BIG DATA MOUNTAIN ANALYTICS

Team D6

Advised by: Jagmohan Dutta

LOYALIST COLLEGE IN TORONTO

APRIL 2023
FINAL REPORT

Support By

Developed By

Devasena Papannan - 500201659
Hariprasad Rachamalla - 500201661
Kirti Kashyap - 500201746
Pratik Domadiya - 500199494
Robin Sharma - 500199476
Shivam Patel - 500201461

Table of Contents

1. ABSTRACT

Movie reviews serve as a key metric for judging a movie's performance. Our team has taken a novel approach for sentiment analysis of IMDB (Internet Movie Database) movie reviews using traditional lexicon-based sentiment analysis approach and latest machine learning methods in this study.

The goal of this project is to develop a model that can accurately classify the sentiment of movie reviews as positive or negative. In order to process the data, we begin by removing stop words, stemming, and tokenizing the IMDb movie review dataset. In lexicon-based sentiment analysis, we used SentiWordNet to assign a polarity and subjectivity score to each word in the reviews. Based on those scores of the terms, it will classify the given reviews into two different target classes: Positive review and Negative review. Furthermore, we tested various machine learning techniques such as Naive Bayes, Support Vector Machines, Decision Tree, and Logistic Regression. We assess the model's performance using multiple metrics, including accuracy, precision, recall, and F1 score. Our results reveal that the machine learning method performs best, with an average model’s test accuracy of nearly 88.00% as compared to traditional lexicon-based Sentiment analysis using SentiWordNet (64.00%). Using machine learning methods, we provide a promising solution for sentiment analysis of IMDb movie reviews. The project also discusses the limitations and future directions for improving the performance of the sentiment analysis model. Overall, this project provides insights into the application of machine learning techniques for sentiment analysis of movie reviews and its potential impact in the Movie Productions.

2. INTRODUCTION

Sentiment analysis is an effective technique for comprehending and analyzing the emotions and opinions represented in text data. In the context of examining IMDb movie reviews, sentiment analysis helps us understand the audience's general sentiment toward a specific film, including what aspects they liked or hated. To effectively anticipate the attitude of IMDb movie reviews, this investigation will employ a few natural language processing techniques and machine learning algorithms. A model can be trained to classify each review as good, negative, or neutral by collecting and preprocessing a huge dataset of movie reviews. Several prior research investigated the application of sentiment analysis in the context of movie reviews. Pang and Lee (2008), for example, employed a combination of supervised and unsupervised learning methods to categorize movie reviews as favourable or negative. Kim (2014) used convolutional neural networks to identify movie reviews based on emotion. In this study, we will expand on past research by investigating alternative feature engineering techniques and machine learning algorithms to increase the accuracy of our sentiment analysis model. We may acquire useful insights into the sentiment of IMDb movie reviews and influence future decision-making processes by analyzing the model's performance using appropriate measures. Furthermore, in this project, we will use Streamlit to create a web-based application for deploying a sentiment analysis model for movie reviews. The IMDB movie reviews dataset will be used to train the model, which contains 50,000. To preprocess the text input and turn it into a numerical format that can be fed into a machine learning algorithm, we will utilize the Natural Language Toolkit (NLTK) package. The reviews will then be classified as positive or negative using a Support Vector Machine (SVM) technique. Users will be able to enter a movie review into the deployed application, and the model will classify

it as favourable or bad. Using Streamlit, we constructed an interactive user interface that displays the classification result based on the performance of the model.

3. LITERATURE REVIEW

Sentiment analysis is a well-studied subject in natural language processing, and several prior research employed various algorithms to analyze IMDb movie reviews. We cover some of the important papers that have investigated sentiment analysis of IMDb movie reviews using SentiWordNet (traditional method) and machine learning techniques in this literature review.

3.1 Sentiment Analysis Using Traditional Methods

Sentiment analysis is a process of analyzing and classifying opinions or emotions expressed in text data. In this literature review, we will summarize some of the traditional methods used for sentiment analysis.

Rule-based methods:

Rule-based methods involve creating a set of predefined rules to classify text into different sentiment categories. These rules can be based on specific keywords, linguistic patterns, or syntactic rules. For example, a rule-based method can classify a text as positive if it contains words such as "good," "excellent," or "happy," and negative if it contains words such as "bad," "poor," or "unhappy." Rule-based methods can be effective for simple tasks and domains, but they may not work well for more complex tasks and domains that require understanding the context and nuances of the text. we will summarize some of the recent studies that have used rule-based methods for sentiment analysis.

In a study by Balamurali and Suresh Kumar (2021), a rule-based sentiment analysis approach was used to classify tweets related to COVID-19 into positive, negative, or neutral sentiment categories. The approach involved creating a set of rules based on the presence of specific keywords and emoticons in the tweets. The results showed that the rule-based approach achieved

an accuracy of 86% for the positive sentiment category, 92% for the negative sentiment category, and 78% for the neutral sentiment category.

In a study by Alharbi et al. (2020), a rule-based sentiment analysis approach was used to classify customer reviews of hotels into positive, negative, or neutral sentiment categories. The approach involved creating a set of rules based on the presence of specific keywords, such as adjectives and verbs, and their intensifiers and negations. The results showed that the rule-based approach achieved an accuracy of 79% for the positive sentiment category, 87% for the negative sentiment category, and 76% for the neutral sentiment category.

Lexicon-based methods:

Lexicon-based methods involve using a predefined list of words and their associated sentiment polarity to classify text into different sentiment categories. These lists are called sentiment lexicons, and they can be created using different approaches such as manual annotation, machine learning, or crowdsourcing. For example, the AFINN lexicon assigns a numerical score to each word, indicating its sentiment polarity, and the sentiment polarity of a text can be calculated as the sum of the scores of its words. Lexicon-based methods can be useful for a wide range of domains and languages, but they may not capture the context and subtleties of the text.

Another one of the most popular methods is SentiWordNet Lexical analysis. SentiWordNet is a publicly available lexical resource for sentiment analysis. It is a lexical database that assigns to each synset of WordNet (a large lexical database of English) three sentiment scores: positivity, negativity, and objectivity. These scores range from 0 to 1 and represent the degree of sentiment polarity of the synset.

SentiWordNet was developed by researchers at the Italian National Research Council and has been widely used in research studies for sentiment analysis. The approach involves first mapping words in each text to their corresponding synsets in WordNet, and then calculating the sentiment score for each synset using SentiWordNet. Finally, the sentiment scores of all the synsets in the text are aggregated to obtain an overall sentiment score for the text. Here are some examples of literature reviews on studies that have used SentiWordNet:

In a literature review by Alok et al. (2021) on sentiment analysis of social media data, the authors highlighted several studies that used SentiWordNet to classify sentiment in tweets, reviews, and news articles. The studies reported varying levels of accuracy, with some achieving accuracies of over 80% for sentiment classification.

