Sentiment Analysis and Text Summarization of Reviews

Sudheeksha Garg
Aditya Jayanti
Ishan Shah

ABSTRACT

Sentiment analysis and text summarization are two of the most exciting areas of natural language processing. In this paper, the authors have demonstrated web scraping techniques for data collection from movie reviews aggregation website - IMDB. The authors have showcased the comparative performance of naive bayes, logistic regression and support vector machine algorithms for sentiment analysis of movie reviews. Logistic regression displayed the highest accuracy amongst the three while the other two models display a comparable accuracy. Additionally, two different extractive text summarization methods have been implemented in this paper. Tf-idf and textrank algorithms were used to obtain short summaries from the movie reviews.

Keywords: sentiment analysis, naive bayes, text summarization, tf-idf, textrank

1. INTRODUCTION

Sentiment analysis and text summarization are an integral part of Natural Language Processing (NLP). The former helps in determining the overall expression of the text to classify the information into positive, neutral and negative classes and consequently informed decisions can be made based upon the subjective results. In recent times, the extensive use of sentiment analysis has been instrumental in obtaining the public opinion for various studies in product marketing, politics, and social media amongst other subjects. Text summarization is another area of Natural Language Processing which filters large amounts of text and reduces it to the most significant information for the reader. This process results in effective evaluation of large texts and helps the reader identify the usefulness of the same.

Significantly, President Obama's campaign used sentiment analysis to identify the voter's attitude and devise strategies for raising donations. They were successful in their efforts as they raised more than US$1 billion for the presidential campaign in 2012. Day over day, we see new use-cases that effectively employ results about user sentiments from analyses of the texts to drive the decision-making process.

With the abundance of data around us, news sources like 'Inshorts' have revolutionized the process of news generation using text summarization techniques to enhance newsletters with a stream of summaries rather than a collection of links and limit the number of words to sixty, making it convenient for mobile users to receive a concise and meaningful summary.

The paper has been organized as follows: Section 1.1 and Section 2 explains the process of collecting movie reviews through web scraping strategies. The dependencies and python libraries required for this project are detailed in Section 2.1. Section 3 provides a brief background on current research and various methods of implementations with respect to sentiment analysis and text summarization. In Section 4, the results and analysis for sentiment analysis and text summarization of IMDB movie reviews are discussed. Conclusions and future work have been elaborated in Section 5.

1.1 Data Description

For this project, we initially planned to use reviews collected from several movie websites by using the official API's of IMDB and Rotten Tomatoes or through a web scrapping process if the official APIs posed us with any restrictions. The data obtained through web scraping or the official APIs wouold potentially contain noise such as hyperlinks, duplicate values, and metadata which would be removed using data cleaning techniques. We shall improve the data quality using a regular expression library in Python and obtain a consistent representation of the movie reviews which can be used in the learning algorithms. We shall apply the data cleaning methods on the data of five recent movies and store the reviews in a JSON file which would streamline the subsequent processes.

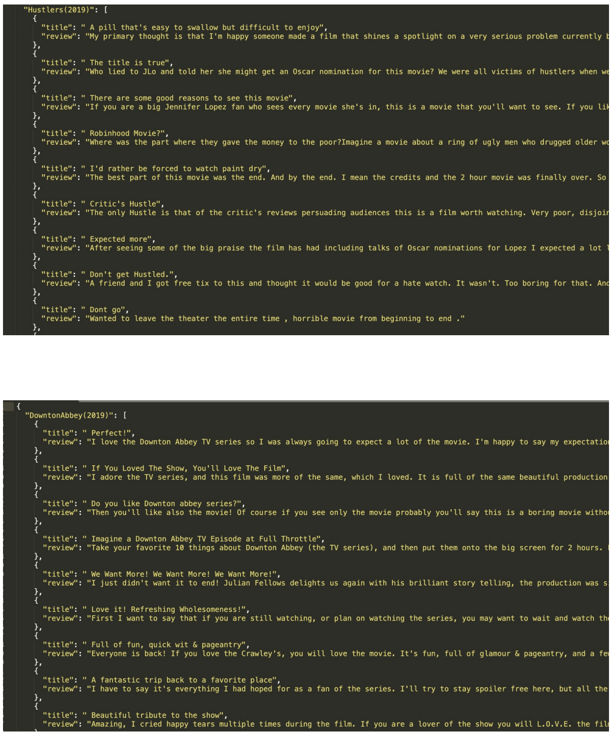
Additionally, if there is a restriction on using the API for IMDB client and Rotten Tomatoes, we plan to overcome this problem using BeautifulSoup. BeautifulSoup is a Python library which is commonly used to extract data from web pages. In addition, we shall use Selenium to automate the process of data gathering and reduce the human intervention in collection of the required data from the web pages. This method shall require a comprehensive data cleaning process as compared to one which would be required for data obtained from the official API. The data cleaning and preparation process would entail handling HTML tags, eliminating hyperlinks and CSS data, and extracting only the data pertaining to the movie title and reviews using regular expressions.

In the next section, we have explained the process of data extraction and give an overview of the structure of the data.

2. DATA EXTRACTION

IMDB is one of the most comprehensive sources of movie and TV show reviews on the web. The database consists of more than 6 million titles and approximately 83 million registered users. For the purpose of the project, we shall automate the process of obtaining the movie reviews from IMDB using Beautiful Soup and Selenium. Beautiful Soup helps us extract the review texts from the web page while Selenium assists in driving the browsing activity of visiting the IMDB web page for each movie under consideration. By visiting every movie page, we can obtain the reviews and assemble them in a JSON file.

