of film psychology. In the latest years, film psychologists have shown an interest in understanding the perception and cognition of a film by its viewers (Tan, 2018:p.15). This could help constructing movies categorizations based on genres and sub-genres, and emotion profiles with regard to movie synopses and styles.

A challenge, furthermore, would be the utilization of this project into the context of the recently built Emotion Aware Recommender Systems (EARSs) helping RSs to scale by recommending new items based on affective item features and users' emotional reactions (Mizgajski & Morzy, 2019:p.345). This could improve the similarity estimation between users, producing more personalized and targeted recommendations and, in addition, breaking the cold start problem in RSs.

This dissertation could also contribute to the improvement of folksonomy in the context of RSs which was described in the literature review. It could be achieved by giving more elements for more personalized content based on users' preferences and the flow of emotions in their watchlists.

Overall, the project's model could help creating a system that would automatically extract emotional tags from movie synopses, contributing to the problem of automatic movie profile generation (Kar *et al.*, 2018:p.7). By extension, the model could be applied in narrative texts of several fields, such as storylines of books and literature, video games, description of music playlists, and a wide range of multimedia content. Finally and as a subsequence, it could contribute to the improvement both of information seeking (user perspective) and information retrieval (system perspective) in the context of RSs and social tagging (Kar, Maharjan & Solorio, 2018:p.2879).

Another experiment the researcher suggests lies in the context of trying to reduce the amount of manual labelling in emotions: Sentiment analysis' polarity could be used to label emotions "happiness" and "sadness". If sentiment analysis indicates a positive polarity in movies plots, then the movie's emotion could be assigned with happiness, otherwise with sadness. Although, as described in the NLP sections, sentiment analysis does not identify notions of

emotions, however the above experiment could be implemented for those two particular emotions in the context of control vocabulary from the organization theories (Ishida, Shimizu & Yoshikawa, 2020), and in the context of trying to reduce the subjectivity encompassed from people's labelling, increasing hopefully in this way the validity of emotions' predictions.

Moreover, the csv files provided via a zip file could be used as data for various future work and experiments. Among the files provided quite useful could be deemed the file "movies_final.csv", which encompasses the fetching of 55,877 movie overviews from the TMDb and their harmonic merge with their metadata provided by MovieLens, as well as the "model_predictions_df.csv" which contains the predictions for 55,577 movies with their binary and decision function confidence scores.

Lastly, future experiments could display the notion and prediction of additional emotions (primary or secondary) as well the investigation of potential correlations among emotions and other movie metadata[95], such as movie genres and tags voted by users.

[95] This could also be implemented similarly to this project where the correlation among emotions and ratings was investigated.

Bibliography

Academy, K. (2020) *Counting, Permutations, and Combinations*. [Online]. 2020. www.khanacademy.org. Available from: https://www.khanacademy.org/math/statistics-probability/counting-permutations-and-combinations [Accessed: 10 August 2020].

Allison, B., Guthrie, D. & Guthrie, L. (2006) Another look at the data sparsity problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [Online] 4188 LNCS, 327–334. Available from: doi:10.1007/11846406_41.

Analytics, V. (2017) *Understanding Word Embeddings: From Word2Vec to Count Vectors*. [Online]. 2017. Analytics Vidhya. Available from: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/ [Accessed: 12 August 2020].

Bajpai, A. (2019) *Recurrent Neural Network & LSTM : Deep Learning for NLP*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/recurrent-neural-networks-deep-learning-for-nlp-37baa188aef5 [Accessed: 12 August 2020].

Biering, B. (2020) *Getting Started with AI: How Much Data do you Need?* [Online]. 2020. 2021.ai. Available from: https://2021.ai/getting-started-ai-how-much-data-needed/ [Accessed: 10 August 2020].

Bitbrain (2019) *What's the Difference between Feelings and Emotions?* [Online]. 2019. Bitbrain Technologies. Available from: https://www.bitbrain.com/blog/difference-feelings-emotions [Accessed: 7 June 2020].

