Sentiment Analysis of Movie Reviews by Merging Comments from two Well-Known Platforms

Georgios Liachoudis
STUDENT NUMBER: 2035829

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

dr. Henry Brighton
prof. dr. Max Louwerse

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
June 2020

Preface

After a year and a half of hard work, including many successes and some failures, I completed my scientific research, which was a new experience for me, as it was the first thesis project in my academic history. First, I want to show appreciation to my supervisor Henry Brighton for his helpful feedback and dedication. In addition, I want to thank my parents and my good friends from the Netherlands and Greece as well, who encouraged me psychologically to deal with such a difficult area of study. This work has been done under extraordinary circumstances, considering the burdensome consequences of the coronavirus on our lives. For the readers, I hope that my process will help them to extract some useful insights and think of further improvements in the field of Sentiment Analysis.

Sentiment Analysis of Movie Reviews by Merging Comments from two Well-Known Platforms

Georgios Liachoudis

Sentiment Analysis is a specific category in Natural Language Processing (NLP), also known as Opinion Mining, a mechanism which converts text sources into a measure that reflects people's feelings. The key element in this procedure is the replacement of traditional, human-based supervision, by an automated decision-making arrangement capable of forecasting emotions. This method provides fundamental insights into the field of web-based application systems that contain comments to be translated as guidance to aid user choice. Significant work has been carried out in the area of identifying opinions from given text data on an individual web-platform. This suggested a further step, applying Sentiment Analysis and employing data from two movie recommendation systems to investigate to what extent merging sources improves on the use of an individual source. This idea underpins the research goal of this thesis and gave birth to the title. In this study, use is made of existing extracted data provided by IMDB and Rotten Tomatoes websites. In addition, target columns having been labeled relevantly, so that the requirement of supervised learning is fulfilled. A number of Machine Learning classifiers have been employed: Logistic Regression, Support Vector Machines (SVM), Multinomial Naive Bayes, Extreme Gradient Boosting (XGBoost), Random Forest, K-Nearest Neighbors (K-NN), all included in both the Bag of Words (BoW) and the Term Frequency - Inverse Document Frequency (TF-IDF) representations, but also some Deep Learning algorithms have been used: Word Embedding, Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (Bid-LSTM). All models were exploited for the merged and the original sets. The conclusion of this thesis is that combined datasets provide equal or nearly equal higher performance, accuracy and relevance than those provided by individual and separated sets.

1 Introduction

1.1 Context

The main target of Natural Language Processing (NLP) is to develop formulas to process text using computers in order to analyze, extract and finally to express information variously. This procedure relates various categories to parts of the text (Sentiment Analysis), changes the format of text (parsing), or converts it to other patterns (translation, summarization) (Conneau et al. 2016). This thesis focuses on the first approach, Sentiment Analysis, which is a function for measuring favorable versus unfavorable feelings, emotions and opinions, employing movie reviews (Mesnil et al. 2014). More specifically, Sentiment Analysis skills were applied to already extracted movie reviews from two well-known platforms, IMDB and Rotten Tomatoes. The goal of the project was to identify miscellaneous Machine and Deep Learning techniques to predict the

sentiments projected in each individual system, but in addition, to investigate how accurately each algorithm would classify emotions when reviews were from both sources. The previous scientific literature suggests that various techniques and newly clarified procedures were introduced in the field of classifying text data sources, in order to estimate sentiments and their relationship to human behavior. This measure of determining feelings was achieved either in the area of supervised learning, by classifying targets into two groups, as in this case, but also into multi-class representations; or in the sphere of unsupervised learning by generating clusters from scratch. This research yielded a supplementary direction for accounting for the reactions of users, when reviews from two recommendation systems are connected or merged. The impact of this approach can offer more sophisticated technical outcomes, in order to make comparisons between the performance of each individual platform and those gained after merging both sources. Further, an additional societal influence was observed, that may motivate a movie user to take into account the comments of movie pundits and to what extent the responses of the system, divided into positive and negative outcomes, guides the user's predisposition and decision-making regarding movie selection. More particularly, the introduction of this path leads to useful conclusions regarding the level of reliability that both recommendation systems provide for their users. Both impacts highlight the project's importance and relevance, arising from its core goal of determining which of the two web systems governs the joined dataset with respect to the accuracy of the metrics.

1.2 Research questions

Having briefly explained the main objective of this thesis, a further indication of the strategy to be pursued in terms of research questions shows the depth of the task. Firstly, how accurate the results of the thesis are, depends on the accuracy of the algorithms being applied. Secondly, all programming techniques will be exploited in each prior dataset and also in the merged ones. Thus, we can formulate our research questions as follows:

RQ1. How does the large set to be developed perform against the partial ones?

RQ2. Is a Bag of Words (BoW) model more accurate than a Term Frequency - Inverse Document Frequency (TF-IDF) model, regarding this attempt in the field of Machine Learning binary representation?

RQ3. To what extent is a Deep Learning classifier preferable to a Machine Learning one?

1.3 Findings

The main findings can be considered as outcomes that emphasize the research questions specified previously. Starting with the major goal of the thesis, the algorithms applied in IMDB's dataset provided greater accuracy in comparison to the Rotten Tomatoes' dataset, in almost all situations. Also, the merged set seems to act in accordance with the most accurate part, in this case the IMDB set. A definite rule could not be established whether a Bag of Words (BoW) or a Term Frequency - Inverse Document Frequency (TF-IDF) model performs better, because estimations have showed that the selection varies among different classification concepts. Lastly, a similar conclusion is drawn regarding the last scientific task. It is clear, that a definite conclusion could not be made whether a Machine Learning classifier is more accurate than a Deep Learning classifier or not.

An accurate conclusion may only be reached if each classification scheme is taken into account separately.

2 Related Work

2.1 Sentiment Analysis

Sentiment Analysis can be implemented in disparate ways, starting from the primary requirement, (knowing or not the values in the target column), which is correlated with the text data someone desires to work with, if it already exists, in terms of a specific format, or, must be extracted from a web system, given that it is granted as an Application Programming Inference (API). Furthermore, the next key element is that this mode of Natural Language Processing (NLP) can be applied in three cases: supervised, semi-supervised and unsupervised learning. This kind of analysis is applicable in interdisciplinary tasks and yields to beneficial solutions. In this project, extracted data is already available, the target is known and exists in labeled columns, meaning that all the implementations are supervised. Finally, in terms of the movie reviews, in order to improve previous research, by introducing the connection between two web platforms is introduced. This leads into important and interesting outcomes, related to reliability in both a technical and communal view i.e its reliability to users. The result should increase user trustworthiness of IMDB and Rotten Tomatoes, regarding each system's practice of separating comments into two groups, positive and negative, but it might conclude in upgrading of industrial results in the manner of binary classification problems, using text data.

2.2 Previous work on Sentiment Analysis of movie reviews

Document classification in the sphere of movie reviews by both IMDB and Rotten Tomatoes, but also other sources, has been commonly used in order to examine people's sentiments, as the literature has indicated, also suggesting alternative steps to enhance them.

The first paper that underlines the previous work in the scientific area was authored by Baid, Gupta, and Chaplot (2017). The writers analyzed movie reviews from IMDB including two columns, the first containing the text sources and the other the target one, divided into "positive" versus "negative" emotions. The dataset was imported through software called "WEKA", using "Text Directory Loader" and an Exploratory Data Analysis (EDA), which suggests that it was completely balanced regarding the mentioned sub-classes. Moreover, "StringToWordVector" and "AttributeSelection" filters were applied as basic methods during the pre-processing phase. The last stage was connected with the training algorithms exploited to deal with the binary problem. Three Machine Learning classifiers were employed, Naive Bayes, K-Nearest Neighbors (K-NN) and Random Forest, both for training and evaluating the performance. Naive Bayes achieved 81.4%, which was the highest rate, Random Forest classifier was in the second position with a 78.65% and at the bottom was the K-Nearest Neighbors (K-NN) method, using k=3, gaining only a 55.3%, in terms of the accuracy metric. 10-fold cross-validation was used for the validation set and for hyper-parameter tuning. The authors concluded that further improvement in performances might be possible, hence the motivation to do so in our analogous scientific cases.

Pouransari and Ghili (2014) apply two different concepts to Sentiment Analysis. The first one, deals with the binary representation problem by employing Machine Learning

techniques, whereas the second, is a multinomial illustration using a Recursive Neural Network (RNN). In this research, only the first approach was considered, regarding our binary representation tasks. The key issue that strengthens the possibility of successfully adding a scientific value, is that the dataset taken from IMDB is similar. A small difference appears because the authors implemented a clustering method on an extra unlabeled set, something that is irrelevant to our procedure. The labeled dataset was partitioned into equal numbers of "positive" and "negative" feelings, then it was cleaned appropriately by removing stop-words, punctuation marks and finally a tokenization method was applied. Furthermore, the text sources were transferred into numerical vectors by introducing the Bag of Words (BoW) model, one of the most accessible methods. The implementation of Bag of Words (BoW) created models that were trained and tested under the Random Forest classifier, but also the Support Vectors Machines (SVM) and the Logistic Regression algorithms. The first process produced an accuracy of an 84.0%, in contrast to the other two processes that returned 85.8% and 86.6% respectively. "L2-regularization" was applied as part of the validation set, to speed up the performances of Support Vectors Machine (SVM) and Logistic Regression.

In another paper Palkar et al. (2016) considers issues comparable to this research, using movie reviews from the same web platform. The IMDB provider appears again during this process, containing text data with labeled target columns available for binary representation supervised learning algorithms. This scientific work complements this work regarding the fact that various classifiers and techniques were trained in the area of Machine Learning, such as Naïve Bayes, Support Vector Machines (SVM), Maximum Entropy, Classification and Regression Trees, and Random Forest. A classic pre-processing step was taken here, during which the documents were cleaned by removing numbers, stop-words, punctuation marks, white spaces, sparse terms and the words were also stemmed. Term Frequency - Inverse Document Frequency (TF-IDF) illustration was exploited in order to be ready for the final stage of training the classifiers. The experimental results demonstrated competitive performances between the different methods, a fact that motivates the search for further improvement. Lastly, the paper stimulates readers to go a step further by building some Deep models for future comparisons, a suggestion equal to our expectations that were successful during this process, by applying it to alternative data sources.

Further research worth noting is that of Dridi and Recupero (2017), that shows a close correlation with one of our research questions, in terms of identifying relations between a Bag of Words (BoW) and a Term Frequency - Inverse Document Frequency (TF-IDF) model, that examines the results of Opinion Mining. Once more, the same IMDB's dataset was used by training the Multinomial Naive Bayes classifier, which is also included in his research, using numerical vectors obtained from the previously mentioned models. Before those implementations, a usual pre-processing stage was included, in order to clean the documents before the "n_gram" method was applied to decide the number of words during the tokenization part. Also, two approaches that highlight the role of semantics in Sentiment Analysis of movie reviews, called "frame semantics" and "lexical resources", were applied in to achieve better performances. Dridi and Recupero's paper concludes that further work could be done by employing these ideas, such as introducing the Rotten Tomatoes set to provide wider scientific insights.

In the same direction, one more piece of related work of text mining is by Dey et al. (2016). This research struggled to make worthwhile comparisons between a dataset available from IMDB and reviews contributed by a hotel. Both sets contained target columns already labeled, divided into "positive" vs "negative" emotions which were trained under two supervised classifiers, the K-Nearest Neighbors (K-NN) on the one

hand and the Naive Bayes on the other. The fundamental difference in comparison to previous efforts was that the above algorithms were run various times, changing the amount of reviews during each process, depending on their importance. That was accomplished by applying chi-squared tests to identify the top ones only. Moreover, besides the important issue of accuracy, other evaluation metrics were taken into consideration, such as Precision and Recall. The results have shown that the Naive Bayes classifiers obtained better outcomes when they referred to the IMDB situation, in contrast to the hotel reviews domain, in which K-Nearest Neighbors (K-NN) and Naive Bayes gave equally poorer performances. This was a motivation for the current research, suggesting exploiting other Machine Learning algorithms which would produce more accurate results. From the point of view of this research, the question of merging text data from distinct sources remains unanswered, hence the inspiration for this research to merge reviews from different web sites.

The necessity of presenting a Deep Learning view of Sentiment Analysis requires citation of the Desai et al. (2018) paper. In this case, the writers used text data from the Rotten Tomatoes' recommendation system and the SAR14 dataset, but with a small difference, which lies in the fact that the target columns, in both sets, were divided into more than two groups. This did not introduce an obstacle to this project, as a main aim was to extract knowledge for our Deep models' architectures. Regarding the Deep Learning tasks, a Sequential model was developed by implementing an Embedding layer as input and the hybrid architecture linked the Convolutional Neural Network (CNN) either with a Recurrent Neural Network (RNN) or a Long Short-Term Memory (LSTM) one, for the training purposes. It was clear from the results that the Deep process produced higher accuracy than the common Machine Learning modes in the past. In addition, the Long Short-Term Memory (LSTM) was preferred against the Recurrent Neural Network (RNN), because it is a more complex and effective Recurrent Neural Network (RNN), which is overcome with the vanishing gradient problem, hence it is capable of learning long term dependencies. It should be noted that the three layers mentioned suggested what mechanisms might used in our Deep Learning tasks. In the same manner, two scientific jobs were taken into consideration in the process of this research, in order to understand the terminology of a Bidirectional Long Sort-Term Memory (Bid-LSTM) model in Baziotis, Pelekis, and Doulkeridis (2017), but also to suggest possible improvements in performance, by adding a Convolutional Neural Network (CNN) layer, preferably an 1D (Chen et al. 2017). These two papers were drawn on extensively during the Experimental Setup section of this research. Finally, another scientific work Zhou et al. (2016) was taken into account, in order to extract some knowledge about Deep methods to be used. Here the authors applied a 2D Pooling operation and a 2D Convolutional layer, to sample more useful and meaningful information from text data.