While initializing the process of the data collection, we made a formal request to gain access to the Rotten Tomatoes data using their API. However, we were not able to obtain the permission and hence, we will use data obtained from IMDB throughout the course of this project.



```
  "Hustlers(2019)": [
    {
      "title": " A pill that's easy to swallow but difficult to enjoy",
      "review": "My primary thought is that I'm happy someone made a film that shines a spotlight on a very serious problem currently b
    },
    {
      "title": " The title is true",
      "review": "Who lied to JLo and told her she might get an Oscar nomination for this movie? We were all victims of hustlers when we
    },
    {
      "title": " There are some good reasons to see this movie",
      "review": "If you are a big Jennifer Lopez fan who sees every movie she's in, this is a movie that you'll want to see. If you lik
    },
    {
      "title": " Robinhood Movie?",
      "review": "Where was the part where they gave the money to the poor?Imagine a movie about a ring of ugly men who drugged older wo
    },
    {
      "title": " I'd rather be forced to watch paint dry",
      "review": "The best part of this movie was the end. And by the end. I mean the credits and the 2 hour movie was finally over. So
    },
    {
      "title": " Critic's Hustle",
      "review": "The only Hustle is that of the critic's reviews persuading audiences this is a film worth watching. Very poor, disjoin
    },
    {
      "title": " Expected more",
      "review": "After seeing some of the big praise the film has had including talks of Oscar nominations for Lopez I expected a lot l
    },
    {
      "title": " Don't get Hustled.",
      "review": "A friend and I got free tix to this and thought it would be good for a hate watch. It wasn't. Too boring for that. And
    },
    {
      "title": " Dont go",
      "review": "Wanted to leave the theater the entire time , horrible movie from beginning to end ."
    },
```

```
{
  "DowntonAbbey(2019)": [
    {
      "title": " Perfect!",
      "review": "I love the Downton Abbey TV series so I was always going to expect a lot of the movie. I'm happy to say my expectatio
    },
    {
      "title": " If You Loved The Show, You'll Love The Film",
      "review": "I adore the TV series, and this film was more of the same, which I loved. It is full of the same beautiful production
    },
    {
      "title": " Do you like Downton abbey series?",
      "review": "Then you'll like also the movie! Of course if you see only the movie probably you'll say this is a boring movie witho
    },
    {
      "title": " Imagine a Downton Abbey TV Episode at Full Throttle",
      "review": "Take your favorite 10 things about Downton Abbey (the TV series), and then put them onto the big screen for 2 hours.
    },
    {
      "title": " We Want More! We Want More! We Want More!",
      "review": "I just didn't want it to end! Julian Fellows delights us again with his brilliant story telling, the production was s
    },
    {
      "title": " Love it! Refreshing Wholesomeness!",
      "review": "First I want to say that if you are still watching, or plan on watching the series, you may want to wait and watch th
    },
    {
      "title": " Full of fun, quick wit & pageantry",
      "review": "Everyone is back! If you love the Crawley's, you will love the movie. It's fun, full of glamour & pageantry, and a fe
    },
    {
      "title": " A fantastic trip back to a favorite place",
      "review": "I have to say it's everything I had hoped for as a fan of the series. I'll try to stay spoiler free here, but all the
    },
    {
      "title": " Beautiful tribute to the show",
      "review": "Amazing, I cried happy tears multiple times during the film. If you are a lover of the show you will L.O.V.E. the fil
```

Figure 1: Example of data extracted

2.1 Installation and Usage

For this project, the main Python packages that need to be installed are BeautifulSoup, Selenium, and web drivermanager. In addition to these packages, a Chrome Driver needs to be installed depending on the operating system and the Chrome version. We have written a Python script that automates the process of retrieving the data required for our project. Data must be cleaned before any analysis can be performed and for the purpose of this project, we do not need the complete information about the movie reviews and hence, we remove the inessential text from the extracted data. We have further cleaned the data to eliminate Unicode and Line Break characters and then stored it in JSON format that consists of key-value pairs, where the key represents movie name and value describes a list of reviews along with the title of the review. In total, we collected approximately 10,000 reviews for five different movies.

3. BACKGROUND

3.1 Sentiment Analysis

Sentiment analysis plays a crucial role in applications ranging from social media marketing to analyzing patient feedback [7]. It uses natural language processing (NLP) and text analysis to derive subjective knowledge which can be used to enhance the quality of a product or service. In general, it helps to determine the consensus about the item of interest which could either be positive, negative or neutral. The initial step in sentiment analysis requires the identification of objective and subjective components in a text. The former contains factual information and the latter describes sentiments in the form of adjectives [4]. Sentiment analysis can be performed at [4] three levels namely:

- Document: Determining polarity of the entire document.
- Sentence: Each sentence in the document is associated with a polarity.
- Phrase: Analysis and classification of phrases into positive, negative or neutral sentiments.

There are various techniques to perform sentiment classification on reviews. The two main approaches described in a survey [4] are lexicon or knowledge-based and machine learning approach. This sentiment classification was performed using various Natural Language Processing (NLP) pre-processing tasks [4] such as N-grams, Part of Speech (POS) tagging, stemming, stop words, conjunction, and negation handling. One form of data visualization of text extracted from review-aggregation websites is through word clouds. It is a popular method used to visualize a text corpus [10] to understand frequently used words. The font size is determined using the frequency of occurrence of a word. Figure 2 represents a word cloud constructed using reviews obtained from IMDB. Words like ”loved”, ”great”, ”good”, ”brilliant”, ”wonderful”, and ”perfect” indicate that on average the audience had positive feedback for the movie ’Downton Abbey’.

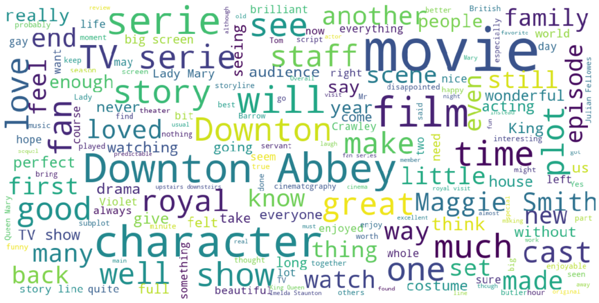


Figure 2: Word cloud for the movie Downton Abbey.

Some of the most popular machine learning algorithms to perform sentiment classification are support vector machine (SVM), naive bayes, and logistic regression. However, a novel approach introduced in [2] for sentiment analysis was found to have greater accuracy compared to naive bayes, SVM, and logistic regression. Latent Dirichlet Allocation (LDA) is a probabilistic topic modeling method that is composed of two processes namely the generative process and inference. While the former process is performed to generate the document using known values of word-topic probability, the latter determines the probabilistic values of word-topic of known documents [2]. In this paper, we have performed sentiment analysis using three different approaches on each review in a collection of reviews for a specific movie to determine whether the audience liked or disliked the movie. The following section discusses different techniques in relation to text summarization.