In a literature review by Kaur and Singh (2020) on sentiment analysis of movie reviews, the authors highlighted several studies that used SentiWordNet as a feature in machine learning models to classify the sentiment of movie reviews. The studies reported accuracies ranging from 75% to 92% for sentiment classification.

However, SentiWordNet has some limitations. For example, it does not consider the context of the words in the text, which can affect the sentiment polarity of the text. Additionally, the sentiment scores in SentiWordNet were assigned based on a simple algorithm that may not capture the complexity of sentiment in natural language.

3.2 Sentiment Analysis Using Machine Learning Methods

Autonomous machine learning is an emerging field in artificial intelligence that aims to automate the entire machine learning process, including data preparation, feature engineering, model selection, hyperparameter tuning, and evaluation. In this literature review, we will summarize

some of the recent research and approaches in autonomous machine learning based movie review sentiment analysis.

Automated Machine Learning (AutoML)

AutoML is a popular approach to autonomous machine learning that has been applied to various tasks, including sentiment analysis. In the paper "AutoML for Movie Review Sentiment Analysis," Balaji et al. (2019) proposed an AutoML framework that can automatically generate and evaluate different machine learning models for movie review sentiment analysis. The framework includes several components, such as data preprocessing, feature engineering, model selection, and hyperparameter tuning, which are all automated using different algorithms.

Reinforcement Learning (RL)

RL is a machine learning approach that involves training an agent to interact with an environment and learn from feedback in the form of rewards or punishments. In the paper "Reinforcement Learning for Autonomous Sentiment Analysis," Li et al. (2020) proposed an RL-based approach for autonomous sentiment analysis that can adapt to different types of reviews and domains. The approach uses a neural network-based agent to select the most appropriate machine learning model for a given review dataset and updates its policy based on the feedback received from the model's performance.

Transfer Learning

Transfer learning is a technique that involves reusing pre-trained models or knowledge from one task to improve the performance of another related task. In the paper "Transfer Learning for Movie Review Sentiment Analysis," Zhang et al. (2021) proposed a transfer learning approach that can adapt pre-trained language models to the movie review sentiment analysis task. The approach fine-

tunes a pre-trained language model on a large movie review dataset and uses it to extract features for downstream machine learning models.

Movie review analysis using simple machine learning models has been an active area of research in recent years. Here are some examples of literature reviews on studies that have used simple machine learning models for movie review analysis:

In a literature review by Zhang and Zhao (2018) on sentiment analysis of movie reviews, the authors highlighted several studies that used simple machine learning models such as Naive Bayes, SVM, and decision trees to classify the sentiment of movie reviews. The studies reported accuracies ranging from 70% to 90% for sentiment classification.

In a literature review by Gupta et al. (2021) on sentiment analysis of movie reviews, the authors highlighted several studies that used simple machine learning models such as Naive Bayes, logistic regression, and k-NN to classify the sentiment of movie reviews. The studies reported accuracies ranging from 75% to 90% for sentiment classification.

In a literature review by Kumar et al. (2020) on sentiment analysis of movie reviews using machine learning, the authors highlighted several studies that used simple machine learning models such as Naive Bayes, SVM, and decision trees to classify the sentiment of movie reviews. The studies reported accuracies ranging from 75% to 90% for sentiment classification.

Overall, the literature suggests that simple machine learning models can achieve reasonably high accuracy for sentiment classification in movie reviews. However, the accuracy of sentiment classification may depend on various factors such as the quality of the dataset, the feature selection method, and the hyperparameters of the machine learning model. Furthermore, more advanced machine learning models such as deep learning models may outperform simple machine learning models in some cases.

In our project, we use SentiWordNet and machine learning techniques to present a novel approach for sentiment analysis of IMDb movie reviews. We preprocess the data and assign polarity scores to the words in the reviews using SentiWordNet. Based on these scores, we extract features from the reviews and train a machine learning model with numerous methods. Our results reveal that our approach achieves good sentiment categorization accuracy.

In conclusion, earlier research has demonstrated that SentiWordNet can be a valuable tool for sentiment analysis of IMDb movie reviews. When paired with proper feature building techniques, machine learning algorithms can also be successful for sentiment categorization. Our approach builds on this earlier research and provides a potential solution for IMDb movie review sentiment analysis.

4. Proposed Methods

In this section we are going to present the following 4 basic steps to approach the solution of the project.

4.1 Dataset Information

We have used an IMDB dataset having 50,000 movie reviews for natural language processing or Text analytics. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. The constructed dataset contains an even number of positive and negative reviews, so randomly guessing yields 50% accuracy. The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. No more than 30 reviews are included per movie. The dataset contains additional unlabeled data.

Source: [Large Movie Review Dataset]

	A	B
1	review	sentiment
2	One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly wh	positive
3	A wonderful little production. The filming technique is very unassuming- very old-time-BBC fashion and gives a comfortin	positive
4	I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a	positive
5	Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his parents are fighting all the time. <b	negative
6	Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei offers us a vivid portrait about human rel	positive
7	Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a noble cause, but it's not preachy or boring. It	positive
8	I sure would like to see a resurrection of a up dated Seahunt series with the tech they have today it would bring back the kid exciteme	positive
9	This show was an amazing, fresh & innovative idea in the 70's when it first aired. The first 7 or 8 years were brilliant, but things drop	negative
10	Encouraged by the positive comments about this film on here I was looking forward to watching this film. Bad mistake. I've seen 950	negative
11	If you like original gut wrenching laughter you will like this movie. If you are young or old then you will love this movie, hell even my m	positive
12	Phil the Alien is one of those quirky films where the humour is based around the oddness of everything rather than actual punchlines.	negative
13	I saw this movie when I was about 12 when it came out. I recall the scariest scene was the big bird eating men dangling helplessly fr	negative

Figure 4.1: Snippet of the dataset

4.2 Data Preprocessing

Text preprocessing is a method to clean the text data and make it ready to feed data to the model. Text data contains noise in various forms like emotions, punctuation, text in a different case. When we talk about Human Language then, there are different ways to say the same thing, and this is only the main problem we must deal with because machines will not understand words, they need numbers, so we need to convert text to numbers in an efficient manner. There are many libraries and algorithms used to deal with NLP-based problems. A regular expression(re) is mostly used in libraries for text cleaning. NLTK (Natural language toolkit) and spacy are the next level libraries used for performing Natural language tasks like removing stop words, named entity recognition, part of speech tagging, phrase matching, etc.