Browniee, J. (2019a) *A Gentle Introduction to Normality Tests in Python*. [Online]. 2019. Machine Learning Mastery. Available from: https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/ [Accessed: 15 August 2020].

Browniee, J. (2019b) *How to Calculate Critical Values for Statistical Hypothesis Testing with Python*. [Online]. 2019. Machine Learning. Available from: https://machinelearningmastery.com/critical-values-for-statistical-hypothesis-testing/ [Accessed: 16 August 2020].

Browniee, J. (2019c) *How to Calculate Nonparametric Rank Correlation in Python*. [Online]. 2019. Machine Learning Mastery. Available from: https://machinelearningmastery.com/how-to-calculate-nonparametric-rank-correlation-in-python/ [Accessed: 15 August 2020].

Browniee, J. (2020a) *How to Encode Text Data for Machine Learning with scikit-learn*. [Online]. 2020. Machine Learning Mastery. Available from: https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/ [Accessed: 13 August 2020].

Browniee, J. (2020b) *How to Use the ColumnTransformer for Data Preparation*. [Online]. 2020. Machine Learning Mastery. Available from: https://machinelearningmastery.com/columntransformer-for-numerical-and-categorical-data/ [Accessed: 11 August 2020].

Cam-Stei, D. (2019) *Word Embedding Explained, a Comparison and Code Tutorial*. [Online]. 2019. Medium.com. Available from: https://medium.com/@dcameronsteinke/tf-idf-vs-word-embedding-a-comparison-and-code-tutorial-5ba341379ab0 [Accessed: 13 August 2020].

Chakraverty, S. & Saraswat, M. (2017) Review Based Emotion Profiles for Cross Domain Recommendation. *Multimedia Tools and Applications*. [Online] Available from: doi:10.1007/s11042-017-4767-x.

Chawla, R. (2017) *Topic Modeling with LDA and NMF on the ABC News Headlines Dataset*. [Online]. 2017. Medium.com. Available from: https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df [Accessed: 10 August 2020].

Choi, J.D., Tetreault, J. & Stent, A. (2015) It depends: Dependency Parser Comparison using a Web-based Evaluation Tool. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*. [Online] 1, 387–396. Available from: doi:10.3115/v1/p15-1038.

Crowston, K., Allen, E.E. & Heckman, R. (2012) Using Natural Language Processing Technology for Qualitative Data Analysis. *International Journal of Social Research Methodology*. [Online] Available from: doi:10.1080/13645579.2011.625764.

Dillard, J.P. & Meijnders, A. (2012) Persuasion and the Structure of Affect. *The Persuasion Handbook: Developments in Theory and Practice*. [Online] 27 (1), 309–328. Available from: doi:10.4135/9781412976046.n16.

Donges, N. (2018) *Data Types in Statistics*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/data-types-in-statistics-347e152e8bee [Accessed: 15 August 2020].

Dwivedi, P. (2018) *NLP: Extracting the Main Topics from your Dataset using LDA in minutes*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/nlp-extracting-the-main-topics-from-your-dataset-using-lda-in-minutes-21486f5aa925 [Accessed: 10 August 2020].

Edgell, J. (2016) *Permutations and Combinations*. 2nd ed. USA, Farmington Hills, MI: Macmillan.

Ekman, P. (1992) *An Argument for Basic Emotions*. [Online] 6(3-4), 169–200. Available from: doi:10.1080/02699939208411068.

Farzindar, A. & Inkpen, D. (2015) Natural Language Processing for Social Media. *Synthesis Lectures on Human Language Technologies*. [Online]. Available from: doi:10.2200/S00659ED1V01Y201508HLT030.

Favaretto, R.M., Musse, S.R. & Costa, A.B. (2019) *Emotion, Personality and Cultural Aspects in Crowds [internet Resource] : Towards a Geometrical Mind*. 1st. Ed. Springer International Publishing.