3.2 Text Summarization

In the U.S, on an average, seven hundred English-language films are released every single year [12] this led to the growth of review-aggregation websites for films and television (e.g., IMDB, Rotten Tomatoes, etc.). These websites provide a platform for internet users to submit comments and reviews about movies and television shows apart from providing information related to show times, theatres, release dates etc. Using this platform, users can make an informed decision about their movie viewing choices and browse information about movies and the movie artists.

The opinion of people about a product or a service is of great importance to companies, because it gives them the ability to monitor public perception of their products which can be used to make decisions having direct impact on the profitability of the organizations. [8]. However, the process of reviewing information for decision making [8] is not only time-consuming but also challenging for the user as it the abundance of reviews can be overwhelming for most people. Most movie review aggregation services allow the users to rate the movie on a numeric scale. On IMDB, the star ratings which are denoted by a scale of 10, the users can write a review and rate the movie as desired. However, the movie ratings usually fail to relay the plot or the details of the movie to the consumer. The only information one can retrieve from the rating is if the movie is good, bad or average. In a recent survey, the researchers discovered that consumers trust content of the review rather than a overall rating assigned to it [1]. Additionally, such ratings and reviews not only help the viewers, but the producers can also estimate the financial returns of their productions.

Text summarization is the process of representing texts spanning few sentences to large documents in a short and summarized form. In today's world, where the consumer is inundated with information, text summarization techniques have proven to be a blessing to those who seek to understand what large pieces of text aim to convey. There are two approaches for text summarization and they can be [3] classified as follows:

- Extractive: In this approach, the ranking algorithm determines the most significant sentences in the text without modifying the original content. Different approaches can be used to rank the sentences and then the sentences which represent the whole text are used to form the summary. This algorithm is language agnostic as compared to the abstractive method.
- Abstractive: As compared to the extractive method, this is a more sophisticated approach which generates concise phrases in line with how humans would summarize a piece of text. However, this approach is more difficult to implement as it requires understanding the language, the semantics.

In this section, we will discuss about the implementation of our extractive text summarization system. Text summarization systems are used to identify significant sentences from a source ranging from a single document to a cluster of related documents, to form a summary [3]. The procedures within a text summarization system can be broadly classified into three different steps, as represented in figure 3. In the initial stage, a topic representation approach is used to transform a piece of text into an intermediate representation. Numerous techniques use intermediate representations, few of which are sentence scoring based on word frequency, term frequency-inverse document frequency (tf-idf), etc [5]. The second stage involves assigning a score to each sentence using stochastic or machine learning techniques. Some of the factors that affect the scoring metric are the context of the information and the type of input document such as web page, email and news articles. Finally, the text summarization process concludes with the selection of a subset of the most significant sentences using an iterative greedy algorithm to generate a summary. [5].

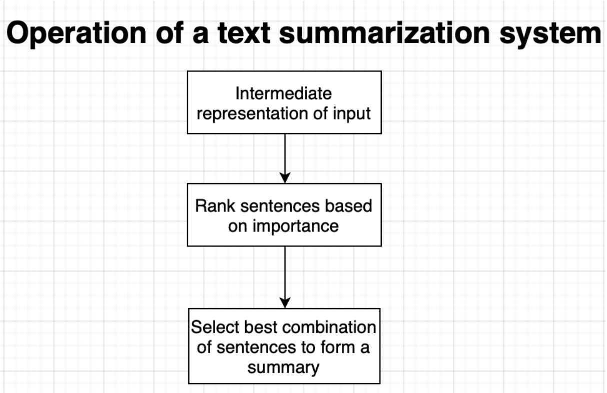

Operation of a text summarization system
Intermediate representation of input
Rank sentences based on importance
Select best combination of sentences to form a summary

Figure 3: Description of the tasks performed by a text summarizer.

Using the knowledge obtained from analyzing the operation of a text summarization system, we investigated various approaches for text summarization and decided to apply a few of these methods to determine the best text summarizing technique. Some of these methods described in [5] are:

- Scoring sentences based on word probability: In this method, the probability of a word is computed as a ratio between the number of occurrences of a word to the total number of words in the input document [5, 11].
- Term Frequency-Inverse Document Frequency: tf-idf consists of a combination of two algorithms term frequency and inverse documents frequency. In this approach, a common stop word list is derived through the NLTK python library and is eliminated from the input document. The score used to represent the term is the product of term frequency and the number of documents that contain that term [5].
- Latent Semantic Analysis (LSA): This is an unsupervised learning approach which involves building an n*m matrix where each row represents a word in the input and each column represents a sentence in the input [11]. The value in the matrix is zero if the sentence does not contain the term and otherwise it is equal to the value computed by tf-idf.

Another approach based on machine learning and SentiWordNet to analyze hotel reviews is presented in [8]. In this work, the performance of various classifiers such as naive bayes, support vector machines and decision trees was compared with SentiwordNet. It was observed that Naive Bayes results in an accuracy of greater than 85% [8]. Upon reviewing previous research in sentiment analysis and text summarization, we implement methods ranging from traditional methods to machine learning in our project for movie reviews obtained through web scraping from IMDB. In the next section, we will discuss the results obtained by implementing a few of the methods above.

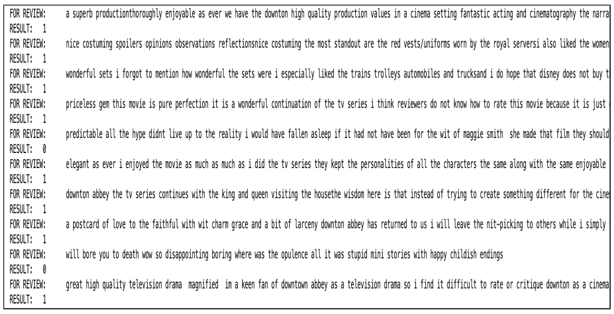
4. RESULTS

This section describes the experimental setup, results and analysis of the methods used to implement sentiment analysis and text summarization of movie reviews.