A final article by Asghar et al. (2014) was used to consider techniques commonly used for pre-processing documents and to prepare the reader for an alternative or convenient way of doing it, that will be discussed later. In conjunction with this, feature categorization and selection were outlined in order to encourage readers to extract the most appropriate documents for training their algorithms. Focusing on the pre-processing step identifies three major processes, the first one is the "Parts of Speech" or "POS tagging", the most traditional method, which assigns a tag to each word in a text depending on its grammatical type, such as verbs, adjectives, nouns and so on. Secondly, "stemming" and "lemmatization" methods are used to reduce the number of words and deal with noisy text, based on different criteria. "Lemmas" is a technique preferred to "stemming", when accuracy is not the basic evaluation metric, thus the

"lemmas" method was used in this research. Finally, the "Stop Words Removal" is highly recommended for removing the most periodic words from a document, so that applying it dimensionality reduction is performed to the review sentences.

2.3 Research gaps and shortcomings of existing methods

The papers considered in the previous section the foundation from which to take further steps to achieve more accurate results in some cases, but led to an examination of new ideas.

Most researchers in this field, start with the pre-processing part, preferring to follow the most popular direction of cleaning text sources to be accessible for building numeric vectors applicable to employee Machine Learning skills. In this project an alternative technique was proposed that may omit or incorporate the document cleaning step, known as a Pipeline model (Cranfill 2016). This method is available from "scikit-learn" library, and can be imported using the Python programming language. During this operation the pre-processing stage can be combined with either the Bag of Words (BoW) or the Term Frequency - Inverse Document Frequency (TF-IDF) representation and additionally, with the hyper-parameter tuning for our validation scope. This process emphasizes a robustness and the expectation of a more automated and accurate Machine Learning approach. Another possible gap is related to how most authors prefer to train only specific classifiers, sometimes using either only a Bag of Words (BoW) model or a Term Frequency - Inverse Document Frequency (TF-IDF) one. This research suggests that it is important to examine both modes, because each classifier meets different outcomes on each occasion, hence we cannot develop a strict rule for all tasks. Moreover, papers that consider the Deep Learning approaches provide useful knowledge on how to develop layers for the Sequential models. The only difference appears to be on the fact that the writers here connect all the methods together, in contrast to this research, that utilized them separately, producing three distinct algorithms. Previous work done in the field of Deep Learning may have reached more interesting conclusions if it had included some Machine Learning approaches, in order to identify further correlations and possibly more skillful responses, in terms of the accuracy metric.

Most of the previous work discussed, applied Sentiment Analysis of movie reviews using more than one platform, therefore the researchers' experiments were limited to training various classifiers and representing relations between two datasets. What was omitted was the ambition to merge documents, in order to develop a more complex procedure and draw more general conclusions, from both scientific and societal perspectives. This unanswered research problem was behind what motivated the attempt to develop a novel idea for text mining by trying to explain the helpful impacts of this activity.

2.4 Research questions and goals of current work

This section will reproduce the thesis research questions, the probable achievements reached during the current work, the way the technical gaps explained previously were tackled and the extent to which this project stands out from prior research. Lastly, the datasets and software used will be outlined.

RQ1. How does the large set to be developed perform against the partial ones?
The answer to this question was the main goal, which highlighted the uniqueness of this project, and made the research relevant to previous associated work done in the area of Sentiment Analysis of movie reviews. The results show that when the performance

of the datasets is evaluated individually, almost all classifiers from both Machine and Deep Learning fields of activity indicated that the IMDB's set produced a higher accuracy in comparison to the Rotten Tomatoes' set. In addition, the problem of merging new datasets, using reviews from both IMDB and Rotten Tomatoes demonstrated that, despite the large dataset, it achieved an accuracy that is equal to the best offered during the separated attempts by IMDB. This was a key success that suggests some useful conclusions. Particularly, it is noticeable that IMDB provides better information to help users with movie selection than Rotten Tomatoes, but additionally, from a technical point of view, it guides all the classifiers of the joined set to perform better.

RQ2. Is a Bag of Words (BoW) model more accurate than a Term Frequency - Inverse Document Frequency (TF-IDF) model, regarding this attempt in the field of Machine Learning binary representation?
The reason for using two ways of representing text documents, was to prove that both methods are important for each classification task, but also a demonstration that fills possible gaps in relevant work. The results show that we cannot develop a strict rule on whether a Bag of Words (BoW) model is better than a Term Frequency - Inverse Document Frequency (TF-IDF) one, for all cases. Perhaps a Term Frequency - Inverse Document Frequency (TF-IDF) model performs more accurately in most situations, though some classifiers achieved higher accuracy in the form of a Bag of Words (BoW) version. This is evident for both IMDB and Rotten Tomatoes' recommendation systems. A Term Frequency - Inverse Document Frequency (TF-IDF) method was more accurate than a Bag of Words (BoW), not only in the case of its original sets, but also in the case of the merged set put together for this research.

RQ3. To what extent is a Deep Learning classifier preferable to a Machine Learning one?
The idea of comparing various algorithms between the fields of Deep and Machine Learning rests on the need to avoid losing helpful insights and the possibility of establishing robust rules regarding the acceptability of a particular method. The Deep techniques achieved significant accuracy compared to, the Machine Learning Methods, although some Machine algorithms gave better results. Likewise, the previous statement, the large dataset reacted in a like manner, always having the IMDB's set as a leader, in terms of the efficiency of the outcomes.

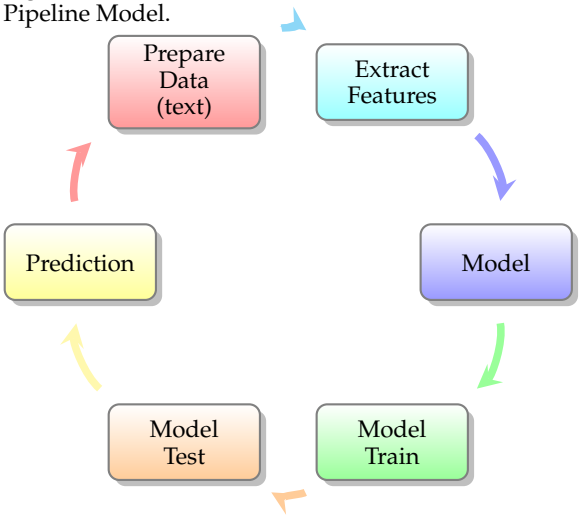
Finally, two datasets provided by the IMDB and Rotten Tomatoes provided relevant platforms. Both of them contained already extracted comments, which could be used as the attributes to capture features as numerical vectors. The target column was labeled as "positive" versus "negative", whereas the target in this research was divided into "fresh" against "rotten". That helped to retain the notion of supervised learning, to provide enough time for training various classifiers, instead of creating new groups, in order to provide responses to the above scientific reflections.

3 Methods

3.1 Machine Learning Methods

Machine learning Pipelines include various stages to train a model, but the term 'Pipeline' is ambiguous as it involves a one-way flow of data (Figure 1). Alternatively, Machine Learning Pipelines are periodic and iterative as every step is repeated to promote the accuracy of the model frequently and accomplish a workable algorithm. To create better Machine Learning models, and extract the most value from them, convenient, extensible and durable storage solutions are preferable, paving the way for

Figure 1
Pipeline Model.



on-promises object storage (Zhou 2018). This approach is also applicable to text mining, by incorporating the various classifiers, which will be introduced during the upcoming section. Additionally, to measure the sentiment of movie reviews we need to transform our text sources into two different numerical schemes; the Bag of Words (BoW) and the Term Frequency - Inverse Document Frequency (TF-IDF).

Bag of Words (BoW) is a typical approach for presenting text similarity, in terms of its terms' frequencies. This method was applied with the use of "scikit-learn" and its module called "CountVectorizer", which transformed the review text into vector representations based on the frequency of the chosen features. These vector representations were then put together in a sparse matrix, storing the indexes of the non-zero counts of tokens in the analyzed text (Silva et al. 2018).

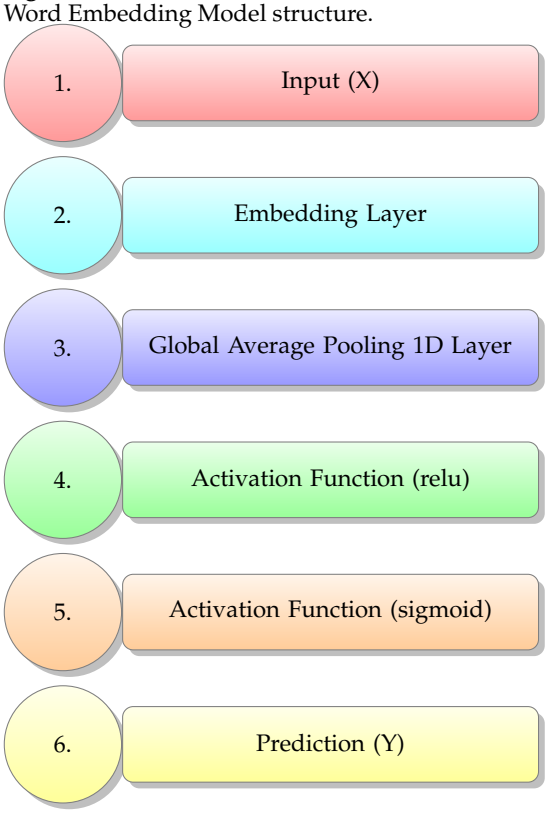
The word representation approach, Term Frequency - Inverse Document Frequency (TF-IDF) was used to measure similarities between documents. Kwon et al. (2018), defines term frequency, tf(i,j) as the frequencies of the term i in document j, which could be either a sentence or a compilation of sentences. A high tf usually implies high relevance. Document frequency df(i) measures how popular the term i is among documents. The assumption is that if the document frequency of i is high, it means that the term is highly generic and not informative. Less popular terms are considered more important to measure document similarity. To make the comparison more intuitive, the concept inverse document term frequency tf - idf is introduced, which is inversely proportional to the document frequency ($tf - idf = tf_{ij} \times \log \frac{N}{df_i}$). This method was applied with the use of "scikit-learn" and its module called "TfidfVectorizer".

3.2 Deep Learning Methods

For the Deep techniques employed in this research, three distinct models were developed incrementally. The following diagrams describe in detail the layers employed in order to develop relevant methods.

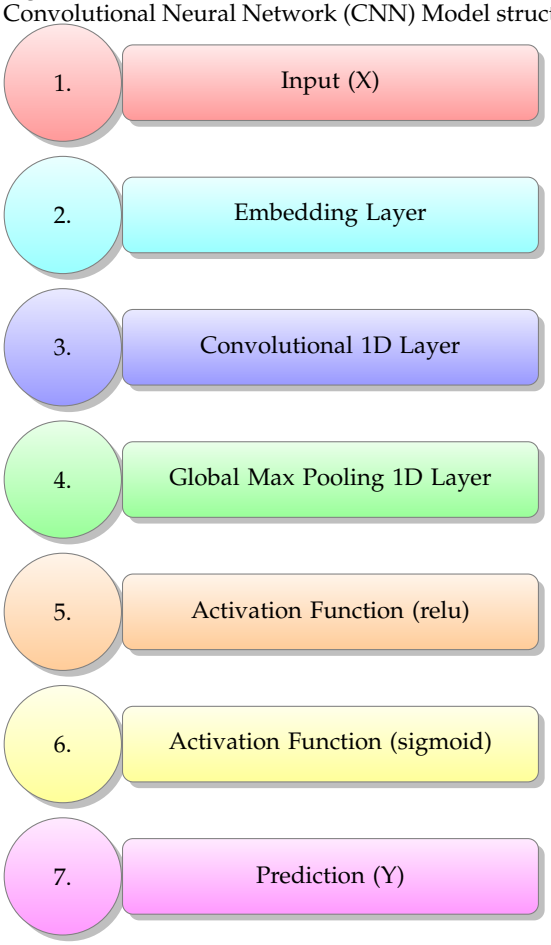
The following graph describes the simple Word Embedding approach (Figure 2), which is the first hidden layer for the next more complex tasks. The first layer served as the Input for the data. Then, an Embedding layer was added to the model which can be used when dealing with text data. The input is required to be encoded as integers and subsequently each word is represented by a unique integer. After the Embedding, a Global Average Pooling 1D layer was added. In addition, we made use of two activation functions; the "relu" first and then the "sigmoid".

Figure 2
Word Embedding Model structure.



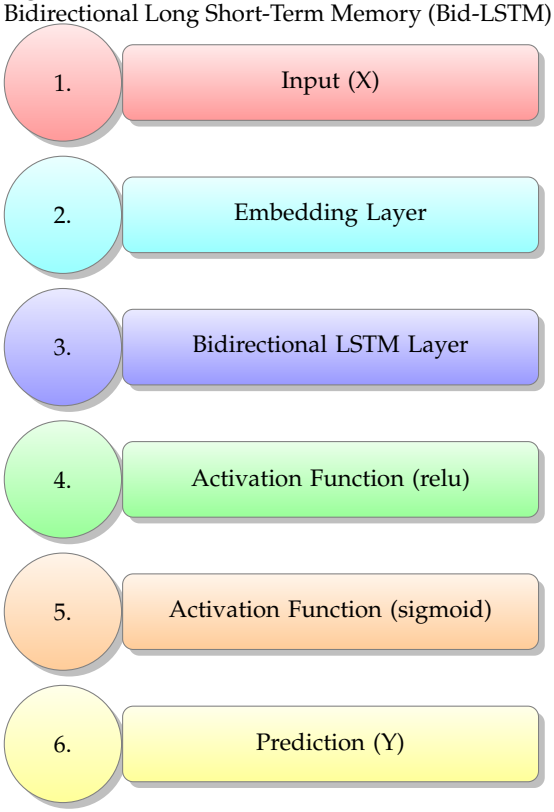
In Figure 3, the same process with the aforementioned model presented above is followed. However, there are two differences here. After the Embedding layer, a Convolutional 1D layer was added and the Global Average Pooling 1D layer was replaced by a Global Max Pooling 1D layer. This was done in order to exploit the Convolutional Neural Networks' (CNN) approach when dealing with text data.