Font, F., Serrà, J. & Serra, X. (2013) Folksonomy-Based Tag Recommendation for

Collaborative Tagging Systems. *International Journal on Semantic Web and Information Systems*. [Online] 9 (2), 1–30. Available from: doi:10.4018/jswis.2013040101.

Galarnyk, M. (2018) *Understanding Boxplots*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51 [Accessed: 15 August 2020].

Geographyfieldwork (2020) *Spearman's Rank Calculator: Rs, p-value, Scatter Graph and Conclusion*. [Online]. 2020. Barcelona Field Studies Centre S.L. Available from: https://geographyfieldwork.com/SpearmansRankCalculator.html [Accessed: 15 August 2020].

Géron, A. (2019) *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd Ed. Sebastopol, CA : O'Reilly Media, Inc.

Google Colaboratory (n.d.) *Emotion Classification using Fine-tuned BERT model*. [Online]. Available from: https://colab.research.google.com/drive/1nwCE6b9PXIKhv2hvbqf1oZKIGkXMTi1X#scrollTo=pSzoz9InH0Ta [Accessed: 10 August 2020].

Gouws, S. (2017) *How is GloVe Different from Word2Vec?* [Online]. 2017. Deeplearning.lipingyang.org. Available from: http://clic.cimec.unitn.it/marco... [Accessed: 12 August 2020].

Gross, J.J. & Levenson, R.W. (1995) Emotion Elicitation Using Films. *Cognition and Emotion*. [Online] 9 (1), 87–108. Available from: doi:10.1080/02699939508408966.

GroupLens/MovieLens (2020) *MovieLens Documentation - ml-25m*. [Online]. 2020. MovieLens. Available from: http://files.grouplens.org/datasets/movielens/ml-25m-README.html [Accessed: 8 August 2020].

GroupLens (2020) *What is GroupLens? | GroupLens*. [Online]. 2020. Grouplens.org. Available from: https://grouplens.org/about/what-is-grouplens/ [Accessed: 17 April 2020].

Hale, J. (2019) *Scale, Standardize, or Normalize with Scikit-Learn*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02 [Accessed: 11 August 2020].

Hand, D.J. & Till, R.J. (2001) A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*. [Online] Available from: doi:10.1023/A:1010920819831.

Harper, F.M. & Konstan, J.A. (2015a) The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*. [Online] 5 (4). Available from: doi:10.1145/2827872.

Harper, F.M. & Konstan, J.A. (2015b) The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*. [Online] 5 (4). Available from: doi:10.1145/2827872.

Health, I. for W.& (2015) Primary Data and Secondary Data. *At Work Issue 54*. [Online] 1–2. Available from: https://www.iwh.on.ca/what-researchers-mean-by/primary-data-and-

secondary-data [Accessed: 8 August 2020].

Hirschberg, J. & Manning, C.D. (2015) Advances in Natural Language Processing. *Science*. [Online]. Available from: doi:10.1126/science.aaa8685.

Honold, A. (2020) *Using ColumnTransformer to Combine Data Processing Steps*. [Online]. 2020. Towards Data Science. Available from: https://towardsdatascience.com/using-columntransformer-to-combine-data-processing-steps-af383f7d5260 [Accessed: 13 August 2020].

Hutto, C.J. & Gilbert, E.E. (2014) VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14).". *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.

Innes-Ker (2015) Film and Music in Laboratory Experiments: Emotion Induction. *Music and the Moving Image*. [Online] 8 (2), 58. Available from: doi:10.5406/musimoviimag.8.2.0058.

Invidi (2018) *INVIDI Addressable Advertising - How It Works*. [Online]. 2018. Youtube.com. Available from: https://www.youtube.com/watch?v=YRgse4ncM3A [Accessed: 25 May 2020].

Ishida, Y., Shimizu, T. & Yoshikawa, M. (2020) An Analysis and Comparison of Keyword Recommendation Methods for Scientific Data. *International Journal on Digital Libraries*. [Online] Available from: doi:10.1007/s00799-020-00279-3.