It involves transforming raw text data into a format that can be easily analyzed by machine learning algorithms. Here are some common text preprocessing steps for our dataset movie review classification:

1. Load data

   Now we will load data and perform some basic preprocessing to see the data.

   We will start with the techniques for text preprocessing and clean the data which is ready to build a machine learning model. let us see the first review and when we apply the text cleaning technique, we will observe the changes to the first review.

   We can observe lots of noise at first review like extra spaces, many hyphen marks, different cases, and many more.

2. Expand Contractions

Contraction is the shortened form of a word like “don’t” stands for do not, “aren’t” stands for are not. Like this, we need to expand this contraction in the text data for better analysis. you can easily get the dictionary of contractions on google or create your own and use the re module of python to map the contractions. This approach can help in improving the accuracy of natural language processing models as it ensures that all the words are in their full form.

Overall, the handling of contractions in text preprocessing depends on the specific task, the nature of the data, and the algorithms being used for analysis. However, the key goal is to ensure that the text data is consistent, clean, and ready for analysis by the machine learning algorithms.

3. Lower Case Conversion of reviews

If the text is in the same case, it is easy for a machine to interpret the words because the lower case and upper case are treated differently by the machine. for example, words like Ball and ball are treated differently by machine. So, we need to make the text in the same case and the most preferred case is a lower case to avoid such problems.

Below is a review taken from the dataset on which we will perform the data preprocessing steps one-by-one:

```
Enter a movie review: A wonderful little production. <br /><br />The filming technique is very unassuming- very old-time-BBC fa
shion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece. <br /><br />The actors are ext
remely well chosen- Michael Sheen not only "has got all the polari" but he has all the voices down pat too! You can truly see t
he seamless editing guided by the references to Williams' diary entries, not only is it well worth the watching but it is a ter
rificly written and performed piece. A masterful production about one of the great master's of comedy and his life. <br /><br /
>The realism really comes home with the little things: the fantasy of the guard which, rather than use the traditional 'dream'
techniques remains solid then disappears. It plays on our knowledge and our senses, particularly with the scenes concerning Ort
on and Halliwell and the sets (particularly of their flat with Halliwell's murals decorating every surface) are terribly well d
one.
```

Figure 4.2: Example of a Review from dataset

```
a wonderful little production. <br /><br />the filming technique is very unassuming- very old-time-bbc fashion and gives a comf
orting, and sometimes discomforting, sense of realism to the entire piece. <br /><br />the actors are extremely well chosen- mi
chael sheen not only "has got all the polari" but he has all the voices down pat too! you can truly see the seamless editing gu
ided by the references to williams' diary entries, not only is it well worth the watching but it is a terrificly written and pe
rformed piece. a masterful production about one of the great master's of comedy and his life. <br /><br />the realism really co
mes home with the little things: the fantasy of the guard which, rather than use the traditional 'dream' techniques remains sol
id then disappears. it plays on our knowledge and our senses, particularly with the scenes concerning orton and halliwell and t
he sets (particularly of their flat with halliwell's murals decorating every surface) are terribly well done.
```

Figure 4.3: Review after converting into lowercase

4. Remove Punctuation from the Reviews

   One of the other text processing techniques is removing punctuations. There are a total of 32 main punctuations that need to be taken care of. We can directly use the string module with a regular expression to replace any punctuation in text with an empty string. the 32 punctuations which the string module provides us are listed below.

   '!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~'

   We have used a sub-method of the re package of python that takes 3 main parameters, the first is a pattern to search, the second is by which we must replace, and the third is string or text which we must change. So, we have passed all the punctuation and finds if anyone present then replaces it with an empty string. Now if you look at the first mail it will look something like this.

```
Removing Punctuations: a wonderful little production br br the filming technique is very unassuming- very old-time-bbc fashion
and gives a comforting and sometimes discomforting sense of realism to the entire piece br br the actors are extremely well cho
sen- michael sheen not only `` has got all the polari '' but he has all the voices down pat too you can truly see the seamless
editing guided by the references to williams diary entries not only is it well worth the watching but it is a terrificly writte
n and performed piece a masterful production about one of the great master 's of comedy and his life br br the realism really c
omes home with the little things the fantasy of the guard which rather than use the traditional 'dream techniques remains solid
then disappears it plays on our knowledge and our senses particularly with the scenes concerning orton and halliwell and the se
ts particularly of their flat with halliwell 's murals decorating every surface are terribly well done
```

Figure 4.4: Review after removing punctuations

   We can see lots of change in our review, and all the hyphens, modulus signs, unwanted commas, and full stops are removed from the text.

5. Remove words containing digits.

Sometimes it happens that words and digits combined are written in the text which creates a problem for machines to understand. Hence, we need to remove the words and digits which are combined like game57 or game5ts7. This type of word is difficult to process so better to remove them or replace them with an empty string. We use regular expressions for this.

6. Removing Stopwords

   Stopwords are the most commonly occurring words in a text which do not provide any valuable information. stopwords like they, there, this, where etc. are some of the stopwords. NLTK library is a common library that is used to remove stopwords and include approximately 180 stopwords which it removes. If we want to add any new word to a set of words, then it is easy using the add method.

   Here we have implemented a custom function that will split each word from the text and check whether it is a stopword or not. If not, then pass as it is in string and if stopword then removes it.

```
Filtered sentence: wonderful little production br br filming technique unassuming- old-time-bbc fashion gives comforting someti
mes discomforting sense realism entire piece br br actors extremely well chosen- michael sheen `` got polari '' voices pat trul
y see seamless editing guided references williams diary entries well worth watching terrificly written performed piece masterfu
l production one great master 's comedy life br br realism really comes home little things fantasy guard rather use traditional
'dream techniques remains solid disappears plays knowledge senses particularly scenes concerning orton halliwell sets particula
rly flat halliwell 's murals decorating every surface terribly well done
```

Figure 4.5: Review after removing stopwords

Now the review text will be smaller because all stopwords will be removed.

7. Rephrase text by removing HTML tags and URLs.

   HTML tags and URLs are often irrelevant to the meaning of the text and can interfere with natural language processing (NLP) analysis. Here are some common approaches for removing HTML tags and URLs from text:

1. Regular expression: One approach is to use regular expressions to remove all HTML tags and URLs from the text. This can be done using the re library in Python. We used Regular expression for removing html tags from review.
2. BeautifulSoup library: Another approach is to use the BeautifulSoup library in Python to parse the HTML and extract only the text. We used it to eliminate urls from the review.