Figure 3
Convolutional Neural Network (CNN) Model structure.



The last model that was developed is presented in Figure 4. The first two layers used for this model are the Input and an Embedding layer. A Bidirectional Long Short-Term Memory (LSTM) layer was also added. This layer can provide additional context to the model and eventually lead to a faster and robust learning process. After that, the activation functions "relu" and "sigmoid" were used for the prediction.

Figure 4
Bidirectional Long Short-Term Memory (Bid-LSTM) Model structure.



4 Experimental Setup

4.1 Data

The IMDB dataset (Maas et al. 2011), includes 50,000 reviews for Natural Language Processing (NLP) or Text Analytics. This is a dataset suitable for Sentiment Analysis containing extensively more data than previous benchmark datasets. A "review" column represents the attributes from which the relevant numerical features were extracted. Sentiments, which are partitioned into "positive" versus "negative", consist the target column. This leads to a binary task, regarding the two groups of the labeled column, but also show that we are going to carry out supervised learning functions. The number of positive and negative reviews was predicted using either Machine or Deep Learning algorithms. Finally, the format of the dataset is a "csv" file. The data was inputted, then an Exploratory Data Analysis (EDA) was applied. A description of the IMDB summary, is shown in Table 1. The text data are balanced, in consideration of their labels, hence

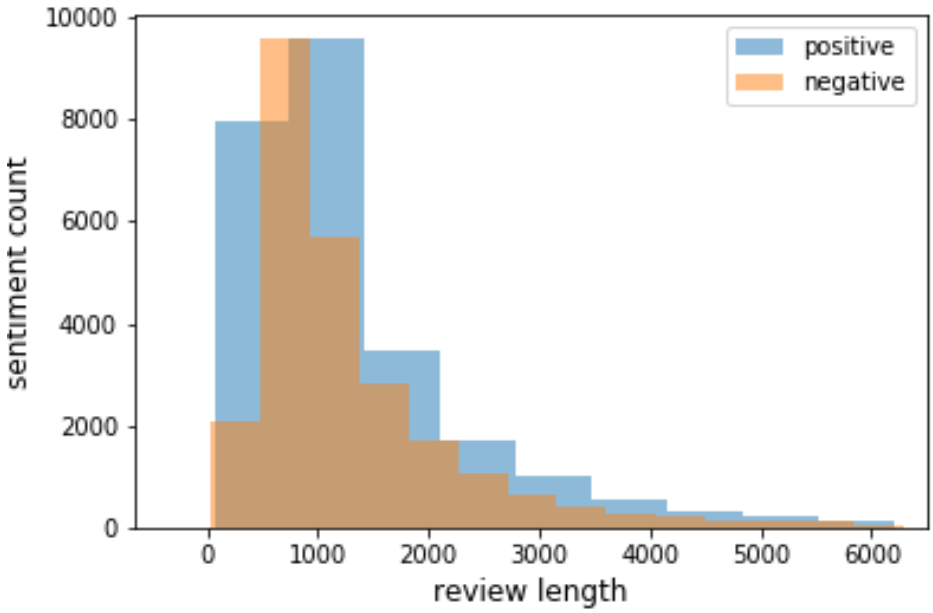
Table 1
Summary of IMDB dataset.

Dataset Statistics	Features Column (reviews)	Target Column (sentiments)
number of rows (missing values excluded)	50000	50000
number of unique rows	49582	2
most common value	Loved today's show!!! It was a variety and not...	positive
most common value's frequency	5	25000

25,000 rows for "positive" sentiments were detected and another 25,000 for "negative" ones.

The next stage was to identify any missing values that should be removed, in order to avoid the text sources becoming noisy. In this case the primary form of the dataset was completely cleaned, meaning that all the rows could be used. A third column for visualization was added to the set in order to determine whether a correlation existed between the sentiment rating and the length of each review (Figure 5).

Figure 5
Positive against Negative sentiments, regarding the length of reviews.



The Rotten Tomatoes dataset consists of two files in "tsv" format, the first one is called "movie_info" and the other one "reviews". There are multiple columns in each file, with the immediate focus on the "id" column, the key to understanding the sources. The second file, containing the column named "review", refers to relevant attributes in this research and the "fresh" column is the target, for which 54,432 rows are available. The last column is separated into "fresh" and "rotten" feelings, in other words sentiments

that correspond to the first dataset's target, and which can be linked for the merged set's binary tasks, in order to perform classifiers and make the comparisons planned. The dataset was scraped from Rotten Tomatoes platform (Miller 2019). The first step was to load the data and then employ an Exploratory Data Analysis (EDA) to propose a quick summary between the comments and the feelings of users, shown in Table 2. In contrast

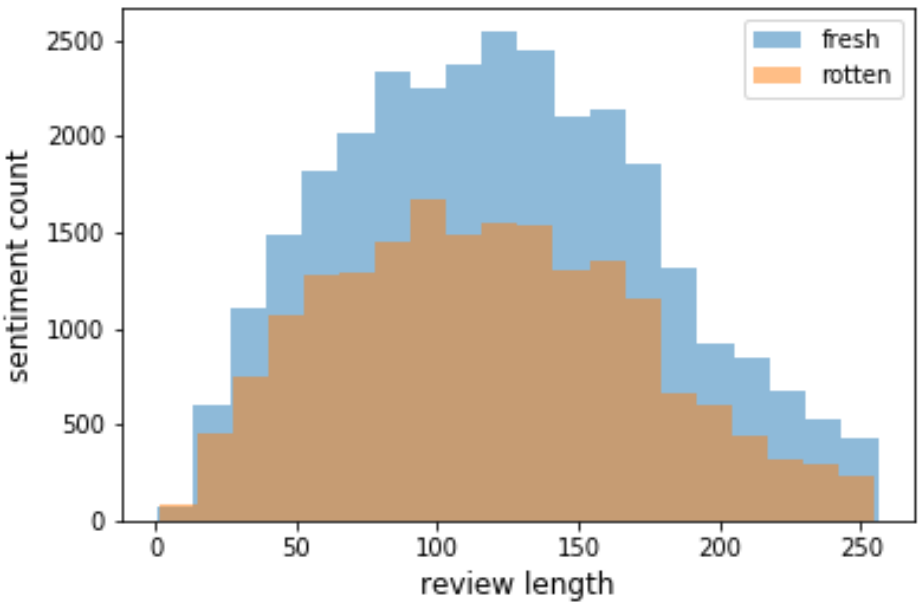
Table 2
Summary of Rotten Tomatoes dataset.

Dataset Statistics	Features Column (reviews)	Target Column (sentiments)
number of rows (missing values excluded)	48869	54432
number of unique rows	48682	2
most common value	Parental Content Review	fresh
most common value's frequency	24	33035

to the IMDB dataset, the dataset here can be considered quite unbalanced regarding the binary separation of the target groups, in which 33,035 are available as "fresh" and the remaining 21,397 as "rotten".

In addition, some rows contained empty documents, which were removed, in order to train the models appropriately. During the next stage we worked in the same way as previously, hence creating a third column containing the length of the reviews, to examine possible relations with the sentiments recorded (Figure 6).

Figure 6
Fresh against Rotten sentiments, regarding the length of reviews.



The pre-processing step for both sets and the merged one follow a distinct direction, regarding the Machine Learning tasks, in opposition to Deep Learning ones. In order to implement the classification algorithms, the Pipeline models method was exploited, which embodies the pre-process of the reviews, such as removing the stopwords, punctuation, using lower case words in a sentence, applying "stemming" and "lemmatization", using either the Bag of Words (BoW) or the Term Frequency - Inverse Document Frequency (TF-IDF) approach. Extra time was gained to train the high rate of classifiers desired. Regarding the Deep modules, a manual analysis was made to set the models' parameters. Specifically, we used a vocabulary size which represented the number of the most common words, a padded type, a type of tokenization and a standard Embedded layer size, to tokenize, sequence and pad our features, to be applicable for the input Deep layer, in order to fit, compile, train and finally evaluate our models. For the joined case, we connected the two datasets, and before doing that the Rotten Tomatoes' target groups renamed as "positive" and "negative", rather than "fresh" and "rotten", so that they matched the IMDB's labels, and worked in accordance with the previous techniques. During all procedures the datasets were split, keeping 70% for the training sets and a 30% for the testing sets. Additionally, a "random state" was used so that scripts could be reproduced by interested users. Lastly, a true value was set to the "shuffle" command.

4.2 Algorithms

This subsection introduces and briefly explains the algorithms used, together with their mathematical definition. The various modes were exploited three times in total, firstly for the IMDB, then for Rotten Tomatoes and finally for the joined set. During the first part, the Machine Learning tasks will be discussed, whereas the second one will refer to the Deep Learning algorithms.

4.2.1 Machine Learning Classifiers

Logistic Regression
In Logistic Regression there are a set of predictors $x' = (x_1, ..., x_n)$ and a binary outcome equal to 0 or 1 called y. If we assume that $x_1 = 1$, then an intercept term for the model is determined. In addition, a probability $p(x) = Pr\{y = 1|x\}$ model is introduced, under two assumptions: a. $y|x \sim Bernoulli[p(x)]$ and b. $\log(\frac{p(x)}{1-p(x)}) = \sum_{j=1}^{k} \beta_j x_j$, where b_j are the coefficients plus the intercept term. The technique calculates the probability of a review being positive or negative using the Sigmoid Function $y = \frac{1}{1+e^{-y}}$ (Hoadley 2020), assuming a default threshold equal to 0.5. If the predicted value is above this threshold the review would be classified as positive sentiment, otherwise as negative.

Linear Support Vector Machines (SVM)
Another common technique in applying Sentiment Analysis of movie reviews, also applicable within a Pipeline method, is the Linear Kernel that Support Vector Machines (SVM) provide, in order to classify two separate groups of user feelings. This is performed by calculating the largest distance between the features and the margin of a spatial hyper-plane (Al Amrani, Lazaar, and El Kadiri 2018). Let $f(x)$ be SVM's hyperplane that partitions features into two bounds, if an instance x_i and its estimation is $f(x_i) > 0$, then x_i is positioned to the right side, in contrast, if $f(x_i) < 0$ then it is

placed on the left side respectively, where $f(x) = w^T + b$, w : normal to the decision boundary and b : bias. By calculating the distance of each x_i, which represent the relevant reviews as numerical vectors, to the two margins developed, two classes are created for classifying positive against negative feelings. Finally, since the Linear mode of Support Vector Machines (SVM) is applied, the hyper-plane is a "stiff" line defined by a Sigmoid Function $f(x) = sgn(w^T + b)$.

Multinomial Naive Bayes
Multinomial Naive Bayes is a well-known probabilistic supervised method for accurately dividing binary problems measuring the sentiment of movie reviews. This approach calculates the probability of individual features of a document d given a class c, requiring that our features are independent of each other (Hemmatian and Sohrabi 2017). Each time it is determined whether a data point belongs to class c or not, in order to predict if a comment belongs to a positive or a negative group. To estimate the former we can apply the following formula $p(c|d) = \frac{p(c)p(d|c)}{p(d)}$.

Extreme Gradient Boosting (XGBoost)
This is another classifier suitable for Sentiment Analysis in the field of handling and analyzing text sources, and is widely used to deal with various Machine Learning and Data Mining tasks (Gaye and Wulamu 2019). This classifier is known as a Decision Tree-based ensemble method that uses a gradient boosting framework. Its system optimization yields three stages: parallelization, tree pruning and the hardware optimization. In addition, this mode makes some important augmentations, such as regularization, sparsity awareness, weighted quantile sketch and cross-validation (Morde and Setty 2019).

Random Forest
Another useful aspect of binary sentiment classification is the Random Forest technique. It is used as an ensemble learning module based on a Decision Tree algorithm. It grows multiple trees simultaneously concluding in a Random Forest. All data points are assigned to a category based on a majority vote rule. The more trees included in the forest, the higher the accuracy is likely to be produced by the Random Forest (Fauzi 2018).

K-Nearest Neighbors (K-NN)
K-Nearest Neighbors (K-NN) classifies an instance by estimating its distance with the k-neighbors and picking the dominant class among them, which is defined by the weighted minimum average distance of that instance to its k-neighbors. A commonly used distance metric is the Euclidean distance used to estimate the similarity between the instances and each k-neighbor (Zhang et al. 2019). Here the weights of the documents are provided either by the Bag of words (BoW) or the Term Frequency - Inverse Document Frequency (TF-IDF) representations. The distance of those weights and a group of k-neighbors is then calculated, to produce the correct match of individual data points. In this research, it proved to be the weakest method in terms of the accuracy produced, as shown in the next section. However, its performance stability between the primary datasets and the merged one was sufficient to include it in the project as the simplest process for comparisons.

4.2.2 Deep Learning Models

Word Embedding
This model is the simplest representation of a Deep Neural Network, that was applied alone and in addition, was the first hidden layer of the more complex models that follow. It contains three aspects: the size of the vocabulary in the text data, the size of the vector in which words will be Embedded, and the length of the input sequences (Brownlee 2019). These parameters were applied during the pre-processing step explained previously. Despite its simplicity, we included it as a separated attempt because of its high performance during all the fields of activity, but also for the importance of providing a baseline method. Keras provides an Embedding layer for Neural Networks applicable for text data (Gal and Ghahramani 2016).

Convolutional Neural Network (CNN)
The 1D Convolutional Neural Network (CNN) is one of the most common techniques for text classification. It takes sentences of varying lengths as input and produces fixed length vectors as output. Before training, Word Embeddings for each word in the glossary of all input sentences are generated. All the word Embeddings are stacked in a matrix. In the input layer, Embeddings of words comprising current training sentences are taken from the matrix. The maximum length of sentences that the network handles is set. Longer sentences are cut, in contrast, shorter sentences are padded with zero vectors (Chen et al. 2017).