4.1 Sentiment Analysis

For this project, we used sentiment analysis to classify each IMDB review into positive, negative, or neutral sentiments. A compilation of IMDB reviews was used to train the classifier. Each review has labels 1 or 0 where one is a positive review, and zero is a negative review. To prepare the raw data for sentiment analysis, we implemented some procedures to clean the data. Using Regular Expressions, data is cleaned by replacing HTML tags with empty strings, converting numeric numbers (0-9) to number in word, and replacing any punctuation ('.', ';', ':', '!', '?') with an empty string. All the text is converted to lower case and the stop words are removed as well. This step is extremely important as it forms the basis for rest of implementation. A minor error like not removing stop words can lead to the most discriminating words being the stop words. In the next step, each review is converted into a numeric representation, which is also known as vectorization. Vectorization is implemented with the help of 'countvectorizer' from Python's sklearn library. The vectorized data is used to classify the reviews using one of the methods mentioned below. Once these models are trained and tested with the above data, the reviews gathered for a movie are predicted as negative or positive using the above classifier. The percentage of negative and positive reviews is displayed. We have implemented three different methods of sentiment analysis techniques in this paper and their results are as follows:

- Logistic Regression: This model provides a good baseline as it is easy to interpret, performs well on sparse datasets, and is very fast compared to other models. This model gives an accuracy of 0.88. In this model, only single word features are used. In order to improve the accuracy of this model, ngrams are used while creating the vectorized data, i.e., instead of just single words, an n-range of words form the model; for example, "didn't like the movie" as opposed to just 'like.' In this model, the n-range is from 1-2 words. They increase the accuracy from 0.88 to 0.89 by helping increase the prediction power of the model. Based on this model, for the movie 'Downton Abbey', there are 84% of positive reviews and 16% of negative reviews. Figure 5 shows the first ten reviews of 'Downton Abbey' and its predicted result. From the figure notice that all the reviews are identified correctly. Also notice figure 6; it shows the confusion matrix of logistic regression. From this figure it can be seen that this model correctly classifies most of the reviews.



```
FOR REVIEW:    a superb productionthoroughly enjoyable as ever we have the downton high quality production values in a cinema setting fantastic acting and cinematography the narra
RESULT:   1
FOR REVIEW:    nice costuming spoilers opinions observations reflectionsnice costuming the most standout are the red vests/uniforms worn by the royal serversi also liked the women
RESULT:   1
FOR REVIEW:    wonderful sets i forgot to mention how wonderful the sets were i especially liked the trains trolleys automobiles and trucksand i do hope that disney does not buy t
RESULT:   1
FOR REVIEW:    priceless gem this movie is pure perfection it is a wonderful continuation of the tv series i think reviewers do not know how to rate this movie because it is just
RESULT:   1
FOR REVIEW:    predictable all the hype didnt live up to the reality i would have fallen asleep if it had not have been for the wit of maggie smith  she made that film they should
RESULT:   0
FOR REVIEW:    elegant as ever i enjoyed the movie as much as much as i did the tv series they kept the personalities of all the characters the same along with the same enjoyable
RESULT:   1
FOR REVIEW:    downton abbey the tv series continues with the king and queen visiting the housethe wisdom here is that instead of trying to create something different for the cine
RESULT:   1
FOR REVIEW:    a postcard of love to the faithful with wit charm grace and a bit of larceny downton abbey has returned to us i will leave the nit-picking to others while i simply
RESULT:   1
FOR REVIEW:    will bore you to death wow so disappointing boring where was the opulence all it was stupid mini stories with happy childish endings
RESULT:   0
FOR REVIEW:    great high quality television drama  magnified  im a keen fan of downtown abbey as a television drama so i find it difficult to rate or critique downton as a cinema
RESULT:   1
```

Figure 4: Screenshot of first 10 reviews of Downtown Abbey and predicted labels using logistic regression

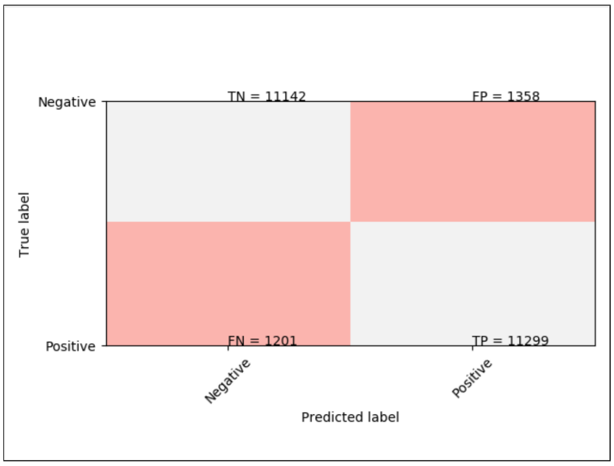


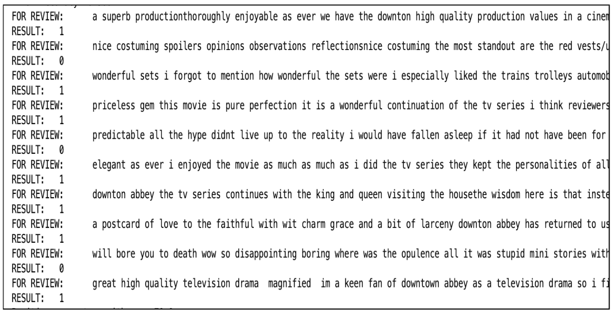
Figure 5: Confusion matrix of sentiment analysis using logistic regression

- Support Vector Machines: This learning algorithm works well for sparse datasets in combination with a linear kernel. A linear SVM model is used with ngrams to get high accuracy of 90%. This model, like logistic regression model, uses n-grams in the range of 1 to 3 words. Based on this model, for the movie 'Downton Abbey,' there are 82% of positive reviews and 18% of negative reviews. The five most discriminating words identified by this model are the same as those determined by logistic regression. However, the accuracy of each word in the support vector machine model is lower compared to that of logistic regression. Figure 7 shows the result of the first ten 'Downton Abbey' reviews and their predicted labels. In this case, one label is misclassified as negative when, in fact, it positive. Also in the notice figure 8, the confusion matrix for support vector machine is displayed. From this figure, it can be seen that this model correctly classifies most of the reviews, but it has more false negatives that logistic regression.