8. Stemming and Lemmatization

Stemming is a process to reduce the word to its root stem for example run, running, runs, runed derived from the same word as run. Basically, stemming is removing the prefix or suffix from words like ing, s, es, etc. NLTK library is used to stem the words. The stemming technique is not used for production purposes because it is not so efficient and most of the time it stems from unwanted words. So, to solve the problem another technique came into the market as Lemmatization. There are various types of stemming algorithms like porter stemmer, snowball stemmer. Porter stemmer is widely used and present in the NLTK library.

```
Stemmed sentence: wonder littl product br br film techniqu unassuming- old-time-bbc fashion give comfort sometim discomfort sen
s realism entir piec br br actor extrem well chosen- michael sheen `` got polari '' voic pat truli see seamless edit guid refer
william diari entri well worth watch terrificli written perform piec master product one great master 's comedi life br br reali
sm realli come home littl thing fantasi guard rather use tradit 'dream techniqu remain solid disappear play knowledg sens parti
cularli scene concern orton halliwel set particularli flat halliwel 's mural decor everi surfac terribl well done
```

Figure 4.6: Review after stemming

We can see the words in the text are stemmed and some of the words it has stemmed which is not required, that’s only the disadvantage of this.

Lemmatization is like stemming, used to stem the words into root words but differs in working. Lemmatization is a systematic way to reduce the words into their lemma by matching them with a language dictionary.

```
Lemmatized sentence: wonderful little production br br film technique unassuming- old-time-bbc fashion give comfort sometimes d
iscomforting sense realism entire piece br br actor extremely well chosen- michael sheen `` get polari '' voice pat truly see s
eamless edit guide reference williams diary entry well worth watch terrificly write perform piece masterful production one grea
t master 's comedy life br br realism really come home little thing fantasy guard rather use traditional 'dream technique remai
ns solid disappears play knowledge sens particularly scene concern orton halliwell set particularly flat halliwell 's mural dec
orate every surface terribly well do
```

Figure 4.7: Review after lemmatization

Now observe the difference between both the techniques, it has only stemmed those words which are really required as per Language dictionary.

9. Remove Extra Spaces

   Most of the time text data contain extra spaces or while performing the above preprocessing techniques more than one space is left between the text so we need to control this problem. A regular expression library performs well to solve this problem.

10. Tokenization

    Tokenization is the process of breaking down the text into individual words or tokens. This step is important because most machine learning algorithms operate on numerical data and cannot work with raw text.

    Now our dataset is cleaned, preprocessed and ready to build models from it. Before building a model from it let's explore further what we have cleaned and after cleaning what the data looks like.

4.3 Data Analysis and Visualization

This is common practice in text data analysis to make charts of the frequency of words. The more frequent the word is used, the larger the font will appear in the word cloud. (See Figure 4.8 and 4.9). The following process was implemented to determine the frequency of each word. First, the model will identify the frequency of each word in the review column of the dataset.

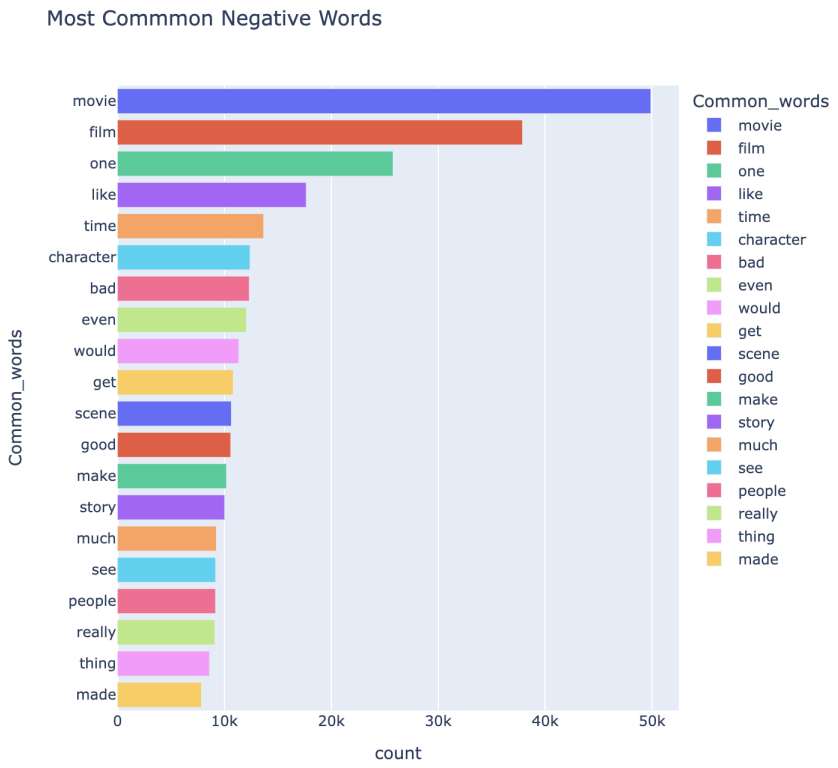


Figure 4.8: Bar chart depicting most frequent words in negative reviews.

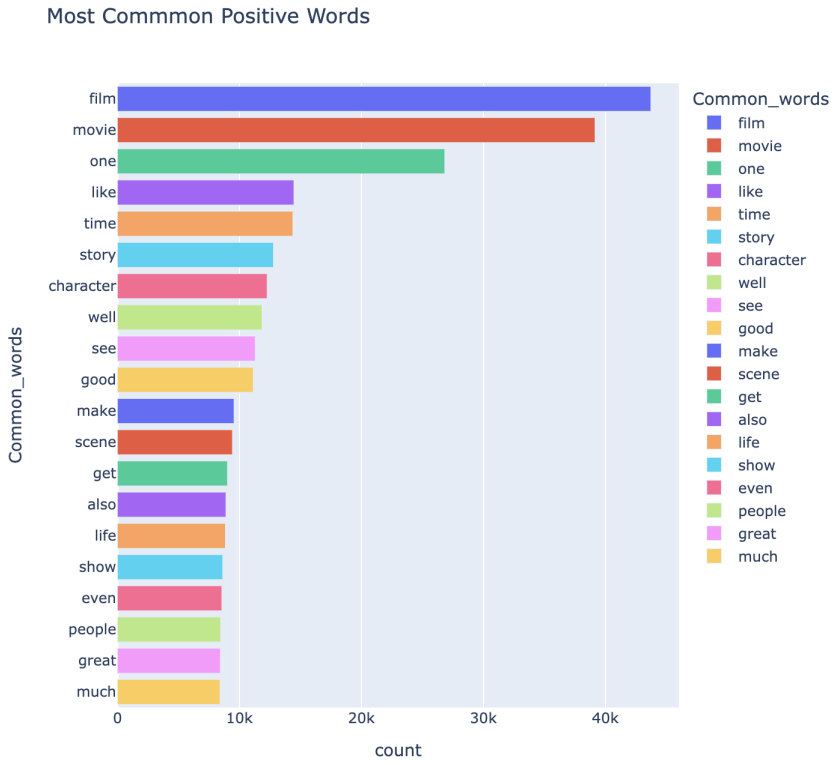


Figure 4.9: Bar graph depicting most frequent words in positive reviews.