Bidirectional Long Short-Term Memory (Bid-LSTM)
The Long Short-Term Memory (Bid-LSTM) is an extension of a Recurrent Neural Network (RNN) method to deal with the gradient problem that the Recurrent Neural Network (RNN) faces. The common Long Short-Term Memory (Bid-LSTM) consists of three stages; the forget, the input and the output gate. This model encodes the sequence from only one direction, but if two are used, a different approach called Bidirectional Long Short-Term Memory (Bid-LSTM) can be developed, in order to encode sequences from both directions (Ma, Peng, and Cambria 2018; Baziotis, Pelekis, and Doulkeridis 2017).

To sum up, for the initialization of the Deep Learning approaches, a Sequential model was developed. First, an Embedded layer along with the corresponding layer for each method were added. After that, for the Word Embedding and the Convolutional Neural Network (CNN) approaches, a Global Average Pooling 1D and a Global Max Pooling 1D layer were included, respectively. All models made use of both the "relu" and the "sigmoid" activation functions. These were compiled by using a "binary_crossentropy" function as a loss measure, while the optimizer was set equal to "adam" and the metric was the "accuracy". The selected models were fitted and trained using 4000 "epochs" in each situation and a "verbose" value equal to 1. Finally, an "EarlyStopping" command was applied, to avoid overfitting and a "ModelCheckpoint" mode to hold only the model which produced the best performance.

4.3 Software

The main programming language was Python, and all the analyses were scripted in a Jupyter notebook.

For each classifier, six scripts were used individually for training and evaluating the models' performances, two for each dataset occasion, either using a Bag of Words (BoW) or a Term Frequency - Inverse Document Frequency (TF-IDF) model. The Word Embedding, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models were applied three times, one for each dataset. In addition, a wide variety of packages were used for visualization, pre-processing the text, extracting the features, training algorithms and evaluating their performances, including the following: "numpy", "pandas", "seaborn", "matplotlib", "scikit-learn", "xgboost", "keras" and "tensorflow".

The "pandas" and "numpy" packages were used to handle the dataframes and arrays that were developed, before and during the training part of the models. "Seaborn" and "matplotlib" were responsible for the visualization stage, during the Exploratory Data Analysis (EDA) and the comparisons of the models. The Machine Learning classifiers were implemented by the contribution of "scikit-learn" library, except the "xgboost" classifier that was taken from the homonymous library. In conclusion, "tensorflow" with the help of "keras" were the basic softwares used for Deep models, taking over all stages, from pre-processing the documents until fitting, training and testing the efficiency of the models.

4.4 Evaluation Metric

To evaluate the results obtained after fitting the above mentioned classifiers, we took into account the Accuracy metric, that rests on the four classes; True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN), administered by the Confusion Matrix.

Accuracy is the ratio of the correctly predicted cases to the total number of input samples, $Accuracy = \frac{TP+FN}{TP+TN+FP+FN}$ (Hossin and Sulaiman 2015). True Positive (TP) reflects the number of cases that are correctly identified as having the positive class, while True Negative (TN) represents cases that are correctly labelled as negative. False Positive (FP), also known as "Type I error", depicts the number of cases that are wrongly predicted as positive, but were actually negative. The last group, False Negative (FN), also known as "Type II error", carries the same interpretation but for the incorrectly identified negative cases. These four measurements are recognizable on a table called Confusion Matrix (Tharwat 2018), (Table 3).

Table 3
Confusion Matrix.

		Prediction Outcome	
		Positive	Negative
Actual Value	Positive	TP	FN
	Negative	FP	TN

5 Results

In this section the results of our scientific research are identified. It is evident from the outcomes that the K-Nearest Neighbors (K-NN) classifier was the poorest performing one, in terms of the classification accuracy metric, hence it was used as the baseline method for the Machine Learning tasks. The Word Embedding model was accepted as

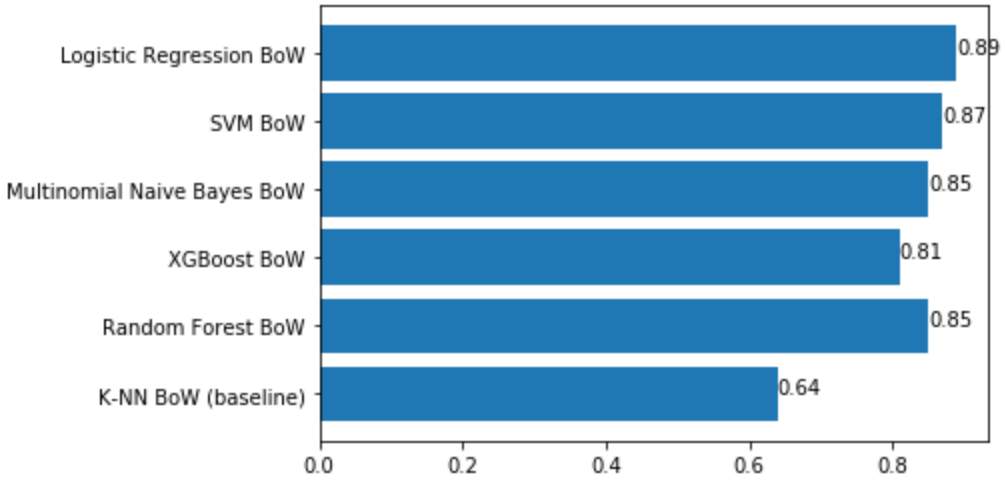
the baseline one, in order to make comparisons against the more complex algorithms implemented in the field of Deep Learning.

5.1 IMDB Dataset

5.1.1 Machine Learning Classifiers

Bag of Words (BoW) Models
In the Bag of Words (BoW) implementation, the Logistic Regression classifier was the strongest one, an 89% accuracy. The second most accurate was the Linear Support Vector Machines (SVM) with 87%. In addition, Multinomial Naive Bayes, Random Forest and Extreme Gradient Boosting (XGBoost) models were reasonably competitive achieving 85%, 85% and 81% respectively. The worst result in the experiments was the K-Nearest Neighbors (K-NN) classifier, which returned 64% accuracy (Figure 7).

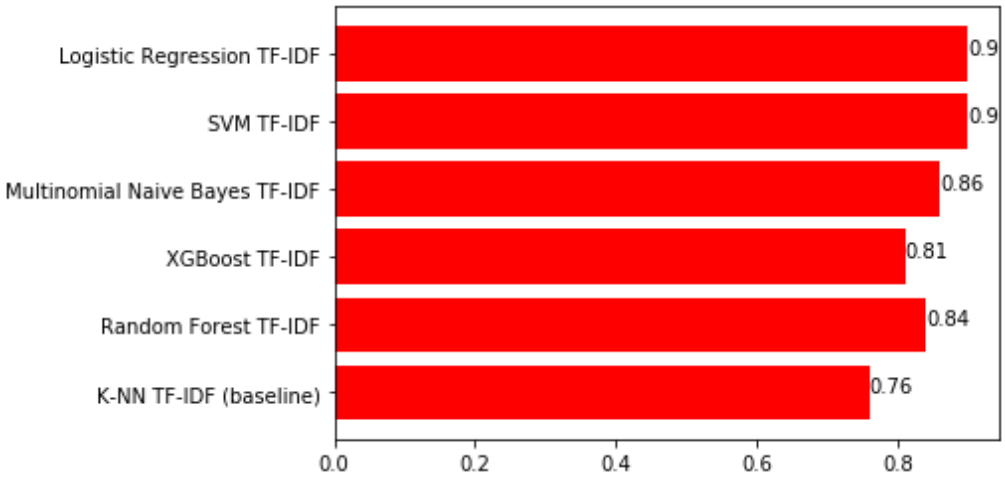
Figure 7
Bag of Words (BoW) Models for IMDB.



Term Frequency - Inverse Document Frequency (TF-IDF) Models
On the other hand, the results were better for the Term Frequency - Inverse Document Frequency (TF-IDF) representations, which provided equal or improved performances. Logistic Regression and Linear Support Vector Machines (SVM) reached 90% accuracy, and Multinomial Naive Bayes and Random Forest gained 86% and 84% respectively. The Extreme Gradient Boosting (XGBoost) model gave the same accuracy as previously (81%). Likewise, K-Nearest Neighbors (K-NN) was the poorest performing classifier, though better than before with 76% (Figure 8).

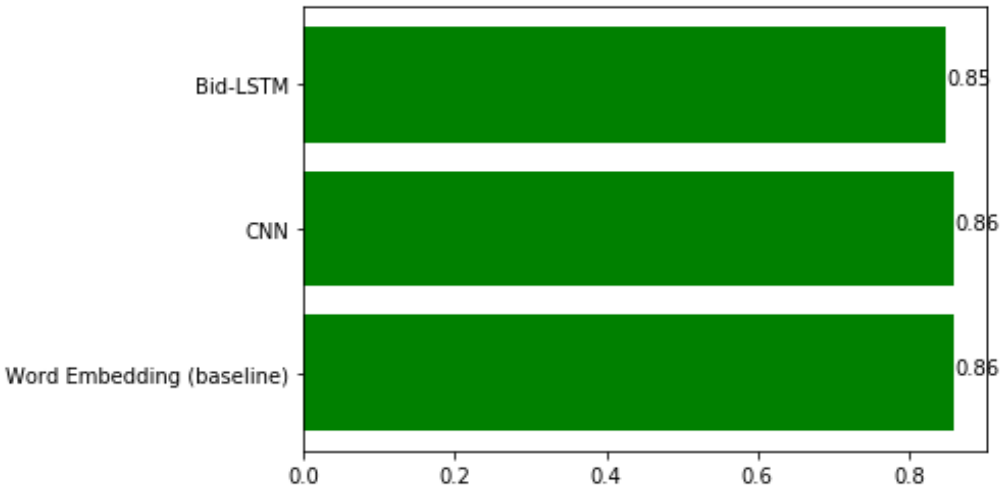
5.1.2 Deep Learning Models
The Deep models in this section returned quite impressive performances, but also showed some stability regarding responses. Thus, it is noticeable that the simplest version of a Word Embedding model and the Convolutional Neural Network (CNN)

Figure 8
Term Frequency - Inverse Document Frequency (TF-IDF) for IMDB.



achieved 86%, whereas the Bidirectional Long Short-Term Memory (Bid-LSTM) returned a slightly lower 85% (Figure 9).

Figure 9
Deep Learning Models for IMDB.

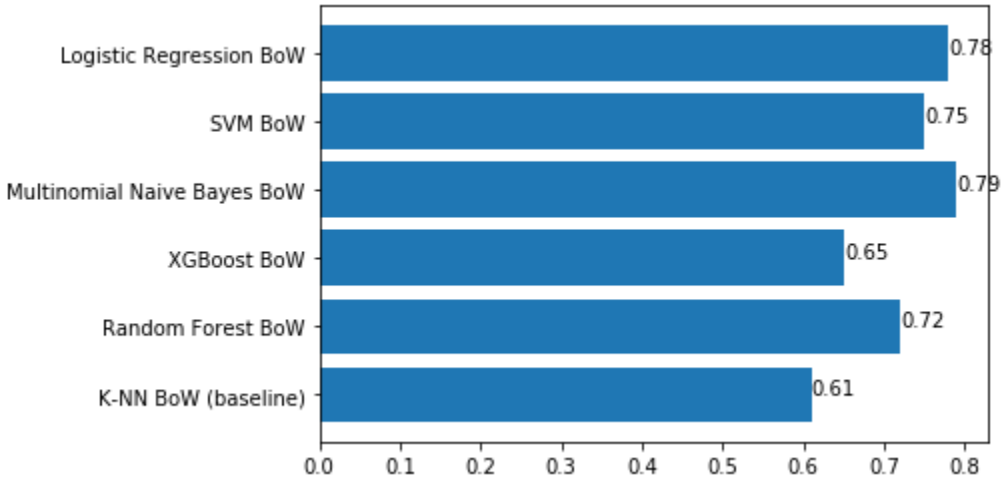


5.2 Rotten Tomatoes Dataset

5.2.1 Machine Learning Classifiers

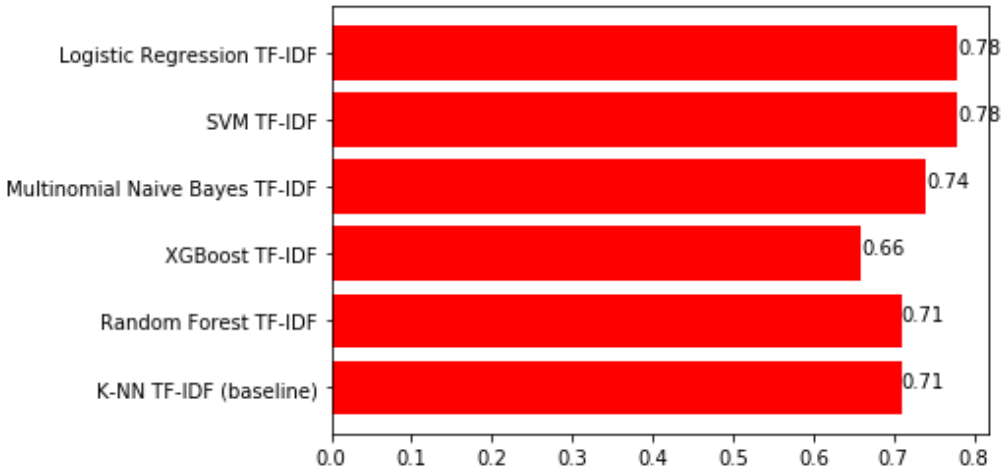
Bag of Words (BoW) Models
Regarding the Bag of Words (BoW) models, the Multinomial Naive Bayes classifier achieved the highest accuracy with 79%, and Logistic Regression followed with 78%. The Linear Support Vector Machines (SVM) returned 75%, followed by Random Forest with 72%. Lastly, the weakest classifiers were the Extreme Gradient Boosting (XGBoost) and the K-Nearest Neighbors (K-NN) a with 65% and 61% accuracy respectively (Figure 10).