- Naive Bayes: This algorithm which is based on Bayes' Theorem is popularly used in classification tasks. It

Table 1: Five most discriminating positive and negative words

excellent	0.789
perfect	0.667
great	0.648
wonderful	0.552
amazing	0.516
worst	-0.941
awful	-0.866
boring	-0.785
waste	-0.748
bad	-0.728



```
FOR REVIEW:     a superb productionthoroughly enjoyable as ever we have the downton high quality production values in a cinem
RESULT:   1
FOR REVIEW:     nice costuming spoilers opinions observations reflectionsnice costuming the most standout are the red vests/u
RESULT:   0
FOR REVIEW:     wonderful sets i forgot to mention how wonderful the sets were i especially liked the trains trolleys automob
RESULT:   1
FOR REVIEW:     priceless gem this movie is pure perfection it is a wonderful continuation of the tv series i think reviewers
RESULT:   1
FOR REVIEW:     predictable all the hype didnt live up to the reality i would have fallen asleep if it had not have been for
RESULT:   0
FOR REVIEW:     elegant as ever i enjoyed the movie as much as much as i did the tv series they kept the personalities of all
RESULT:   1
FOR REVIEW:     downton abbey the tv series continues with the king and queen visiting the housethe wisdom here is that inste
RESULT:   1
FOR REVIEW:     a postcard of love to the faithful with wit charm grace and a bit of larceny downton abbey has returned to us
RESULT:   1
FOR REVIEW:     will bore you to death wow so disappointing boring where was the opulence all it was stupid mini stories with
RESULT:   0
FOR REVIEW:     great high quality television drama  magnified  im a keen fan of downtown abbey as a television drama so i fi
RESULT:   1
```

Figure 6: Screenshot of first 10 reviews of Downton Abbey and predicted labels using SVM

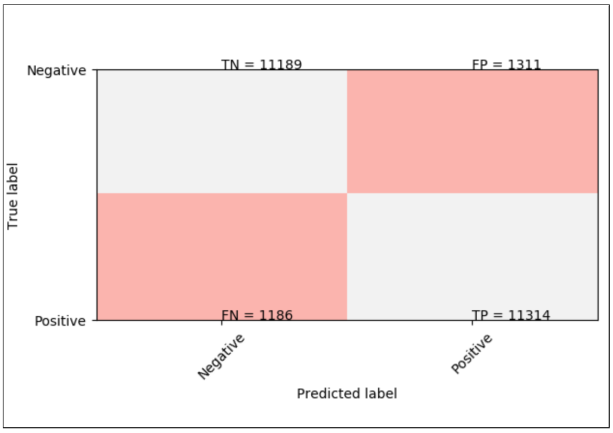
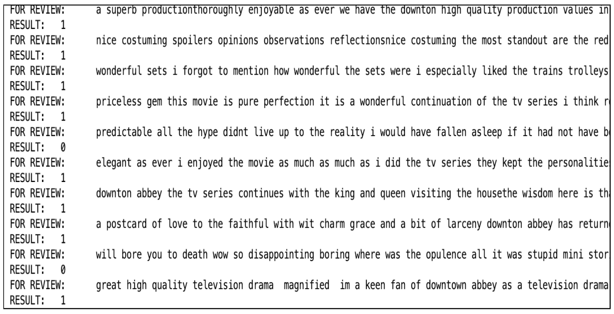


Figure 7: Confusion matrix of sentiment analysis using SVM

assumes that presence of a feature in a class is not related to the presence of the rest of the features. At first, we used Bernoulli's naive bayes but it did not classify the reviews well, so we used multinomial naive bayes instead to classify the reviews better. The difference between the two is that while multinomial Naive Bayes works with occurrence counts, Bernoulli's naive nayes works for binary/boolean features. Using this approach, we obtained results with an accuracy of 0.83. For the 'Downton Abbey' movie, there are 82% of positive reviews and 18% of negative reviews. Based on this model, the top five discriminating words are shown in table 2. Figure 9 shows the screenshot of the first ten movie reviews of 'Downton Abbey,' and its predicted labels. Here, all the labels are classified correctly. Also notice figure 10; it shows the confusion matrix of naive bayes. As you can see in coonfusion matrix, this model has more false negatives and positive than the previous models.



```
FOR REVIEW:     a superb productionthoroughly enjoyable as ever we have the downton high quality production values in
RESULT:   1
FOR REVIEW:     nice costuming spoilers opinions observations reflectionsnice costuming the most standout are the red
RESULT:   1
FOR REVIEW:     wonderful sets i forgot to mention how wonderful the sets were i especially liked the trains trolleys
RESULT:   1
FOR REVIEW:     priceless gem this movie is pure perfection it is a wonderful continuation of the tv series i think r
RESULT:   1
FOR REVIEW:     predictable all the hype didnt live up to the reality i would have fallen asleep if it had not have b
RESULT:   0
FOR REVIEW:     elegant as ever i enjoyed the movie as much as much as i did the tv series they kept the personalitie
RESULT:   1
FOR REVIEW:     downton abbey the tv series continues with the king and queen visiting the housethe wisdom here is th
RESULT:   1
FOR REVIEW:     a postcard of love to the faithful with wit charm grace and a bit of larceny downton abbey has return
RESULT:   1
FOR REVIEW:     will bore you to death wow so disappointing boring where was the opulence all it was stupid mini stor
RESULT:   0
FOR REVIEW:     great high quality television drama  magnified  im a keen fan of downtown abbey as a television drama
RESULT:   1
```

Figure 8: Screenshot of first 10 reviews of Downton Abbey and predicted labels using naive bayes

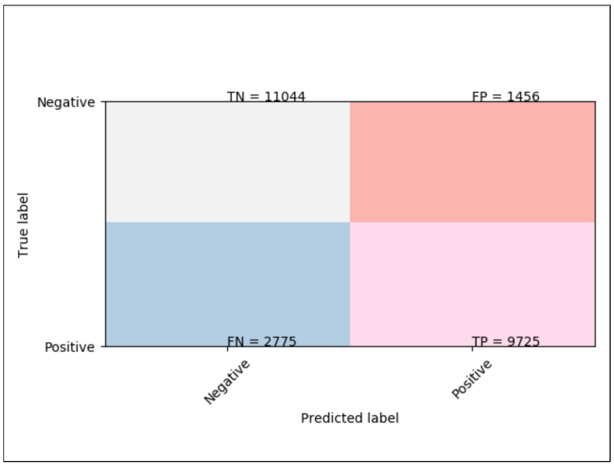


Figure 9: Confusion matrix of sentiment analysis using naive bayes

Table 2: Five most discriminating positive and negative words

film	-0.5779804797916341
movie	-0.5818373519393081
like	-0.8437580336820805
good	-0.9669022625716188
just	-1.0017907662425358
aaaaaaah	-9.433643910491758
aaaaah	-9.433643910491758
aaaahhhhhh	-9.433643910491758
aaaarrgh	-9.433643910491758
aaah	-9.433643910491758

Based on the above results, it can be concluded that logistic regression works the best for this dataset though the results on the above models are almost similar. We believe that logistic regression seems to be the best is because it correctly identifies most of the reviews without being over-fitted. Additionally, we also used the text processing library called 'textblob' to classify the reviews. Using this approach, 94% of the reviews of 'Downton Abbey' were classified as positive and 6% of reviews as negative.