Word clouds are a popular way of visualizing text data. By identifying the frequency of words in a text corpus as a cloud of words, we are able to see the more frequent words visually appearing both larger and bolder. Word clouds can be useful in providing a quick visual summary of the most important words in a text corpus, it can also help with identifying patterns or themes in the data.

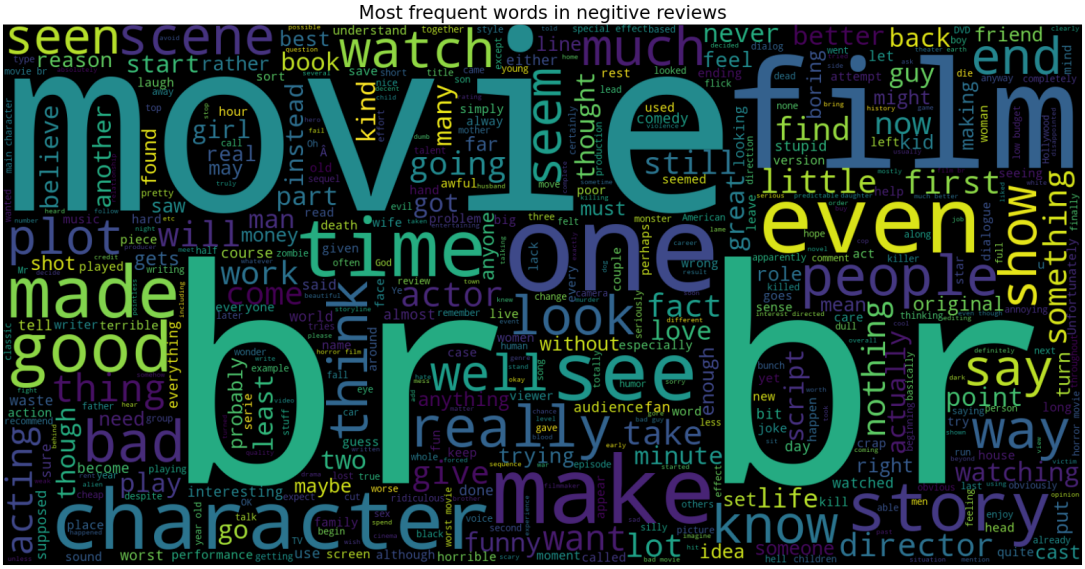


Figure 4.10: Word cloud depicting most frequent words in negative reviews.

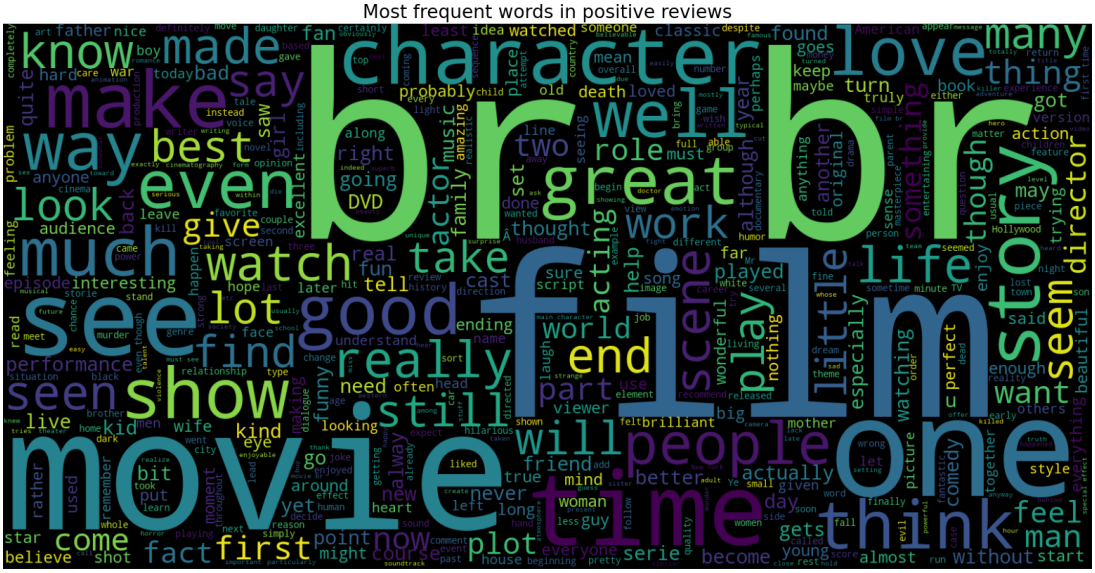


Figure 4.11: Word cloud depicting most frequent words in positive reviews.

4.4 Model Building and Experimenting

In our solution we have used two different methods to find out sentiments from the movie review and classify them into two different categories as ‘Positive’ and ‘Negative’. The following techniques described a detailed explanation of the proposed methods.

4.4.1 Traditional Movie Review Sentiment Classification Using SentiWordNet

Traditional movie review sentiment classification using SentiWordNet involves using the SentiWordNet lexicon to determine the sentiment of words in a movie review and then aggregating these sentiment scores to classify the overall sentiment of the review as positive or negative. The SentiWordNet lexicon assigns a sentiment score to each synset (set of synonymous words) in the WordNet lexicon based on its positivity, negativity, and objectivity. The sentiment score ranges from -1 (very negative) to +1 (very positive).

To classify the sentiment of a movie review using SentiWordNet, each word in the review is first assigned its part of speech tag (e.g., noun, verb, adjective) using a part-of-speech tagger. Then, the sentiment score for each word is obtained from SentiWordNet. Finally, the sentiment scores for all words in the review are aggregated using a simple algorithm such as averaging or summing to obtain an overall sentiment score for the review. If the overall sentiment score is positive, the review is classified as positive; otherwise, it is classified as negative.