Figure 10
Bag of Words (BoW) Models for Rotten Tomatoes.



Term Frequency - Inverse Document Frequency (TF-IDF) Models
In contrast, the Term Frequency - Inverse Document Frequency (TF-IDF) modules provided slightly different results. The greatest accuracy was returned by Logistic Regression and the Linear Support Vector Machines (SVM) with 78%. Multinomial Naive Bayes reached 74%, both Random Forest and K-Nearest Neighbors (K-NN) were 71% accurate. The poorest performance was returned by the Extreme Gradient Boosting (XGBoost) method, which achieved only 66% accuracy (Figure 11).

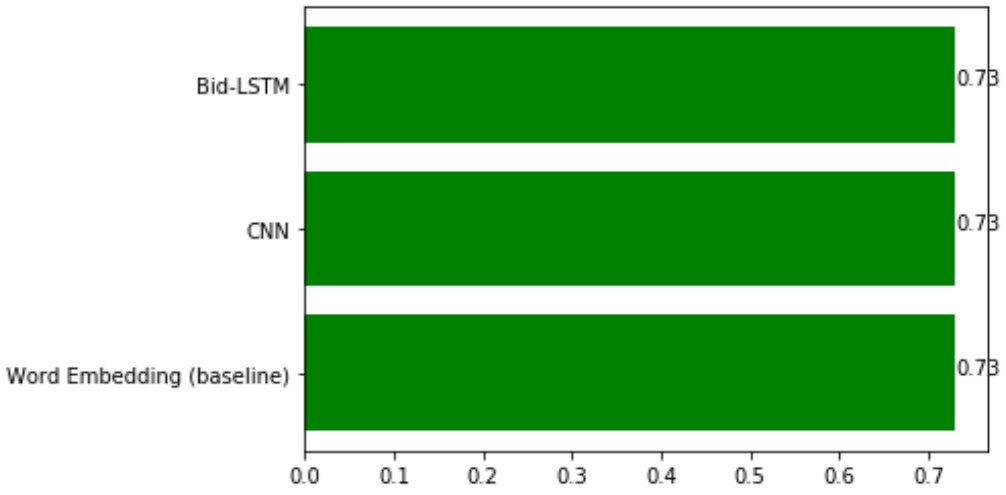
Figure 11
Term Frequency - Inverse Document Frequency (TF-IDF) Models for Rotten Tomatoes.



5.2.2 Deep Learning Models

The Rotten Tomatoes dataset was tested for the efficiency of the three Deep approaches. Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bid-LSTM) methods, compared to the baseline Word Embedding, showed the same accuracy with 73%. This outcome reveals a similarity among those techniques and the importance of using them (Figure 12).

Figure 12
Deep Learning Models for Rotten Tomatoes.

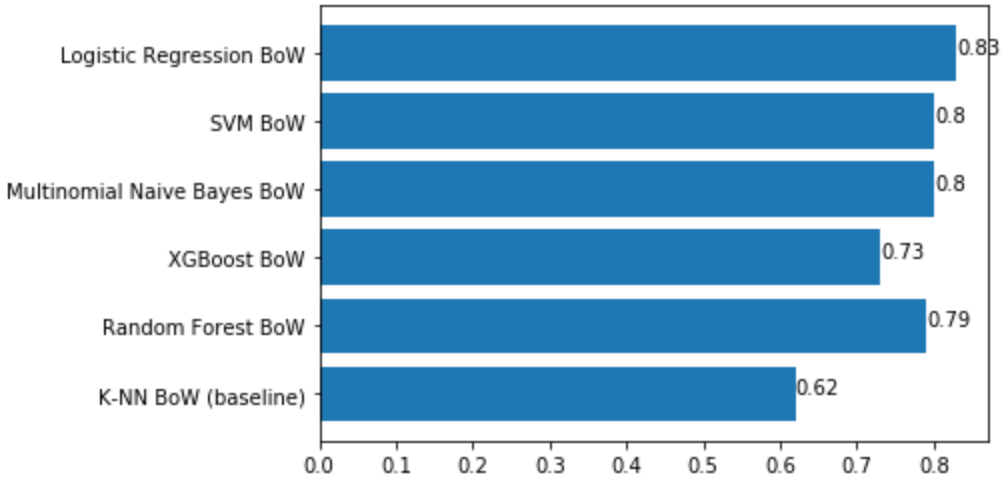


5.3 Merged Dataset

5.3.1 Machine Learning Classifiers

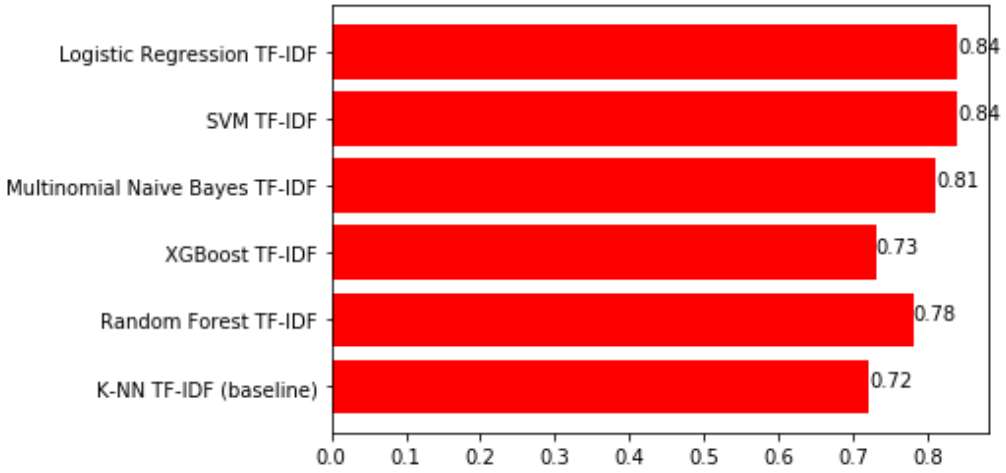
Bag of Words (BoW) Models
The merged set achieved significant results in terms of the Bag of Words (BoW) model. Again the Logistic Regression model was the most accurate with 83%. Linear Support Vector Machines (SVM) and Multinomial Naive Bayes showed a high accuracy of 80%, and Random Forest obtained almost the same with 79%. Next in terms of accuracy was the Extreme Gradient Boosting (XGBoost) model with 73%. Finally, K-Nearest Neighbors achieved the lowest accuracy (62%), an acceptable result, considering its performance during the tests with separated sets (Figure 13).

Figure 13
Bag of Words (BoW) Models for the Merged Dataset.



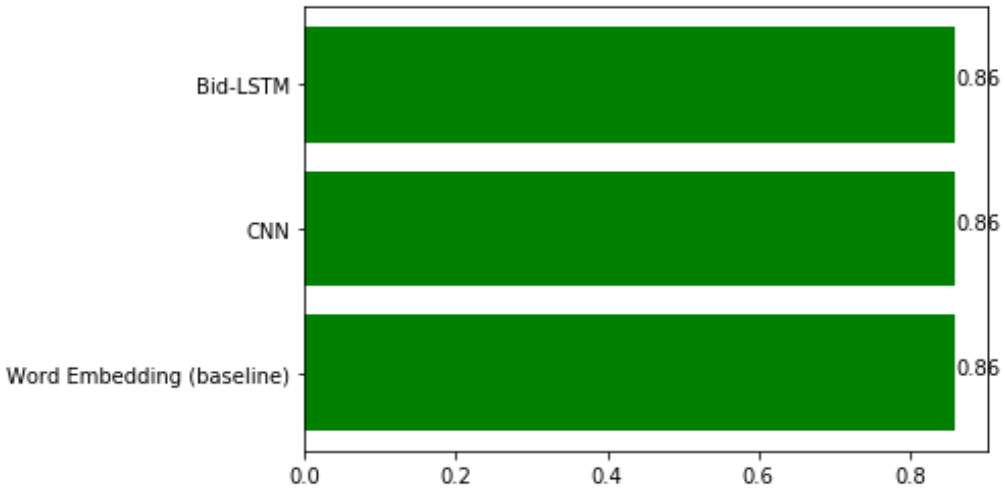
Term Frequency - Inverse Document Frequency (TF-IDF) Models
The Term Frequency - Inverse Document Frequency (TF-IDF) representation gave practical evidence of the models' strong achievements. The most accurate was the Logistic Regression and Linear Support Vector Machines (SVM) classifiers that obtained 84% accuracy. Multinomial Naive Bayes and Random Forest were a little less accurate with 81% and 78% respectively, whereas the Extreme Gradient Boosting (XGBoost) classifier returned the same accuracy as previously (73%). Again K-Nearest Neighbors (K-NN) was the least accurate with 72% (Figure 14).

Figure 14
Term Frequency - Inverse Document Frequency (TF-IDF) Models for the Merged Dataset.



5.3.2 Deep Learning Models
Here the experience was similar to the original sets previously. All the processes reached almost identical performances between 85% and 86%. This provided extra evidence of some harmony among the more complex procedures (Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (Bid-LSTM)) and the simplest-baseline one (Word Embedding) (Figure 15).

Figure 15
Deep Learning Models for the Merged Dataset.



6 Discussion

This section reproduces the research questions and discusses the main findings regarding the results presented previously, in order to clarify to what degree the research questions were significantly answered.

RQ1. How does the large set to be developed perform against the partial ones? This was the question that motivated the attempt to develop a scientific procedure by introducing a comparison between the performances of various algorithms exploited both in the individual sets and in merged sets also. It is evident from the results that the IMDB set returned more accurate outcomes compared to the equivalent algorithms employed by the Rotten Tomatoes set. As previously stated, the large dataset was created by merging the primary sets, resulting in performances that were significantly influenced by IMDB, which produced the most powerful results. Focusing on each approach separately (Appendix 4), regarding the best performing models, it is justifiable to conclude that in these tests an accuracy was reached that was either almost equal to the weighted sum, or almost identical to the highest individual set. This claim is technically acceptable because all the text data from both platforms were merged, in order to make a fair implementation, taking into account every source that was applicable. In terms of IMDB, the most powerful models were the Logistic Regression and Support Vector Machines (SVM) classifiers, determined by a Term Frequency - Inverse Document Frequency (TF-IDF) representation, which is evident by the feature importance applied to plot the top 20 positive words against the top 20 negative words that were taken into account (Appendices 5.2 and 5.3). The main idea of feature importance is, if a coefficient is high, a meaningful word is used for the "positive" group. In contrast, if a coefficient is low, a relevant word is used for the "negative" group. The same implementation measure was used for the Rotten Tomatoes' set, where the most accurate model was the Bag of Words (BoW) Multinomial Naive Bayes, with the only difference that this method used less common words compared to the other techniques, in order to determine the

most important features (Appendix 5.4). All the coefficients were negative, hence the top 20 that were next to the 0 value were accepted as positive, and the 20 which were far from 0, as negative. Finally, the new set that was built proved that a Bidirectional Long Short-Term Memory (Bid-LSTM) Deep model was the most accurate one. It returned higher accuracy and smaller loss in the validation set, in comparison to the equivalent rates that the training set contributed (Appendix 2.3).

RQ2. Is a Bag of Words (BoW) model more accurate than a Term Frequency - Inverse Document Frequency (TF-IDF) model, regarding this attempt in the field of Machine Learning binary representation?
The response to this research question depends on the classifiers, the text sources and the different ways of extracting features. "CountVectorizer" and "TfidfVectorizer" were used to support this research. A head-to-head visualization (Appendix 1) suggests that we cannot strictly argue whether a Bag of Words (BoW) or a Term Frequency - Inverse Document Frequency (TF-IDF) process is better in all cases here. This was not unexpected as each representation reacts differently for each classification task. One conclusion that could be drawn is that a Term Frequency - Inverse Document Frequency (TF-IDF) model was more accurate than a Bag of Words (BoW) model in respect of IMDB and the merged platforms, something that is expected in relation to the fact that a Term Frequency - Inverse Document Frequency (TF-IDF) method takes into account both the most frequent and less frequent terms in a document, in contrast to Rotten Tomatoes where a Bag of Words (BoW) mode proved to be the most preferable. The contribution of the Pipeline models should be noted. This was a motivation to process the documents to fit, train and test models rapidly because of the fact that the process was run three times during the project, in order to cover all the available cases. In this area of the Machine Learning approaches the inclusion of a feature importance reflection method was critical. Three models applied by the Bag of Words (BoW) method were used against six employed by the Term Frequency - Inverse Document Frequency (TF-IDF) representation (Appendix 5), accepted as the most accurate, confirmed that the second mode was slightly superior to the first one, during each of the three stages which were examined.