4.2 Text Summarization

We implemented multiple methods of extractive text summarization and evaluated the performance of these approaches by analyzing the meaningfulness of the summaries which were generated for the same movie review. Each of these methods use a unique measure to determine the sentences which are best at summarizing the review. The original text remains untouched and only the key sentences from the original text are used to build the summary. The details of the implementation are explained below.

- textrank - In the first approach, we used the textrank algorithm to obtain the most significant sentences in the review. textrank algorithm is an adaptation of the pagerank algorithm which is popularly used in online search engines for rank assignment of web pages. Textrank uses a concept similar to pagerank but instead of ranking the webpages, we rank the sentences in the corpus.

  In the pagerank algorithm, the rank of the webpage is determined by the probability of reaching the webpage. Using the collection of links between the webpages, the pagerank score is computed and a matrix is used along with the knowledge of the links to determine the reachability of each webpage udner consideration. This method is independent of the language of the text and does not require an understanding of the language. Additionally, since the original phrases remain intact in the summary, contextual understanding of the text is not necessary in this approach.

  The similarity between two sentences in the textrank algorithm is synonymous to the webpage reachability in pagerank algorithm. To obtain the pair of sentences, we split the text containing the movie review into individual sentences. There are numerous methods to obtain individual sentences in a corpus. Some methods are as simple as splitting the text at ".", while the sophisticated approaches such as sentence tokenizer in Python's nltk library can also be used. We opted to split the text at "." for our experiments. We obtained a list of common stop words from Python's nltk library and removed those words before creating the individual sentence vectors. In addition to stop words, extra punctuation and special symbols were removed before computing the frequency of the words in the sentences.

  The cosine similarity scores are used to determine the similarity between each pair of sentences in the text. The similarity between a pair of sentences can be determined using different techniques. The method could be as simple as counting the number of common words over the set of words in the document or using a measure such as cosine similarity. In the text rank algorithm, different similarity scoring functions will potentially provide different results and hence, the summaries will be differ with respect to the similarity measure used in the algorithm. We used the cosine similarity since it uses a normalized representation of the sentences and disregards the length of the individual sentences in its computation. Once the sentences are converted into vectors using the frequency of the words in them, we computed the cosine similarity score for each pair of sentences in the text. The cosine similarity between two vector sentences provides the cosine angle between them. The cosine angle is nothing but a comparison between two sentences in a normalized space.

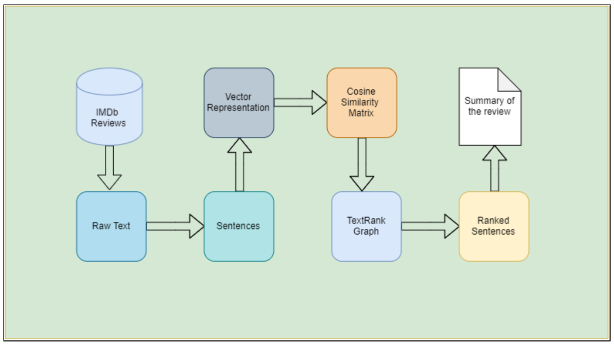


Figure 10: Flowchart of Text Summarization using textrank algorithm

  Subsequently, we used the page rank algorithm to convert the cosine similarity matrix into a graph with sentences as nodes and edges as the similarity scores between them. This graph helped us obtain the top scored sentences in the text and we used those original sentences to form the summary for the movie review.

- Term Frequency-Inverse Document Frequency - In the second approach we used tf-idf, this method involves computing the product obtained from two different algorithms namely term frequency (tf) and inverse document frequency (idf). It determines how important a word is in a collection of documents or corpus and is used in applications involving text mining and information retrieval [13].

  Term frequency is defined as:

  $$tf = \frac{Number\ of\ times\ a\ word\ appears\ in\ a\ document}{Number\ of\ words\ in\ the\ document} \tag{1}$$

  Inverse document frequency is defined as:

  $$idf = log_{10}\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ word}\right) \tag{2}$$

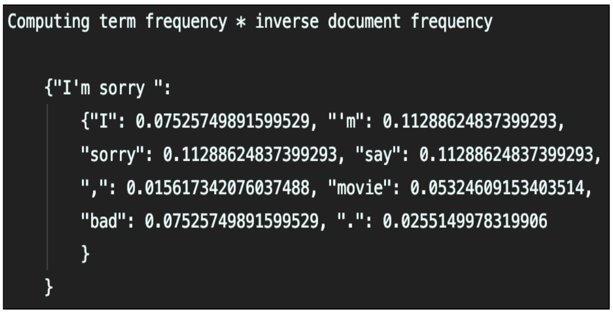
  Term frequency-Inverse document frequency is defined as:

  $$tf - idf(word_i) = tf_i \cdot idf_i \tag{3}$$

  In many cases, reviews may contain characters such as emojis, and hyperlinks these special characters may differ depending upon the operating system. Hence, the first step in this process involves cleaning the data to eliminate non-ASCII and URLs present in the string. Stop words are also eliminated from the string as these words usually have high frequency and do not provide sufficient contextual information required for summarizing the text. Following this, a word tokenizer was used to split strings based on space and punctuation. There are two approaches to convert a word to its root form they are stemming and lemmatization.

In our approach we performed lemmatization using the NLTK library to reduce each word in the list. The benefit of lemmatization is, it considers the context of the text and converts it into a meaningful root form, whereas stemming may often lead to spelling errors [6]. This is performed to group words derived from the same stem. For example, the words 'play' and 'played' are derived from the root 'play'. Ultimately, word frequencies are calculated using only the root words. In our word frequency dictionary, the key represents a sentence and the value represents another dictionary containing frequencies of the root word for that sentence. To simplify the process, we considered maximum sentence length as ten to build the word frequency dictionary [9]. Although considering maximum sentence lengths of up to twenty-five does not alter the results for our data set as perceived through trial and error.