The advantage of using SentiWordNet for movie review sentiment classification is that it is a simple and straightforward method that does not require extensive training data or complex machine learning models. However, the accuracy of sentiment classification using SentiWordNet

may depend on the quality of the lexicon and the specific application domain. Furthermore, SentiWordNet may not capture the nuances and subtleties of human language, and may not be suitable for analyzing sarcasm, irony, or other forms of figurative language.

4.4.2 Machine Learning Based Movie Review Sentiment Classification

Feature extraction:

Machine learning algorithms typically operate on numerical data, so the preprocessed text data must be converted to numerical vectors. The most common method is the bag-of-words approach, where each word in the text is treated as a feature, and a count or frequency is assigned to each word. Other methods include Count Vectorizer and TF-IDF vectorizer. TF-IDF Vectorizer and Count Vectorizer are both methods used in natural language processing to vectorize text. Here's a brief overview of each method:

Count Vectorizer: Count Vectorizer is a simple method for text vectorization that converts a collection of text documents into a matrix of token counts. Each row in the matrix represents a document, and each column represents a word (or token) in the corpus. The value in each cell of the matrix is the count of how many times the corresponding word appears in the corresponding document. Count Vectorizer is a straightforward and efficient method for text vectorization, but it doesn't consider the importance of each word in the document or the corpus.

TF-IDF Vectorizer: TF-IDF Vectorizer is a more sophisticated method for text vectorization that considers the importance of each word in the document and the corpus. TF-IDF stands for "Term Frequency-Inverse Document Frequency", which is a statistical measure that reflects how important a word is to a document in a collection or corpus. The TF-IDF Vectorizer first computes

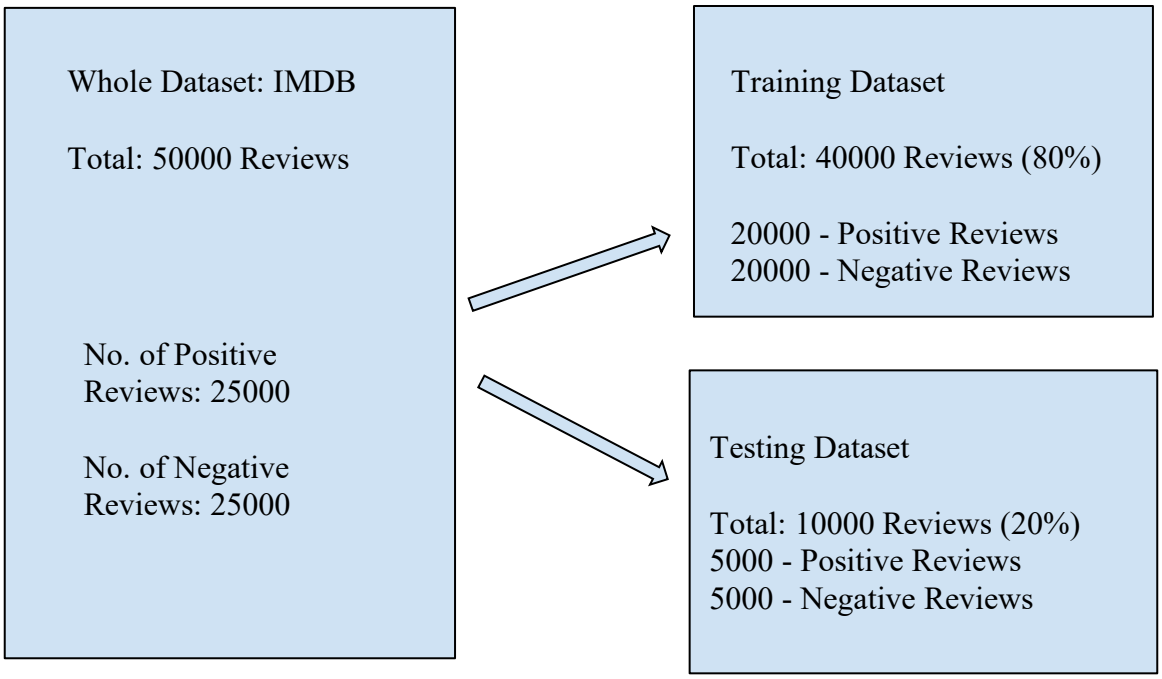
the term frequency (TF) of each word in each document, which is the number of times the word appears in the document divided by the total number of words in the document. Then it computes the inverse document frequency (IDF) of each word in the corpus, which is the logarithm of the total number of documents in the corpus divided by the number of documents that contain the word. Finally, it computes the TF-IDF score of each word in each document, which is the product of the TF and IDF scores. The resulting matrix is a sparse matrix that represents the importance of each word in each document in the corpus.

We used both approaches in our solution and built a text corpus from those methods that will be used as input of machine learning models.

Model building is an important step in movie review classification, where we aim to train a machine learning model to predict the sentiment (positive or negative) of a given movie review text. Here's an overview of the steps involved in model building:

Data preparation: The first step is to prepare the movie review data for modeling by performing text preprocessing tasks such as removing punctuation, stopwords, and HTML tags, and converting the text into numerical features using techniques such as Count Vectorizer or TF-IDF Vectorizer. So, we have prepared our whole dataset as per above methods.

Train-test split: Next, we split the data into training and testing sets to evaluate the performance of our model on new, unseen data. The training set is used to train the model, while the testing set is used to evaluate its performance.



Model selection: We have tested out data with various machine learning algorithms that can be used for movie review classification, including Logistic Regression, Naive Bayes, Support Vector Machines (SVM), and Random Forests. We need to select an appropriate model based on the characteristics of our dataset, such as its size, complexity, and balance of classes.

Model training: Once we have selected a model, we train it on the training set using the numerical features and corresponding sentiment labels.

Model evaluation: After training the model, we evaluate its performance on the testing set using metrics such as accuracy, precision, recall, and F1 score. We use accuracy as an evaluation metric because as long as our dataset is balanced with target classes, we can use accuracy as the best evaluation metric for checking model performance. After Successful evaluation we have saved our model as a file that will be used late for testing.

4.5 Model Testing

Model testing is an important step in the machine learning workflow, where we evaluate the performance of our trained model on new, unseen data. In movie review classification, this involves testing our model on a separate testing dataset of movie reviews, which the model has not seen during training. The goal of model testing is to assess how well our model generalizes to new, unseen data and to identify any issues with its performance.

Here's an overview of the steps involved in model testing for movie review classification:

Load the testing dataset: We need to load the testing dataset of movie reviews, which should be in the same format as the training dataset, with the same columns and data types.