RQ3. To what extent is a Deep Learning classifier preferable to a Machine Learning classifier?
Regarding the original datasets, we can observe that the IMDB recommendation system models returned the highest accuracy (90%) namely Logistic Regression and Linear Support Vector Machines (SVM), in terms of a Term Frequency - Inverse Document Frequency (TF-IDF) representation, whereas the simple Word Embedding only reached 85% accuracy. This suggests that a Machine Learning approach is more accurate than a Deep Learning one (Appendix 3.1). The Rotten Tomatoes web system gave similar results, as the Multinomial Naive Bayes classifier obtained an accuracy of 79%, as a Bag of Words (BoW) module, whereas the Deep attempts achieved 73% to 74% accuracy (Appendix 3.2). Going to the merged stage, a different scenario is observable. The non-linear Deep methods outperformed the linear Machine Learning processes showing 85% to 86% accuracy, whereas Logistic Regression and Linear Support Vector Machines (SVM) achieved 84% accuracy (Appendix 3.3). One explanation could be that the more text data that were added, the more complicated the model became, hence a non-linear technique responded more accurately. This is in opposition to the original sets which were simpler, in terms of handling fewer documents, where a linear module classified the available reviews more precisely than a Deep one. Lastly, a clearer picture is evident if we focus on the performance of the Bidirectional Long Short-Term Memory (Bid-LSTM) model, gained in the joined set, where the test accuracy was more accurate than

the trained one, but also showed lower loss (Appendix 2.3), in comparison to those that were implemented during the original datasets' process (Appendices 2.1 and 2.2). This conclusion might be considered unwelcome, but the explanation rests in the fact that the Rotten Tomatoes' classes were unbalanced, something that is a motivation to test sets to achieve higher standards than train sets. Lastly, the poorer performance shown in the Rotten Tomatoes target column is evidence of the poorer accuracy of its Machine Learning modules in comparison to those accomplished by IMDB.

Finally, this project shows that both platforms, IMDB and Rotten Tomatoes, are trying to classify positive versus negative emotions in a similar manner, something that the merged stage proved. In addition, some further plans are suggested that may ameliorate some possible limitations of the current work. The time was limited due to the fact that three times nine algorithms had to be implemented. The Machine Learning classifiers are unlikely to be improved, in terms of accuracy, though future improvement of the Rotten Tomatoes perspective might be possible. On the other hand, the Deep techniques might obtain better results if different layers were added or exclude some less appropriate ones excluded from the current attempts.

7 Conclusion

This project has produced results to show how accurately a movie search machine can classify users' sentiments into a binary pattern. Both recommendation engines gave promising support to the project's primary aim, which was to determine the double advantage that readers could gain. Firstly, in the technical field, in terms of the algorithms used, in order to ensure that both sites provide a balanced reaction in the way they separate different comments into positive and negative. Secondly, the other objective was the extent of reliability of the platforms to be competitive in terms of people's wishes, in other words an automated decision-making mechanism which could benefit a user's choices about movie selection. These two requirements were largely fulfilled, regarding the results that were produced for the individual work done on each dataset, but additionally, taking into account the further step carried out by connecting the two sources, in order to improve the standards proposed.

RQ1. How does the large set to be developed perform against the partial ones?
The outcomes presented in the previous sections show that developing an extensive dataset that could deal with numerous documents from two engines could respond in an efficient manner, in terms of our evaluation metric. Moreover, the new recommendation system established succeeded in classifying emotions, set against the performances of both original sets. The results here demonstrate that a merged set can match the highest accuracy displayed by the two distinctive datasets, or at least attain middle level accuracy, taking the average efficiency that the lone sets reached. This is encouraging both from a technical and a social perspective.

RQ2. Is a Bag of Words (BoW) model more accurate than a Term Frequency - Inverse Document Frequency (TF-IDF) model, regarding this attempt in the field of Machine Learning binary representation?
Firstly, it should be noted that both representations provided powerful targets, in terms of classification accuracy. K-Nearest Neighbours (K-NN) was exploited as the simplest-baseline method, enabling the creation of more accurate schemes for making comparisons. As stated previously, it cannot be determined conclusively whether a Bag of Words (BoW) model is more accurate than a Term Frequency - Inverse Document Frequency (TF-IDF) or not. That is not a conclusion that can be answered directly in this investigation, nonetheless, it is based on the fact, that dissimilar classifiers learn their

inputs in disparate ways when they are geared with either the first or the second method of extracting numerical features from raw text data. What is an important feature of this research is the high level of accuracy obtained by employing such methods during almost all the fields of activity.

RQ3. To what extent is a Deep Learning classifier preferable to a Machine Learning one?
Both sets of results, presented as percentages and visualization figures, contributed to the conclusion that a Deep Learning classifier could be competitive against the common Machine Learning tasks. That was evident when the focus was on the merged set that was developed, which showed that a baseline model containing only a Word Embedding layer, but also a more complicated one, such as a Bidirectional Long Short-Term Memory (Bid-LSTM) or a Convolutional Neural Network (CNN), could achieve greater accuracy among all the applied methods. In contrast, that was not the case when we tested the performances of individual sets where a Machine Learning classifier, as mentioned above, returned a greater degree of accuracy. Returning to the first research question, which underlines the main expectations and objective of this project, then it might be observed that a Deep model is more preferable than a simple one, in terms of the merged experiment. Another possibility might suggest selecting between Logistic Regression or Support Vector Machines (SVM) classifiers, implemented by the Term Frequency - Inverse Document Frequency (TF-IDF) method, in order to overcome the unwelcome results of the Deep processes of the merged dataset, which was caused by the unbalanced groups of sentiments.

In conclusion, Sentiment Analysis is an area of study that requires further research. One feasible possibility would be to extend the procedure in this research, by applying Opinion Mining on the equivalent data sources, and by applying modifications during the pre-processing and the model selection stage. Machine Learning methods appear limited, if compared to the results obtained using a rapid Pipeline model, as in this research. It does not mean that different classifiers could not be used, or existing ones altered. On the other hand, the Deep techniques are waiting for further improvement, either by the introduction of new models, or by adding new layers to the existing ones. Nine models were employed three times in this project, twice for the original sets and once for the merged set, making twenty-seven efforts. Time became limited for further detailed work, thus it may be advantageous for future researchers to split an extra validation set for hyper-parameter tuning, as cross-validation was not applied here, but which might have lead to more accurate results.

8 Acknowledgements

At this point, I would like to thank my parents for their support during the writing of the current scientific work, which was mainly psychological, as well as my supervisor who gave me useful advices in the technical parts and encouraged me to achieve the maximum possible result.

References

Al Amrani, Yassine, Mohamed Lazaar, and Kamal Eddine El Kadiri. 2018. Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127:511–520.

Asghar, Muhammad Zubair, Aurangzeb Khan, Shakeel Ahmad, and Fazal Masud Kundi. 2014. A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research*, 4(3):181–186.

Baid, Palak, Apoorva Gupta, and Neelam Chaplot. 2017. Sentiment analysis of movie reviews using machine learning techniques. *International Journal of Computer Applications*, 179(7):45–49.

Baziotis, Christos, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 747–754.

Brownlee, Jason. 2019. How to use word embedding layers for deep learning with keras. `https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/`.

Chen, Tao, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230.

Conneau, Alexis, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Cranfill, Ryan. 2016. How to build a social media sentiment analysis pipeline with scikit-learn. `https://github.com/ryan-cranfill/sentiment-pipeline-sklearn`.

Desai, Prssanna Digant, Simran Deshmukh, Surbhi Gawande, Arnav Chakravarthy, and Ishani Saha. 2018. Hybrid architecture for sentiment analysis using deep learning.

Dey, Lopamudra, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. 2016. Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

Dridi, Amna and Diego Reforgiato Recupero. 2017. More sense: Movie reviews sentiment analysis boosted with semantics. In *EMSASW@ ESWC*.

Fauzi, Muhammad Ali. 2018. Random forest approach for sentiment analysis in indonesian language.

Gal, Yarin and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1027–1035, Curran Associates Inc., Red Hook, NY, USA.

Gaye, Babacar and Aziguli Wulamu. 2019. Sentimental analysis for online reviews using machine learning algorithms.

Hemmatian, Fatemeh and Mohammad Karim Sohrabi. 2017. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, pages 1–51.

Hoadley, Bruce. 2020. Score engineered logistic regression. *arXiv preprint arXiv:2003.00958*, pages 4–5.

Hossin, Mohammad and MN Sulaiman. 2015. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1.

Kwon, O, Junbeom Kim, Kwang-Ho Choi, Yeonhee Ryu, and Ji-Eun Park. 2018. Trends in deqi research: a text mining and network analysis. *Integrative Medicine Research*, 7.

Ma, Yukun, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Thirty-second AAAI conference on artificial intelligence*.

Maas, Andrew L, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150, Association for Computational Linguistics.

Mesnil, Grégoire, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.

Miller, Stuart. 2019. Tomatopy - rotten tomatoes scraper. `https://github.com/sjmiller8182/tomatopy`.

Morde, Vishal and Venkat Setty, Anurag. 2019. Xgboost algorithm: Long may she reign! https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

Palkar, Raj K, Kewal D Gala, Meet M Shah, and Jay N Shah. 2016. Comparative evaluation of supervised learning algorithms for sentiment analysis of movie reviews. *International Journal of Computer Applications*, 142(1):20–26.

Pouransari, Hadi and Saman Ghili. 2014. Deep learning for sentiment analysis of movie reviews. *Technical report, Stanford University, Tech. Rep.*

Silva, Fernanda B., Rafael de O. Werneck, Siome Goldenstein, Salvatore Tabbone, and Ricardo da S. Torres. 2018. Graph-based bag-of-words for classification. *Pattern Recogn.*, 74(C):266–285.

Tharwat, Alaa. 2018. Classification assessment methods. *Applied Computing and Informatics*.

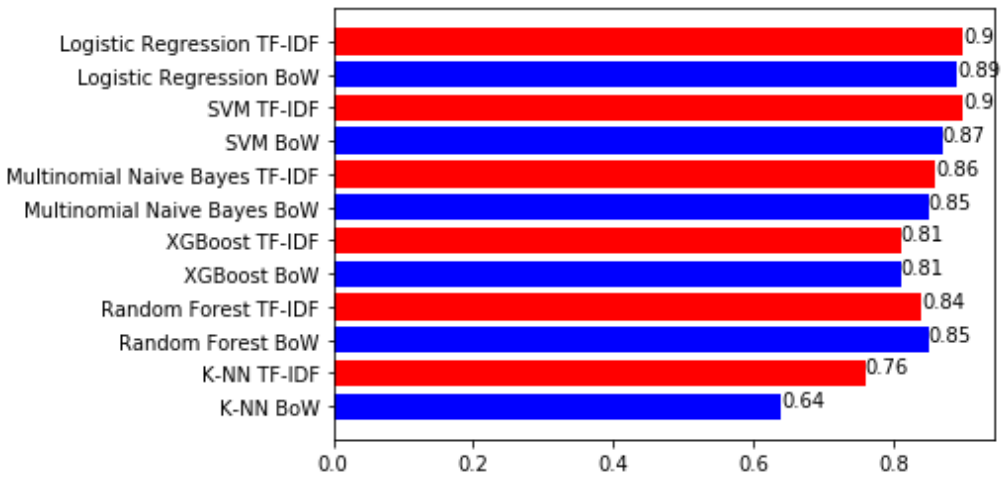
Zhang, Weifeng, Hua Hu, Haiyang Hu, and Jinglong Fang. 2019. Semantic distance between vague concepts in a framework of modeling with words. *Soft Computing*, 23(10):3347–3364.

Zhou, Linda. 2018. How to build a better machine learning pipeline. https://www.datanami.com/2018/09/05/how-to-build-a-better-machine-learning-pipeline/.

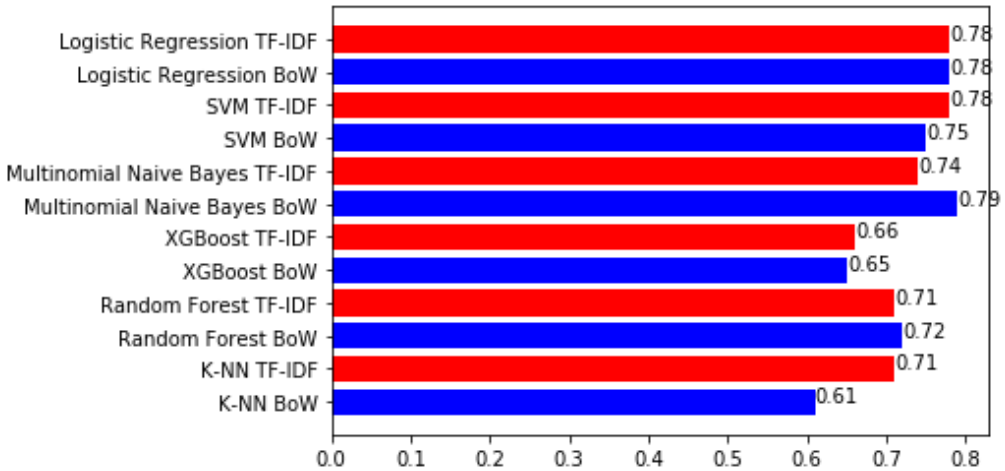
Zhou, Peng, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

1 Appendix A

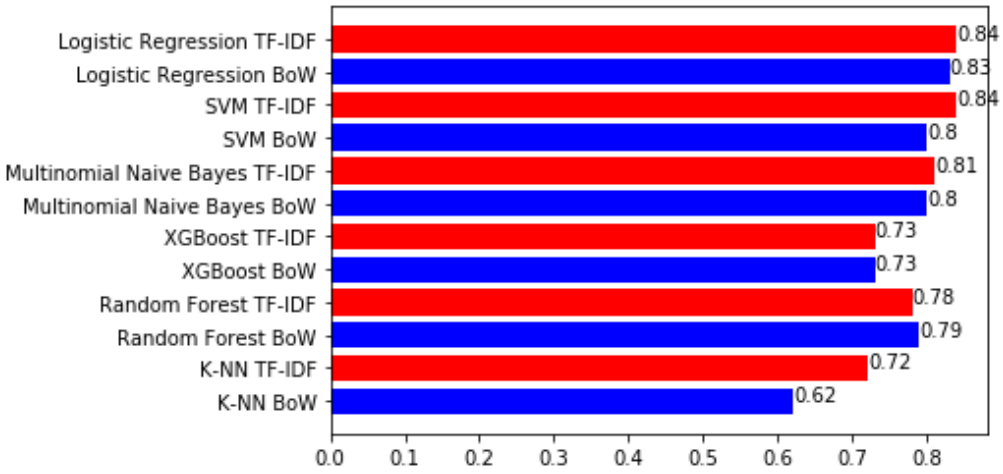
1.1 Bag of Words (BoW) vs Term Frequency - Inverse Document Frequency (TF-IDF) Models for IMDB