In the next step, we compute term frequency for each sentence in the dictionary using equation (1). In which, the document represents either a collection of reviews or a single review, and a term is described as a word that appears in the document. This is followed by computing inverse document frequency represented in equation (2). A dictionary is created that contains the frequency of occurrence of each word in a document, using which the number of documents containing the word is calculated. The tf-idf value is computed as a product of values associated with term frequency and inverse document frequency. In addition to computing tf-idf values, to generate a summary, a sentence scoring algorithm is required to select the most important sentences based on weighted scores. High weighted scores are the most important and will more likely be a part of the summary. Figure 11 shows an example of the tf-idf dictionary for a movie review, wherein the key represents the sentence and values are the associated lemmatized words.



```
Computing term frequency * inverse document frequency


    {"I'm sorry ":
        {"I": 0.07525749891599529, "'m": 0.11288624837399293,
        "sorry": 0.11288624837399293, "say": 0.11288624837399293,
        ",": 0.015617342076037488, "movie": 0.05324609153403514,
        "bad": 0.07525749891599529, ".": 0.0255149978319906
        }
    }
```

Figure 11: Dictionary representation after computing tf-idf

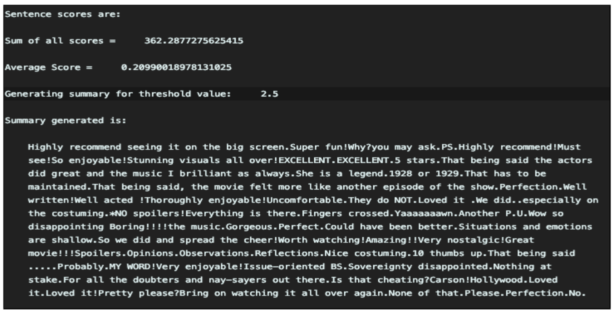
A common scoring metric for identifying the most important sentences using tf-idf values is as shown below:

Scoring metric for identifying most important sentences:

$$score_s = \frac{Sum\ of\ all\ tf - idf\ values_s}{Number\ of\ words\ in\ nested\ dictionary} \tag{4}$$

For example, the score of the word represented in figure. 11 "I'm sorry" is 0.073. Similarly, this is applied to a large corpus or a collection of documents to compute the average score for all the sentences. In addition to calculating average scores, in our project we have also predefined a threshold parameter to control the length of the summary generated. This value is inversely proportional to the length of the summary. Finally, the summary generated contains 'n' sentences that are 's' characters long and all 'n' sentences having a score greater than or equal to the threshold value.

We consider three scenarios to perform text summarization and compare the results, the first approach is to combine all the reviews for a particular movie, the second approach considers a shuffled set of fifty or hundred reviews, and in the final approach we consider one review. Figure 12 is a summary of approximately three-hundred and twenty reviews combined with a threshold value of 2.5 and an average score of 0.21. Although tf-idf does not factor in semantic similarities between words. The summary obtained in figure 12 is comparable to the results obtained in sentiment analysis for the same movie, i.e., most reviewers had a positive feedback for 'Downton Abbey'. In the second approach, we consider fifty movie reviews for text summarization, it is represented in figure 13 with a threshold value of 1.8 and an average score of 0.195. Finally, figure 14 shows the results for a single review with a threshold value of 0.8 and an average score of 0.085.



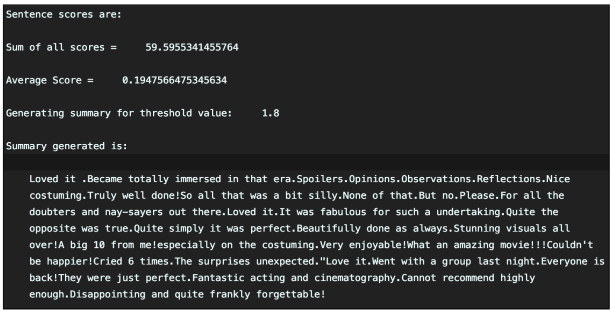
```
Sentence scores are:

Sum of all scores =     362.2877275625415

Average Score =     0.20990018978131025

Generating summary for threshold value:     2.5

Summary generated is:

    Highly recommend seeing it on the big screen.Super fun!Why?you may ask.PS.Highly recommend!Must
    see!So enjoyable!Stunning visuals all over!EXCELLENT.EXCELLENT.5 stars.That being said the actors
    did great and the music I brilliant as always.She is a legend.1928 or 1929.That has to be
    maintained.That being said, the movie felt more like another episode of the show.Perfection.Well
    written!Well acted !Thoroughly enjoyable!Uncomfortable.They do NOT.Loved it .We did..especially on
    the costuming.*NO spoilers!Everything is there.Fingers crossed.Yaaaaaaawn.Another P.U.Wow so
    disappointing Boring!!!!the music.Gorgeous.Perfect.Could have been better.Situations and emotions
    are shallow.So we did and spread the cheer!Worth watching!Amazing!!Very nostalgic!Great
    movie!!!Spoilers.Opinions.Observations.Reflections.Nice costuming.10 thumbs up.That being said
    .....Probably.MY WORD!Very enjoyable!Issue—oriented BS.Sovereignty disappointed.Nothing at
    stake.For all the doubters and nay—sayers out there.Is that cheating?Carson!Hollywood.Loved
    it.Loved it!Pretty please?Bring on watching it all over again.None of that.Please.Perfection.No.
```

Figure 12: Summary of a all movie reviews combined - Downton Abbey(2019)

In conclusion, although tf-idf does not present the most detailed summary, it is comparable to the results obtained in sentiment analysis and easy to implement. The entire process of tf-idf is based on equations (1) and (2). In the next section, we have concluded all of our findings along with the future work.