Jagota, A. (2020) *Topic Modeling In NLP, With a focus on Latent Dirichlet Allocation*. [Online]. 2020. Towards Data Science. Available from: https://towardsdatascience.com/topic-modeling-in-nlp-524b4cffeb68 [Accessed: 10 August 2020].

Jain, S. (2017) *Solving Multi Label Classification Problems (Case studies included)*. [Online]. 2017. Analytics Vidhya. Available from: https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/ [Accessed: 13 August 2020].

Kaggle (2017) *TMDB 5000 Movie Dataset | Kaggle*. [Online]. 2017. Kaggle.com. Available from: https://www.kaggle.com/tmdb/tmdb-movie-metadata [Accessed: 17 April 2020].

Kar, S., Maharian, S. & Solorio, T. (2018) *Folksonomication: Predicting Tags for Movies from Plot Synopses Using Emotion Flow Encoded Neural Network*. [Online]. 2018. Association for Computational Linguistics. Available from: http://ritual.uh.edu/folksonomication-2018/ [Accessed: 17 April 2020].

Kar, S., Maharjan, S., Pastor López-Monroy, A. & Solorio, T. (2018) MPST: A Corpus of Movie Plot Synopses with Tags. In: *LREC 2018 - 11th International Conference on Language Resources and Evaluation*. 2018 p.

Kar, S., Maharjan, S. & Solorio, T. (2018) *Folksonomication: Predicting Tags for Movies from Plot Synopses using Emotion Flow encoded Neural Network*. [Online]. 2018. RiTUAL. Available from: http://ritual.uh.edu/folksonomication-2018/ [Accessed: 9 May 2020].

Karani, D. (2018) *Introduction to Word Embedding and Word2Vec*. [Online]. 2018. Towards

Data Scienceards Data Science. Available from: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa [Accessed: 12 August 2020].

Korstanje, J. (2019) *6 Ways to Test for a Normal Distribution — Which One to Use?* [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/6-ways-to-test-for-a-normal-distribution-which-one-to-use-9dcf47d8fa93 [Accessed: 15 August 2020].

Laerd (2020) *Pearson Product-Moment Correlation*. [Online]. 2020. Laerd Statistics. Available from: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php [Accessed: 15 August 2020].

Lazemi, S. & Ebrahimpour-Komleh, H. (2016) Improving Collaborative Recommender Systems via Emotional Features. In: *Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings*. [Online]. 2016 p. Available from: doi:10.1109/ICAICT.2016.7991703.

Lewinson, E. (2019) *Violin plots explained. Learn how to use violin plots and what are their advantages over box plots!* [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/violin-plots-explained-fb1d115e023d [Accessed: 15 August 2020].

Li, J. (2017) *Keras Models: Sequential vs. Functional*. [Online]. 2017. https://jovianlin.io/. Available from: https://jovianlin.io/keras-models-sequential-vs-functional/ [Accessed: 12 August 2020].

Li, S. (2018a) *Multi Label Text Classification with Scikit-Learn*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/multi-label-text-classification-with-scikit-learn-30714b7819c5 [Accessed: 13 August 2020].

Li, S. (2018b) *Named Entity Recognition with NLTK and SpaCy | by Susan Li |*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da [Accessed: 10 August 2020].

Liu, D., Li, Y. & Thomas, M.A. (2017) A Roadmap for Natural Language Processing Research in Information Systems. In: *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*. [Online]. 2017 p. Available from: doi:10.24251/hicss.2017.132.

Lops, P., Jannach, D., Cataldo, M. & Bogers, T. (2019) Trends in content-based recommendation. *Springer Nature*.

Loukas, S. (2020) *ROC Curve explained using a COVID-19 hypothetical example: Binary & Multi-Class Classification tutorial*. [Online]. 2020. Towards Data Science. Available from: https://towardsdatascience.com/roc-curve-explained-using-a-covid-19-hypothetical-example-binary-multi-class-classification-bab188ea869c [Accessed: 13 August 2020].