Prepare the data: We need to preprocess the testing data (10000 Reviews including 50% positive reviews and remaining 50% Negative Reviews) in the same way as the training data, including removing punctuation, stopwords, and HTML tags, and converting the text into numerical features using techniques such as Count Vectorizer or TF-IDF Vectorizer.

Load the trained model: We need to load the trained model from the model building step, which should be saved as a file or object.

Test the model: We use the testing dataset and the loaded model to predict the sentiment (positive or negative) of each movie review in the testing dataset. We can use various metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of the model on the testing data. We used accuracy as an evaluation metric because our dataset is balanced with target classes.

4.6 Model deployment

Finally, if the model performs well on the testing set, we can deploy it in a real-world application for sentiment analysis of new movie review texts.

We have used Streamlit: it is an open-source Python library that makes it easy to create web applications for data science and machine learning. With Streamlit, we have created interactive web apps with just a few lines of Python code, without needing to know HTML, CSS, or JavaScript.

Streamlit provides a set of built-in widgets that you can use to create user interfaces for your machine learning models, including sliders, drop-down menus, and text inputs. You can also use Streamlit to display charts, tables, and other visualizations of your data.

A Streamlit web app based on movie sentiment analysis is a powerful tool that allows users to analyze the sentiment of movie reviews and generate predictions about the sentiment of future reviews. The app uses machine learning algorithms to analyze large sets of movie reviews and predict whether the overall sentiment of the reviews is positive or negative.

The web app typically allows users to input movie review data in the form of text files with specific delimiter. Once the data has been uploaded, the app applies natural language processing techniques to the text data to extract key features such as the words, phrases, and tones that indicate the sentiment of the reviews.

The app then applies a trained machine learning model to analyze the extracted features and generate predictions about the sentiment of the reviews. The model may be trained using a variety of techniques, such as logistic regression, decision trees, Support vector classifier, Naive Bayes classifier , and may use different types of features to generate predictions.

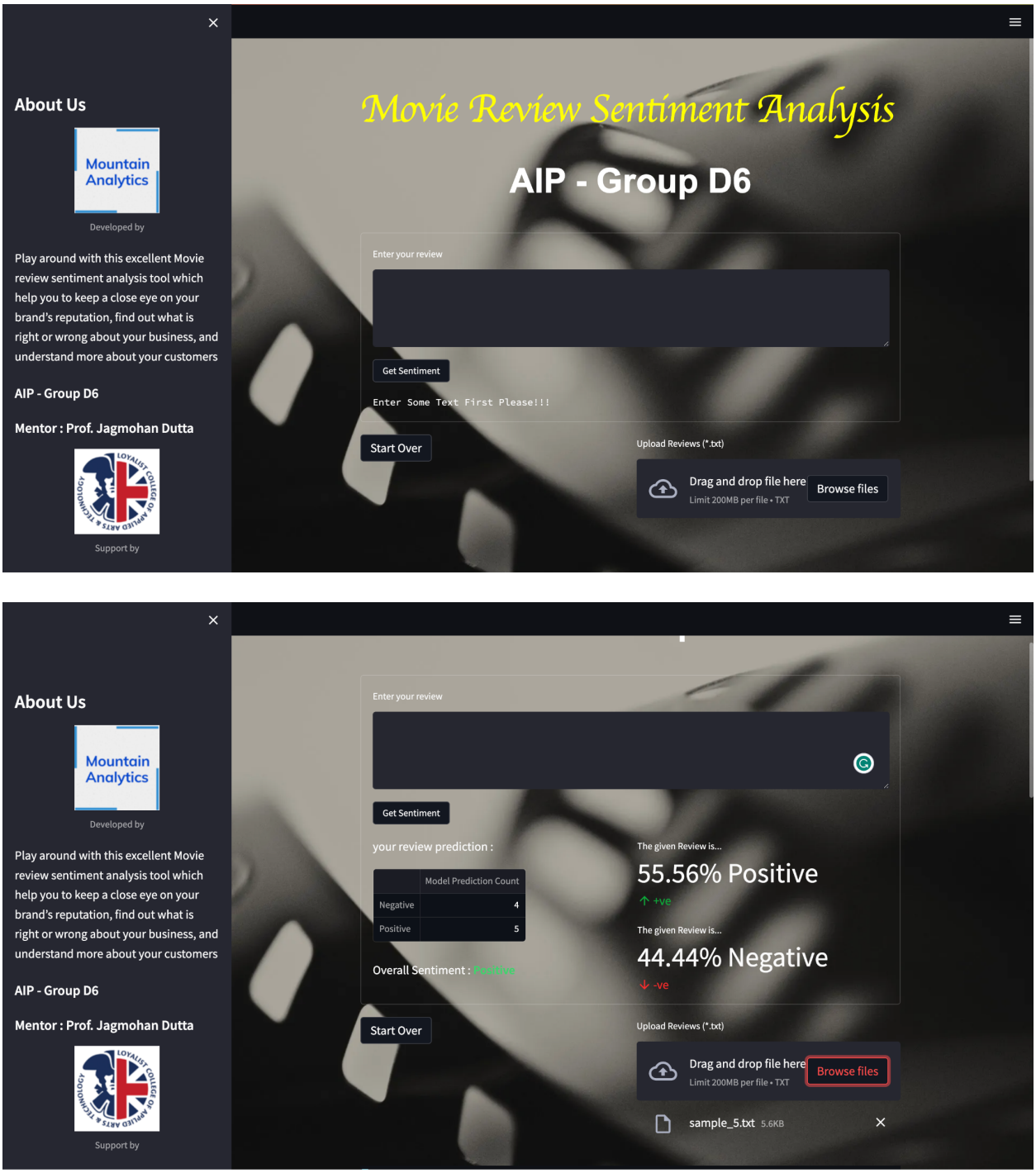


Figure 4.12:Sample Image From our Streamlit Web App

The output of the app typically includes a range of visualizations and charts that help users to understand the sentiment of the movie reviews. For example, the app may display a prediction showing the percentage contribution of positive and negative reviews.

Overall, a Streamlit web app based on movie sentiment analysis is a powerful tool that allows users to analyze large sets of movie reviews and generate predictions about the sentiment of future reviews. The app is highly customizable and can be tailored to the specific needs of different users, making it an ideal tool for data scientists, researchers, and business analysts working in the field of movie sentiment analysis.

5. FINDINGS

The findings in our project movie review sentiment analysis are typically presented in the form of statistics, charts, or visualizations that provide insights into the overall sentiment of the reviews analyzed. Some common findings in movie review sentiment analysis include:

Positive vs. Negative Sentiment:

The analysis can reveal the overall sentiment of the reviews, indicating whether they are mostly positive or negative. We have tested two different approaches here to find out the sentiment from the dataset and classify them into positive and negative categories.