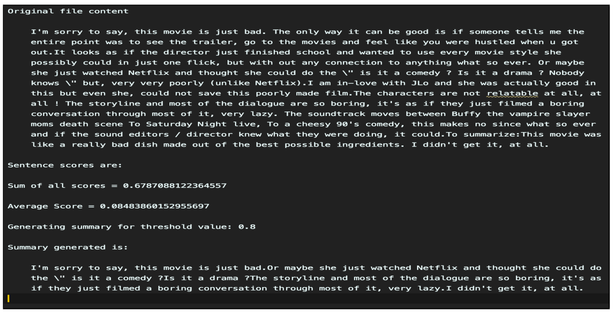
5. CONCLUSION AND FUTURE WORK

In this paper, we illustrated web scraping strategies to obtain raw data from the IMDB web pages. We elaborated the general approach towards data cleaning and preparation which helped us in obtaining data which could be used for



```
Sentence scores are:

Sum of all scores =     59.5955341455764

Average Score =     0.1947566475345634

Generating summary for threshold value:     1.8

Summary generated is:

    Loved it .Became totally immersed in that era.Spoilers.Opinions.Observations.Reflections.Nice
    costuming.Truly well done!So all that was a bit silly.None of that.But no.Please.For all the
    doubters and nay-sayers out there.Loved it.It was fabulous for such a undertaking.Quite the
    opposite was true.Quite simply it was perfect.Beautifully done as always.Stunning visuals all
    over!A big 10 from me!especially on the costuming.Very enjoyable!What an amazing movie!!!Couldn't
    be happier!Cried 6 times.The surprises unexpected."Love it.Went with a group last night.Everyone is
    back!They were just perfect.Fantastic acting and cinematography.Cannot recommend highly
    enough.Disappointing and quite frankly forgettable!
```

Figure 13: Summary of hundred movie reviews combined - Downton Abbey(2019)



```
Original file content

    I'm sorry to say, this movie is just bad. The only way it can be good is if someone tells me the
    entire point was to see the trailer, go to the movies and feel like you were hustled when u got
    out.It looks as if the director just finished school and wanted to use every movie style she
    possibly could in just one flick, but with out any connection to anything what so ever. Or maybe
    she just watched Netflix and thought she could do the \" is it a comedy ? Is it a drama ? Nobody
    knows \" but, very very poorly (unlike Netflix).I am in-love with JLo and she was actually good in
    this but even she, could not save this poorly made film.The characters are not relatable at all, at
    all ! The storyline and most of the dialogue are so boring, it's as if they just filmed a boring
    conversation through most of it, very lazy. The soundtrack moves between Buffy the vampire slayer
    moms death scene To Saturday Night live, To a cheesy 90's comedy, this makes no since what so ever
    and if the sound editors / director knew what they were doing, it could.To summarize:This movie was
    like a really bad dish made out of the best possible ingredients. I didn't get it, at all.

Sentence scores are:

Sum of all scores = 0.6787088122364557

Average Score = 0.08483860152955697

Generating summary for threshold value: 0.8

Summary generated is:

    I'm sorry to say, this movie is just bad.Or maybe she just watched Netflix and thought she could do
    the \" is it a comedy ?Is it a drama ?The storyline and most of the dialogue are so boring, it's as
    if they just filmed a boring conversation through most of it, very lazy.I didn't get it, at all.
```

Figure 14: Summary of a single movie review - Hustlers(2019)

building the various machine learning models. We implemented various methods for sentiment analysis as well as text summarization and compared the results of the various approaches.

For the purpose of sentiment analysis of the movie reviews, we used naive bayes, logistic regression and support vector machines. Logistic regression displayed the highest accuracy amongst the three models. In the future, we can implement stemming and lemmatization which could improve the accuracy of the models. Another area of future work includes performing sentiment analysis using algorithms such as Latent Dirichlet Allocation and compare its accuracy with logistic regression.

We explored the various methods of text summarization in this paper. Since extractive summaries do not modify the original text, abstractive methods could potentially be better at describing a text about any topic. However, there are additional challenges which we might face while obtaining an abstract summary. We not only need to understand the language of the text to generate an abstract summary, but we also need to understand the context of the document.

Extractive summarization proves to be a straight-forward approach with intuitive results in our experiments. We could explore the usage of other similarity scoring functions in the textrank algorithm apart from the cosine similarity score. In the future, we can also extend the work to compare the results of the extractive text summarization with the results of abstractive text summarization.

6. REFERENCES

[1] C. Affairs. Consumers trust content over star ratings. `https://blog.consumeraffairs.com/consumers-trust-content-over-star-ratings/`, 2016. [Online; accessed 28-Nov-2019].

[2] M. F. A. Bashri and R. Kusumaningrum. Sentiment analysis using latent dirichlet allocation and topic polarity wordcloud visualization. In *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, pages 1–5, May 2017.

[3] M. Indu and K. V. Kavitha. Review on text summarization evaluation methods. In *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, pages 1–4, May 2016.

[4] H. Kaur, V. Mangat, and Nidhi. A survey of sentiment analysis techniques. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 921–925, Feb 2017.

[5] A. Nenkova and K. McKeown. *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA, 2012.

[6] S. Prabhakaran. Lemmatization Approaches with Examples in Python. `https://www.machinelearningplus.com/nlp/lemmatization-examples-python/`, 2019. [Online; accessed 01-Dec-2019].

[7] V. Ramanathan and T. Meyyappan. Twitter text mining for sentiment analysis on peopleâĂŹs feedback about oman tourism. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–5, Jan 2019.

[8] V. B. Raut and D. D. Londhe. Opinion mining and summarization of hotel reviews. In *2014 International Conference on Computational Intelligence and Communication Networks*, pages 556–559, Nov 2014.

[9] Raygun. Count Wordsworth. `http://countwordsworth.com/blog/what-is-a-good-average-sentence-length/`, 2013. [Online; accessed 01-Dec-2019].

[10] SumedhKadam. Generating Word Cloud. `https://www.geeksforgeeks.org/generating-word-cloud-python/`, 2019. [Online; accessed 26-Nov-2019].

[11] M. Thaker. Text Summarization Techniques. `https://towardsdatascience.com/comparing-text-summarization-techniques-d1e2e465584e`, 2019. [Online; accessed 28-Nov-2019].

[12] Wikipedia. Cinema of the United States. `https://en.wikipedia.org/wiki/Cinema_of_the_United_States`, 2019. [Online; accessed 27-Nov-2019].

[13] Wikipedia. TFIDF. `https://en.wikipedia.org/wiki/tfidf`, 2019. [Online; accessed 01-Dec-2019].