Magid (2019) A Briefing on the Role of Emotion in Driving Viewership Trends. *Magid*. 1–8.

Magid (2020) *Emotional DNA*. [Online]. 2020. Magid. Available from: https://magid.com/emotional-dna/ [Accessed: 19 June 2020].

Maklin, C. (2019) *TF IDF | TFIDF Python Example*. [Online]. 2019. Towards Data Science.

Available from: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76 [Accessed: 13 August 2020].

Malik (2020) *Python for NLP: Creating Multi-Data-Type Classification Models with Keras*. [Online]. 2020. Stack Abuse. Available from: https://stackabuse.com/python-for-nlp-creating-multi-data-type-classification-models-with-keras/ [Accessed: 12 August 2020].

Malik, U. (2019a) *Python for NLP: Multi-label Text Classification with Keras*. [Online]. 2019. Stack Abuse. Available from: https://stackabuse.com/python-for-nlp-multi-label-text-classification-with-keras/ [Accessed: 12 August 2020].

Malik, U. (2019b) *Python for NLP: Word Embeddings for Deep Learning in Keras*. [Online]. 2019. Stack Abuse. Available from: https://stackabuse.com/python-for-nlp-word-embeddings-for-deep-learning-in-keras/ [Accessed: 12 August 2020].

Marshall, C. (2019) *What is named entity recognition (NER) and how can I use it?* [Online]. 2019. Medium.com. Available from: https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d [Accessed: 10 August 2020].

Mauss, I.B. & Robinson, M.D. (2009) Measures of emotion: A review. *Cognition and Emotion*. [Online] 23 (2), 209–237. Available from: doi:10.1080/02699930802204677.

Maynard, D., Bontcheva, K. & Augenstein, I. (2017) Natural Language Processing for the Semantic Web. *Synthesis Lectures on the Semantic Web: Theory and Technology*. [Online]. Available from: doi:10.2200/S00741ED1V01Y201611WBE015.

McCombes, S. (2020) *Understanding Types of Research*. [Online]. 2020. Scribbr. Available from: https://www.scribbr.com/methodology/types-of-research/ [Accessed: 17 April 2020].

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., et al. (2013) Distributed representations ofwords and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*. 2013 p.

Mizgajski, J. & Morzy, M. (2019) Affective Recommender Systems in Online News Industry: How Emotions Influence Reading Choices. *User Modeling and User-Adapted Interaction*. [Online] 29 (2), 345–379. Available from: doi:10.1007/s11257-018-9213-x.

MovieLens (2020) *MovieLens | GroupLens*. [Online]. 2020. GroupLens. Available from: https://grouplens.org/datasets/movielens/ [Accessed: 17 April 2020].

Nabi, R.L. (2010) The Case for Emphasizing Discrete Emotions in Communication Research. *Communication Monographs*. [Online] 77 (2), 153–159. Available from: doi:10.1080/03637751003790444.

Nag, A. (2019) *Understanding Multi-Label Classification Model and Accuracy Metrics*. [Online]. 2019. Medium.com | Towards AI — Multidisciplinary Science Journal. Available from: https://medium.com/towards-artificial-intelligence/understanding-multi-label-classification-model-and-accuracy-metrics-1b2a8e2648ca [Accessed: 13 August 2020].

Narkhede, S. (2018) *Understanding AUC - ROC Curve*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/understanding-auc-roc-curve-

68b2303cc9c5 [Accessed: 13 August 2020].

Nooney, K. (2018) *Deep Dive into Multi-label Classification (With detailed Case Study)*. [Online]. 2018. Towards Data Science. Available from: https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff [Accessed: 13 August 2020].

Oatley, K. (1992) Basic Emotions: Theory and Measurement. *Cognition and Emotion*. [Online] 6 (3–4), 161–168. Available from: doi:10.1080/02699939208411067.

Odić, A., Tkalčič, M., Tasič, J.F. & Košir, A. (2013) Predicting and Detecting the Relevant Contextual Information in a Movie-recommender System. *Interacting with Computers*. [Online] 25 (1), 74–90. Available from: doi:10.1093/iwc/iws003.