1.2 Bag of Words (BoW) vs Term Frequency - Inverse Document Frequency (TF-IDF) Models for Rotten Tomatoes

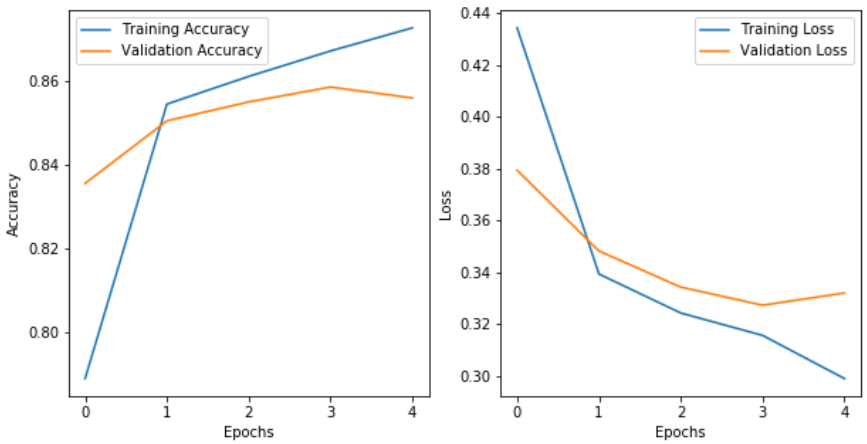


1.3 Bag of Words (BoW) vs Term Frequency - Inverse Document Frequency (TF-IDF) Models for the Merged Dataset

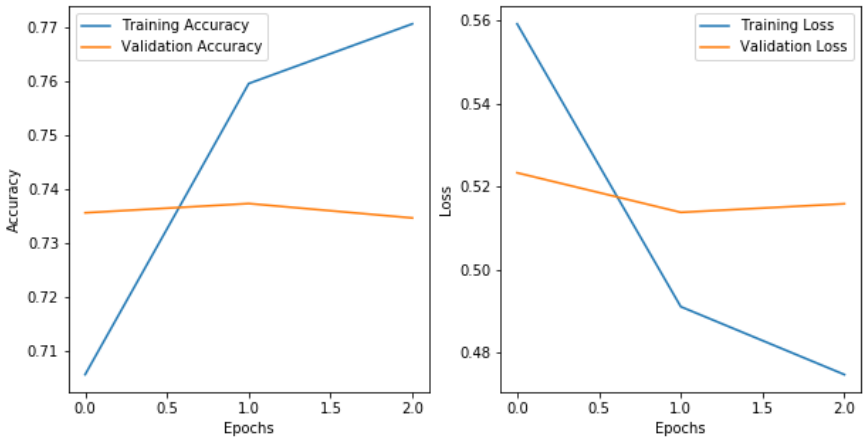


2 Appendix B

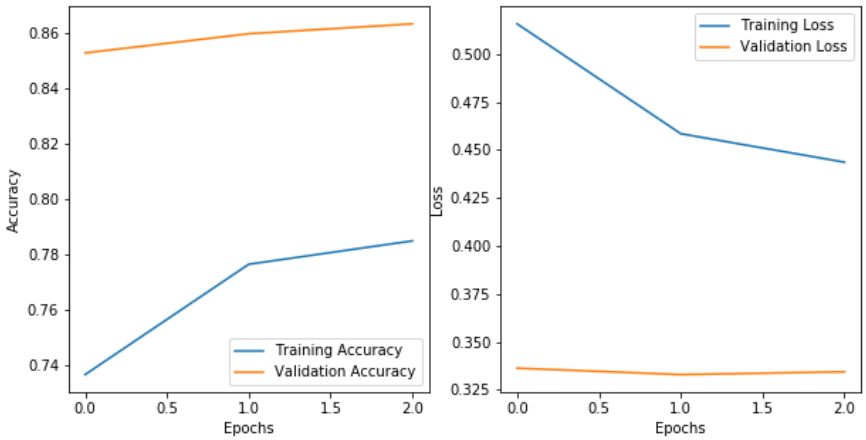
2.1 Accuracy - Loss vs Epochs for the Bidirectional Long Short-Term Memory (Bid-LSTM) Model (IMDB)