Okada, S. (2019) *Spearman's Rank Correlation Coefficient Using Ordinal Data*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/discover-the-strength-of-monotonic-relation-850d11f72046 [Accessed: 15 August 2020].

Pathak, M. (2020) *Handling Categorical Data in Python* -. [Online]. 2020. DataCamp. Available from: https://www.datacamp.com/community/tutorials/categorical-data [Accessed: 11 August 2020].

Peltarion (2020) *Micro F1-score*. [Online]. 2020. https://peltarion.com/. Available from: https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/micro-f1-score [Accessed: 13 August 2020].

Pennington, J., Socher, R. & Manning, C.D. (2014) GloVe: Global vectors for word representation. In: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. [Online]. 2014 p. Available from: doi:10.3115/v1/d14-1162.

Plutchik, R. (2001) The Nature of Emotions. *American Scientist*. 89 (4), 344–350.

Portilla, J. (2019) *NLP - Natural Language Processing with Python*. [Online]. Available from: https://www.udemy.com/course/nlp-natural-language-processing-with-python/.

Riedl, M. & Biemann, C. (2017) There's no 'Count or Predict' but task-based selection for distributional models. In: *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017)*. 2017 p.

Sahu, A.K., Dwivedi, P. & Kant, V. (2018) Tags and Item Features as a Bridge for Cross-Domain Recommender Systems. In: *Procedia Computer Science*. [Online]. 2018 p. Available from: doi:10.1016/j.procs.2017.12.080.

Salgado, R. (2016) *Topic Modeling Articles with NMF*. [Online]. 2016. Towards Data Science. Available from: https://towardsdatascience.com/topic-modeling-articles-with-nmf-8c6b2a227a45 [Accessed: 10 August 2020].

Santhanam, N. (2019) *Explain your Machine Learning with Feature Importance*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/explain-your-machine-learning-with-feature-importance-774cd72abe [Accessed: 11 August 2020].

SAS Institute Inc. (2019) Make Every Voice Heard with Natural Language Processing. *SAS Visual Text Analytics*.

Sawarn, A. (2019) *Keras for Multi-label Text Classification*. [Online]. 2019. Medium.com | Towards AI — Multidisciplinary Science Journal. Available from: https://medium.com/towards-artificial-intelligence/keras-for-multi-label-text-classification-86d194311d0e [Accessed: 12 August 2020].

Schulz, R. (2018) *Performing Multi-label Text Classification with Keras |*. [Online]. 2018. Mima.com. Available from: https://blog.mimacom.com/text-classification/ [Accessed: 12 August 2020].

Scikit-Learn (2020a) *sklearn.feature_extraction.text.CountVectorizer*. [Online]. 2020. Scikit-Learn 0.23.2 documentation. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html [Accessed: 13 August 2020].

Scikit-Learn (2020b) *sklearn.metrics.f1_score — scikit-learn 0.23.2 documentation*. [Online]. 2020. scikit-learn.org. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html [Accessed: 13 August 2020].

Scikit-Learn (2020c) *sklearn.metrics.hamming_loss — scikit-learn 0.23.2 documentation*. [Online]. 2020. https://scikit-learn.org/. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html [Accessed: 13 August 2020].

Scikit-Learn (2020d) *sklearn.metrics.multilabel_confusion_matrix — scikit-learn 0.23.2 documentation*. [Online]. 2020. scikit-learn.org. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.multilabel_confusion_matrix.html [Accessed: 13 August 2020].

Scikit-Learn (2020e) *sklearn.metrics.roc_auc_score — scikit-learn 0.23.2 documentation*. [Online]. 2020. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html [Accessed: 13 August 2020].

Scikit-Learn (2020f) *sklearn.svm.LinearSVC — scikit-learn 0.23.2 documentation*. [Online]. 2020. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html [Accessed: 11 August 2020].

Shantikumar, S. (2016) *Parametric and Non-parametric tests for comparing two or more groups |*. [Online]. 2016. Health Knowledge | Campbell MJ. Available from: https://www.healthknowledge.org.uk/public-health-textbook/research-methods/1b-statistical-methods/parametric-nonparametric-tests [Accessed: 15 August 2020].

Shapiro, S., MacInnis, D.J. & Park, C.W. (2002) Understanding Program-Induced Mood Effects: Decoupling Arousal from Valence. *Journal of Advertising*. [Online] 31 (4), 14–26. Available from: doi:10.1080/00913367.2002.10673682.

Shelar, H., Kaur, G., Heda, N. & Agrawal, P. (2020) Named Entity Recognition Approaches

and Their Comparison for Custom NER Model. *Science and Technology Libraries*. [Online] 00 (00), 1–14. Available from: doi:10.1080/0194262X.2020.1759479.

Srinidhi, S. (2019) *Fit vs. Transform in SciKit libraries for Machine Learning*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/fit-vs-transform-in-scikit-libraries-for-machine-learning-3c70e6300ded [Accessed: 11 August 2020].

Srivastava, T. (2019) *NLP: A Quick Guide to Stemming*. [Online]. 2019. Medium.com. Available from: https://medium.com/@tusharsri/nlp-a-quick-guide-to-stemming-60f1ca5db49e [Accessed: 11 August 2020].

Statistics, S.S. (2020) *Spearman's Rho Calculator (Correlation Coefficient)*. [Online]. 2020. Socscistatistics.com. Available from: https://www.socscistatistics.com/tests/spearman/default.aspx [Accessed: 15 August 2020].

Statstutor (2020) *Search Statstutor*. [Online]. 2020. Statstutor.ac.uk. Available from: https://www.statstutor.ac.uk/search/?q=spearman&saudience=all&cat%5B%5D=5&cat%5B%5D=9&cat%5B%5D=3&cat%5B%5D=6&cat%5B%5D=4&cat%5B%5D=11&cat%5B%5D=13&cat%5B%5D=14&cat%5B%5D=15&cat%5B%5D=16&cat%5B%5D=18 [Accessed: 15 August 2020].

Stevens, K., Kegelmeyer, P., Andrzejewski, D. & Buttler, D. (2012) Exploring Topic Coherence over many models and many topics. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*. (July), 952–961.

Surbhi, S. (2016) *Difference Between Primary and Secondary Data*. [Online]. 2016. Key Differences. Available from: https://keydifferences.com/difference-between-primary-and-secondary-data.html [Accessed: 8 August 2020].

Szymánski, P. & Kajdanowicz, T. (2019) Scikit-multilearn: A scikit-based Python environment for performing multi-label classification. *Journal of Machine Learning Research*. [Online] Available from: doi:10.5281/zenodo.3670933.

Tan, E.S. (2018) A psychology of the Film. *Palgrave Communications*. [Online] 4 (1). Available from: doi:10.1057/s41599-018-0111-y.

Tan, T. (2019) *Nuances in the Usage of Word Embeddings: Semantic and Syntactic Relationships*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/nuances-in-the-usage-of-word-embeddings-semantic-and-syntactic-relationships-780940fe28f [Accessed: 14 August 2020].

Tapidze, G. (2017) *The Power of Big Data and Psychographics in the Electoral Process*. [Online]. 2017. Slideshare.net. Available from: https://www.slideshare.net/GMline/the-power-of-big-data-and-psychographics-in-the-electoral-process [Accessed: 27 May 2020].

Teng, A. (2019) *Dealing with Multiclass Data*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/dealing-with-multiclass-data-78a1a27c5dcc [Accessed: 13 August 2020].

Terence, S. (2020) *Understanding the Confusion Matrix and How to Implement it in Python*. [Online]. 2020. Towards Data Science. Available from: https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d [Accessed: 13 August 2020].