2.2 Accuracy - Loss vs Epochs for the Bidirectional Long Short-Term Memory (Bid-LSTM) Model (Rotten Tomatoes)

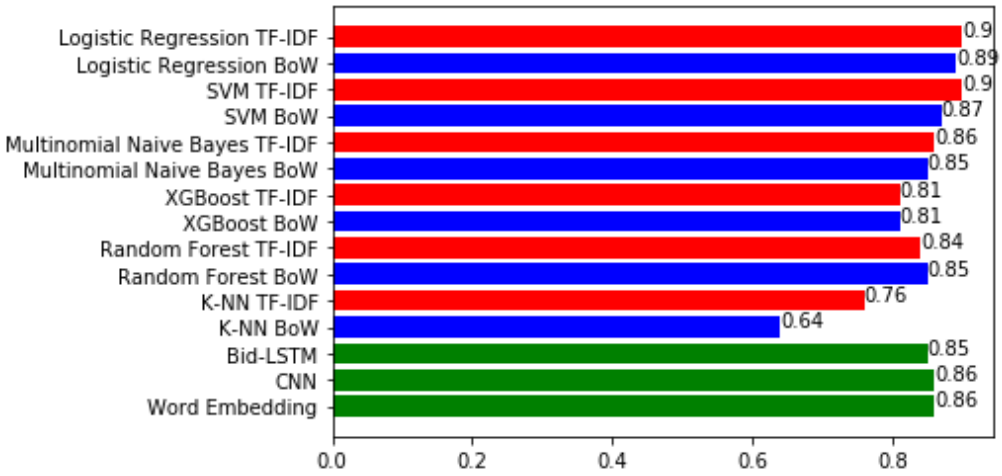


2.3 Accuracy - Loss vs Epochs for the Bidirectional Long Short-Term Memory (Bid-LSTM) Model (Merged Dataset)

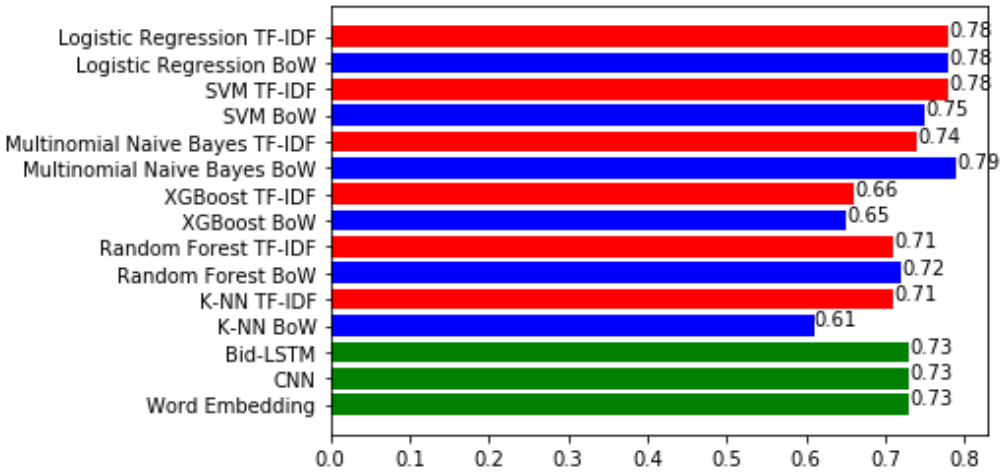


3 Appendix C

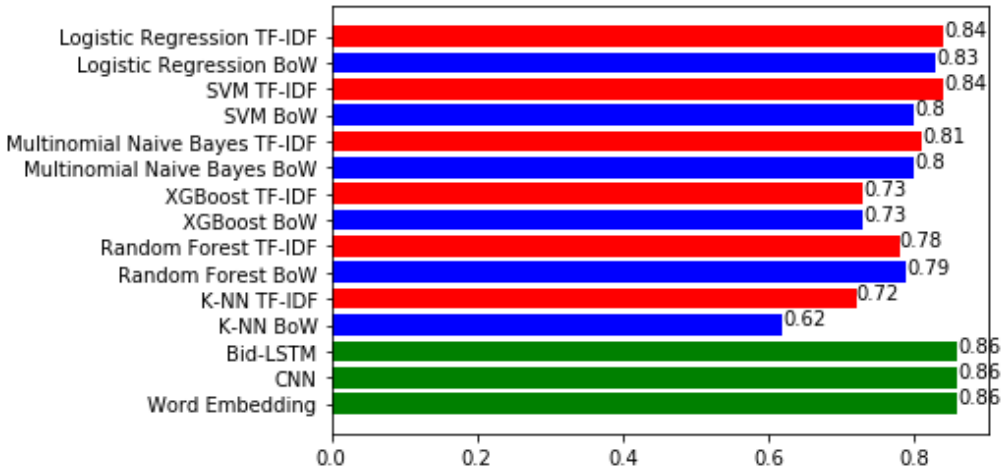
3.1 All Models for IMDB



3.2 All Models for Rotten Tomatoes

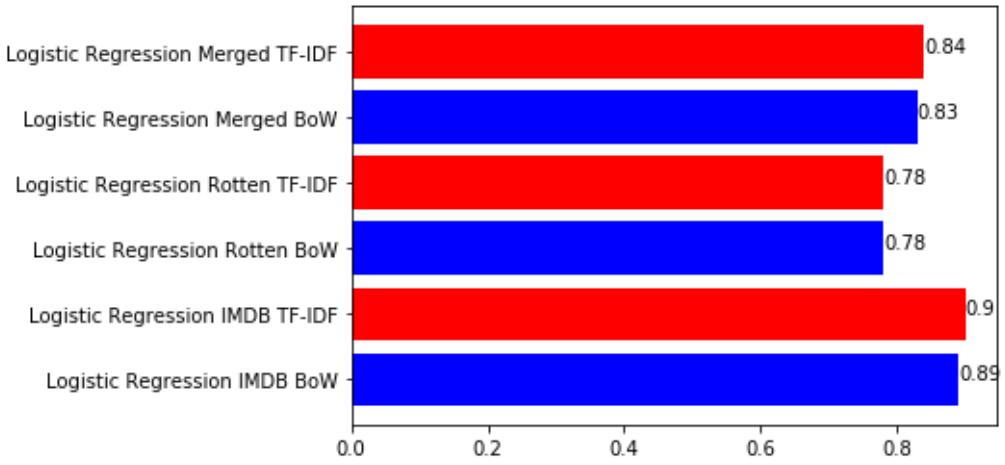


3.3 All Models for the Merged Dataset

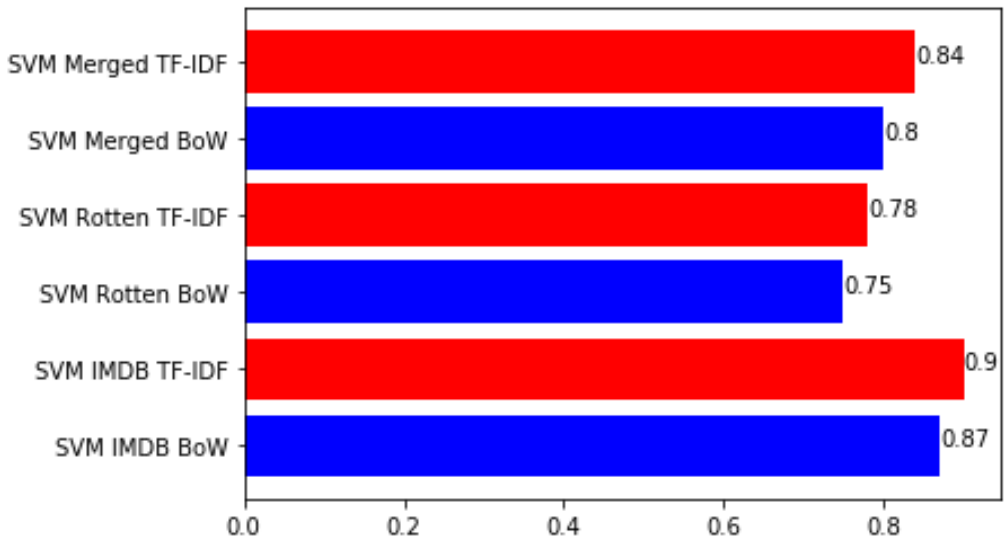


4 Appendix D

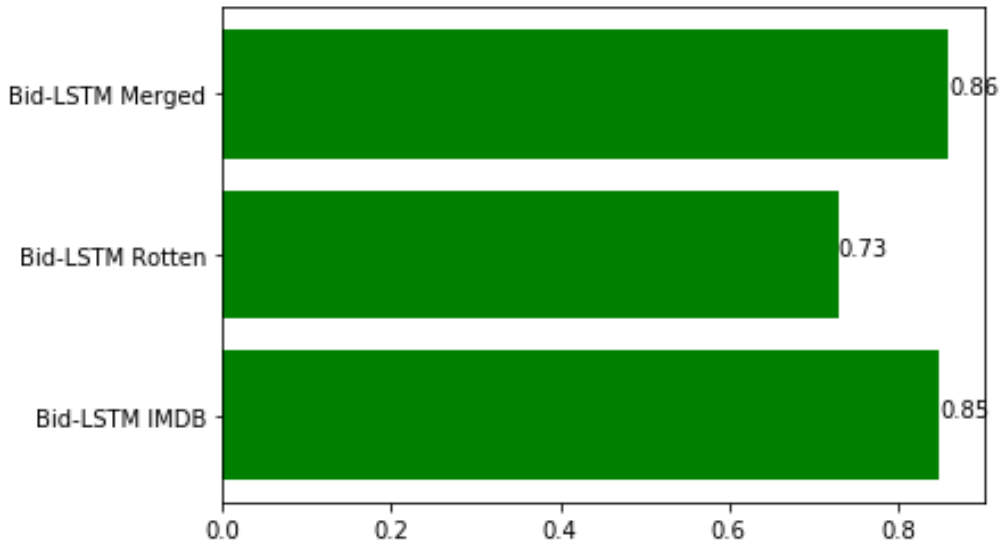
4.1 Logistic Regression Models



4.2 Support Vector Machines (SVM) Models

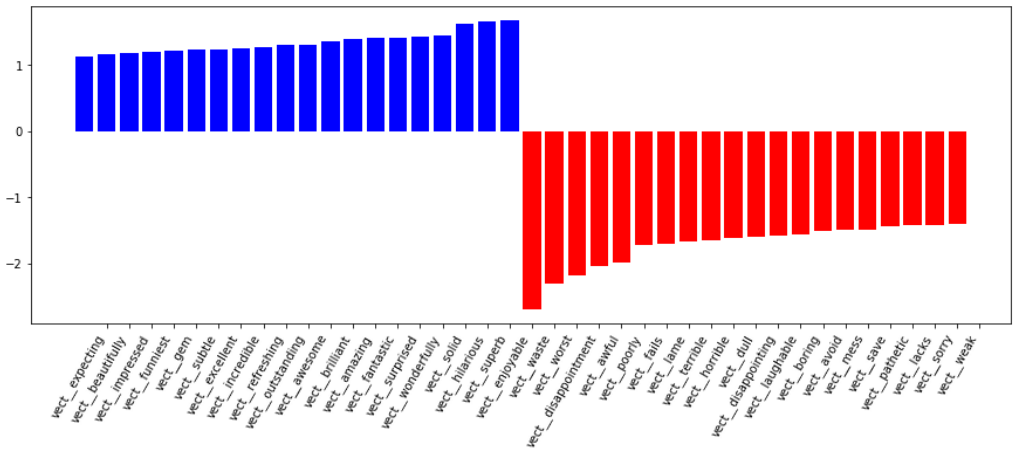


4.3 Bidirectional Long Short-Term Memory (Bid-LSTM) Models

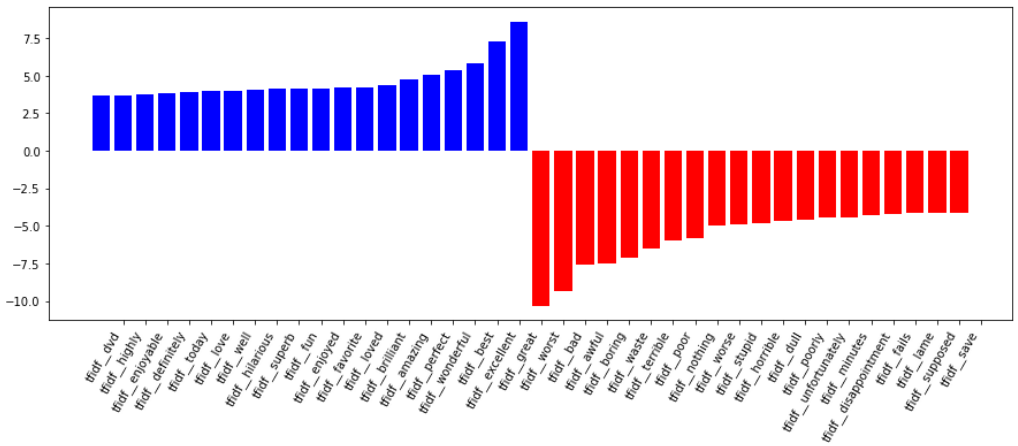


5 Appendix E

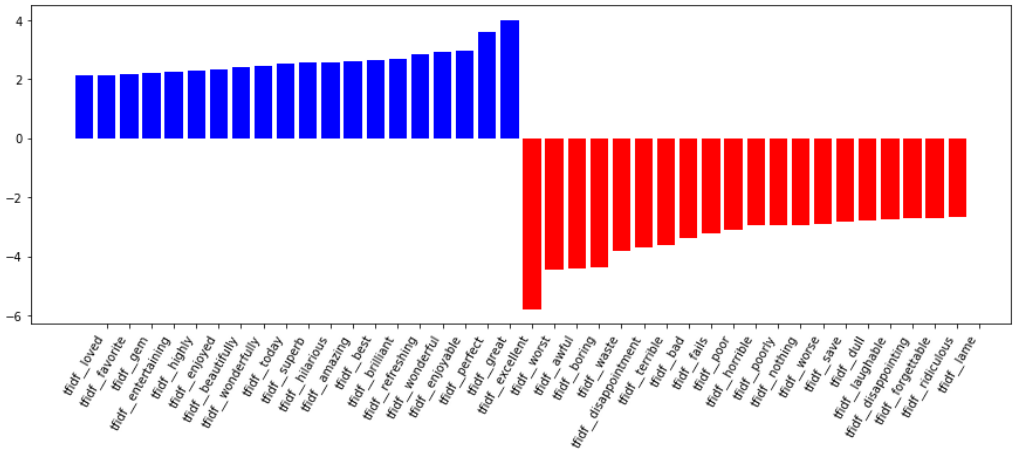
5.1 Feature Importance - Logistic Regression Bag of Words (BoW) Model (IMDB)



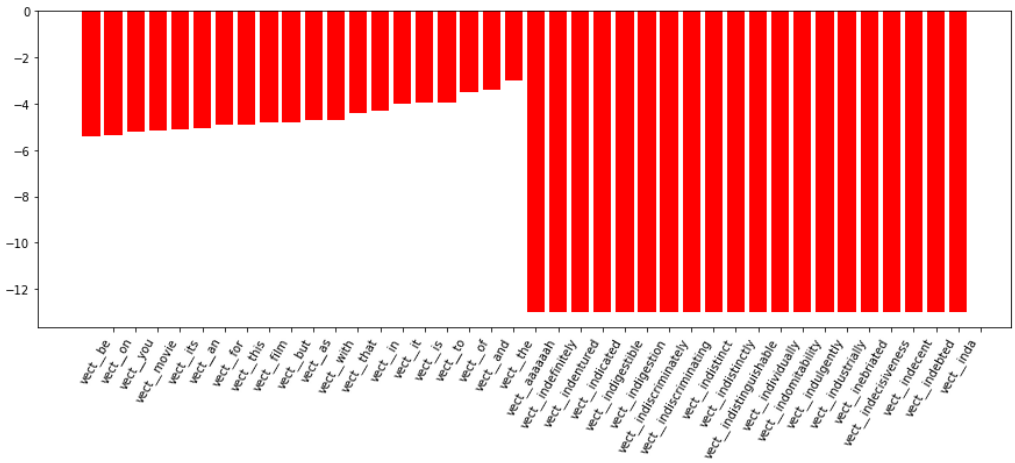
5.2 Feature Importance - Logistic Regression Term Frequency - Inverse Document Frequency (TF-IDF) Model (IMDB)



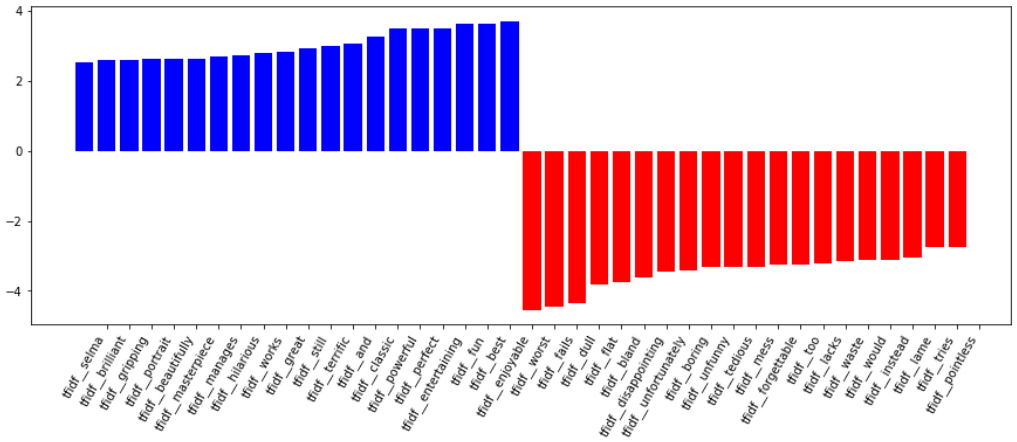
5.3 Feature Importance - Support Vector Machines (SVM) Term Frequency - Inverse Document Frequency (TF-IDF) Model (IMDB)



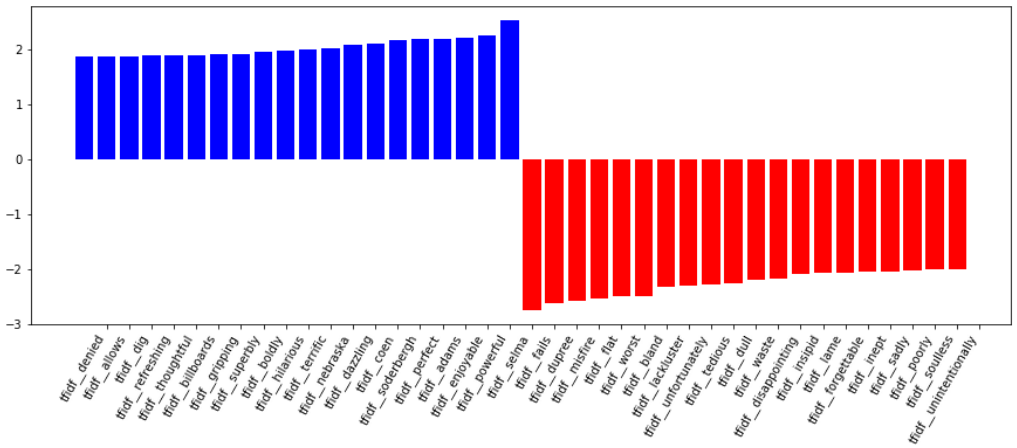
5.4 Feature Importance - Multinomial Naive Bayes Bag of Words (BoW) Model (Rotten Tomatoes)



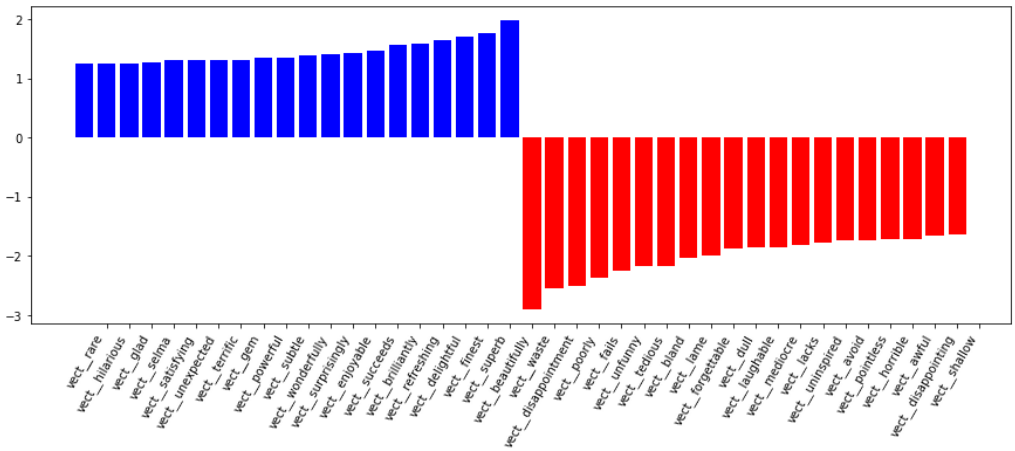
5.5 Feature Importance - Logistic Regression Term Frequency - Inverse Document Frequency (TF-IDF) Model (Rotten Tomatoes)



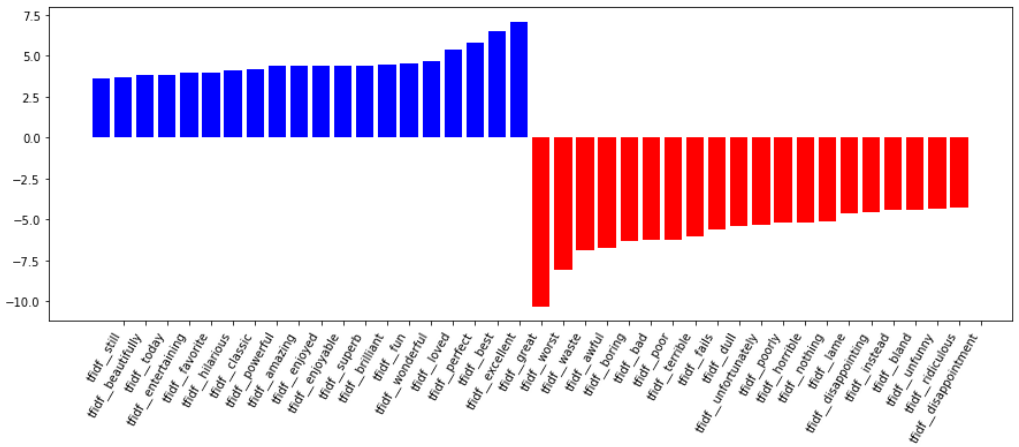
5.6 Feature Importance - Support Vector Machines (SVM) Term Frequency - Inverse Document Frequency (TF-IDF) Model (Rotten Tomatoes)



5.7 Feature Importance - Logistic Regression Bag of Words (BoW) Model (Merged Dataset)



5.8 Feature Importance - Logistic Regression Term Frequency - Inverse Document Frequency (TF-IDF) Model (Merged Dataset)



5.9 Feature Importance - Support Vector Machines (SVM) Term Frequency - Inverse Document Frequency (TF-IDF) Model (Merged Dataset)