Tkalčič, M., Tan, D., Vanderdonckt, J., De Carolis, B., et al. (2016) *Emotions and Personality in Personalized Services: Models, Evaluation and Applications*. 1st ed. [Online]. Switzerland, Springer International Publishing. Available from: doi:10.1007/978-3-319-31413-6.

Topal, K., Koyutürk, M. & Özsoyoğlu, G. (2017) Effects of Emotion and Topic Area on Topic Shifts in Social Media Discussions. *Social Network Analysis and Mining*. [Online] Available from: doi:10.1007/s13278-017-0465-y.

Toshniwal, R. (2020) *Demystifying ROC Curves - How to Interpret and when to Use Receiver Operating Characteristic Curves*. [Online]. 2020. Towards Data Science. Available from: https://towardsdatascience.com/demystifying-roc-curves-df809474529a [Accessed: 13 August 2020].

Trivedi, K. (2019) *Introducing FastBert - A simple Deep Learning library for BERT Models*. [Online]. 2019. Medium. Available from: https://medium.com/huggingface/introducing-fastbert-a-simple-deep-learning-library-for-bert-models-89ff763ad384 [Accessed: 16 August 2020].

Venkatesan, R. & Joo Er, M. (2014) Multi-label classification method based on extreme learning machines. *IEEE - 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. [Online] 619–624. Available from: doi:10.1109/ICARCV.2014.7064375.

Vig, J., Sen, S. & Riedl, J. (2012) The Tag Genome: Encoding Community Knowledge to Support Novel Interaction. *ACM Transactions on Interactive Intelligent Systems*. [Online] 2 (3). Available from: doi:10.1145/2362394.2362395.

Vlad, D.E. (2020) *Concepts of Quality Connected to Social Media and Emotions*. 1st edition. [Online]. Available from: https://doi.org/10.1007/978-3-658-28867-9.

Vu, D. (2019) *(Tutorial) Generate Word Clouds in Python*. [Online]. 2019. DataCamp. Available from: https://www.datacamp.com/community/tutorials/wordcloud-python [Accessed: 14 August 2020].

Weisler, C. (2018) *ABC and Accenture Strategy Discover the Secret of Sales ROI MediaVillage*. [Online]. 2018. Media Insights. Available from: https://www.mediavillage.com/article/abc-and-accenture-strategy-discover-the-secret-of-sales-roi/?utm_campaign=nl-daily&utm_medium=email&utm_source=ABC+and+Accenture+Strategy+Discover+the+Secret+of+Sales+ROI [Accessed: 19 June 2020].

Willems, K. (2019) *KERAS Tutorial: Deep Learning in Python*. [Online]. 2019. Datacamp.com. Available from: https://www.datacamp.com/community/tutorials/deep-learning-python [Accessed: 12 August 2020].

Wirth, W. & Schramm, H. (2005) Media and Emotions. *Communication Research Trends*. 24

(3).

Yiu, T. (2019) *The Curse of Dimensionality - Why High Dimensionality Data Can Be So Troublesome*. [Online]. 2019. Towards Data Science. Available from: https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e [Accessed: 12 August 2020].

Zablocki, M. (2020a) *Custom Classifier on top of Transformers Language Models - Example with PolEmo2.0 Sentiment Classification*. [Online]. 2020. Google Colaboratory. Available from: https://colab.research.google.com/drive/1sajgpLTrTJDzRSlxycy8aE6ysxGqaT18 [Accessed: 10 August 2020].

Zablocki, M. (2020b) *Marcin Zabłocki blog | Custom classifier on top of BERT-like Language Model - guide*. [Online]. 2020. Zablo.net. Available from: https://zablo.net/blog/post/custom-classifier-on-bert-model-guide-polemo2-sentiment-analysis/ [Accessed: 10 August 2020].

Zablocki, M. (2020c) *Transformers-Sentiment-Analysis: Custom Classifier on top of Transformers Language Models*. [Online]. 2020. Github.com. Available from: https://github.com/marrrcin/transformers-sentiment-analysis [Accessed: 10 August 2020].