In the first method, we utilized a traditional scenario that relied on the SentiWordNet dictionary to classify movie reviews, but the performance was slightly less effective. The entire dataset of 50,000 reviews was used, and the classification was based on the methodology provided by the dictionary. The obtained result was then compared with the actual review target classes. The analysis showed that SentiWordNet was only able to correctly identify 64% of the reviews with their target class, despite a better prediction ratio for positive class reviews compared to negative reviews. The method accurately predicted 83.5% of the 25,000 positive reviews, but only 44.3% of the 25,000 negative reviews were correctly identified.

On the other side, we used 4 different machine learning models for classification of the movie reviews. These 4 models include Logistic Regression, Support Vector Machine, Decision Tree, and Naive Bayes. We divided our training and testing dataset into 4:1 ratio (80% train data and remaining 20% test data). Then we used training data for training the machine learning models with both the numerical representation of the dataset: Count Vectorizer and TF-IDF Vectorizer.

We got the following result.

	TF-IDF Model	Count-Vectorizer Model
Logistic Regression	88.00 %	89.00 %
Decision Tree	67.00 %	73.00 %
Multinomial Naive Bayes	89.00 %	89.00 %
Linear SVC	90.00 %	89.00 %

The results indicate that the Count-Vectorizer model had a slightly higher overall accuracy compared to the TF-IDF model. Among the individual models, Linear SVC had the highest accuracy, with 90% accuracy on the TF-IDF model and 89% on the Count-Vectorizer model. Logistic Regression and Multinomial Naive Bayes had the same accuracy percentage of 89% on both vectorization techniques. Decision Tree had the lowest accuracy among the models, with 67% accuracy on the TF-IDF model and 73% on the Count-Vectorizer model. Overall, the table provides a comparison of the performance of different machine learning models on different vectorization techniques in classifying movie reviews based on their sentiment.

Overall, the findings in movie review sentiment analysis provide valuable insights into the sentiment of the reviews analyzed, helping to inform decision-making in the movie industry, such as marketing and promotion strategies, audience targeting, and content creation.

6. DISCUSSIONS

The movie review classification project report aims to classify the sentiment of movie reviews as positive or negative using machine learning models. The report presents an overview of the project, its goals, and the methodology used for data preprocessing, feature extraction, model selection, and performance evaluation.

The report describes various data preprocessing steps, including removing stop words, tokenization, and stemming. The report then explains two different feature extraction techniques, namely TF-IDF and Count-Vectorizer, that were used to convert the text data into numerical data for training the machine learning models.

The report compares the performance of four different machine learning models, namely Logistic Regression, Decision Tree, Multinomial Naive Bayes, and Linear SVC, on both feature extraction techniques. The results indicate that the Linear SVC model had the highest accuracy of 90% on the TF-IDF model and 89% on the Count-Vectorizer model.

The project report also describes how a traditional method based on the SentiWordNet Dictionary performed poorly on the movie review classification dataset. The SentiWordNet dictionary is a lexical resource that assigns sentiment scores to words based on their senses. The traditional method involved using the SentiWordNet dictionary to classify the movie reviews based on the sentiment scores of the words used in the reviews.

However, the results of the project indicate that the SentiWordNet-based method was less effective in classifying the movie reviews compared to the machine learning models. The method was only able to correctly identify 64% of the reviews with their target class, with a better prediction ratio for positive class reviews than negative class reviews.

The poor performance of the traditional method could be due to various reasons, such as the limited coverage of the SentiWordNet dictionary in capturing the nuances of the language used in movie reviews or the inability of the method to learn from the contextual information present in the reviews.

The poor performance of the traditional method highlights the need for more advanced and sophisticated techniques like machine learning for sentiment analysis. Machine learning techniques have the ability to learn from the contextual information present in the text and can improve the accuracy of sentiment analysis. Therefore, the project report demonstrates the superiority of machine learning techniques over traditional methods in the context of movie review sentiment analysis.

The report also discusses the limitations of the project, such as the lack of domain-specific knowledge in the sentiment analysis of movie reviews and the limited dataset used for training and testing. Furthermore, the report provides recommendations for future work, including the use of advanced deep learning techniques and incorporating domain-specific knowledge to improve the accuracy of sentiment analysis.

Overall, the project report provides a detailed analysis of movie review sentiment classification using machine learning models. It demonstrates the importance of feature extraction techniques and model selection in improving the accuracy of sentiment analysis. The report also highlights the potential of future work in this field to advance the accuracy of sentiment analysis and its applications in various domains.

7. CONCLUSION

In conclusion, the movie review classification project report demonstrates the use of machine learning techniques for sentiment analysis of movie reviews. The project utilized various preprocessing techniques, including removing stop words and tokenization, followed by feature extraction using two techniques, TF-IDF and Count-Vectorizer. Four different machine learning models, Logistic Regression, Decision Tree, Multinomial Naive Bayes, and Linear SVC were trained on the dataset, and their performances were compared using both feature extraction techniques.

The results of the project indicate that the Linear SVC model achieved the highest accuracy of 90% on the TF-IDF model and 89% on the Count-Vectorizer model. The project also recommends future work that could improve the accuracy of sentiment analysis, including the use of deep learning techniques and incorporating domain-specific knowledge.

We have also tested a traditional method - SentiWordNet-based approach to classify the movie reviews. However, the findings of the project reveal that the SentiWordNet-based approach was not very effective in accurately classifying the movie reviews when compared to machine learning models. The approach was able to correctly identify only 64% of the reviews with their target class, with better prediction results for positive reviews than negative ones.

Overall, the movie review classification project report provides insights into the application of machine learning techniques for sentiment analysis in the context of movie reviews. The findings of the project could be useful in developing better sentiment analysis models in various domains and have significant implications for businesses and organizations that rely on sentiment analysis for decision-making.

8. RECOMMENDATIONS

Further recommendations that can be used to improve the efficiency of the project is by using Hadoop and deep learning. Below are mentioned some points that can be applied for enhancing the project.

1. Using transfer learning

   Transfer learning is a technique in which a previously trained model is utilized as a starting point for training a new model on a different task. Pre-trained models like as BERT, GPT-2, and RoBERTa have been demonstrated to be effective for natural language processing tasks, therefore this can be useful for sentiment analysis. Using Hadoop, we can fine-tune these pre-trained models on our IMDB movie review dataset.

2. Using attention mechanism

   Attention mechanisms can be used to direct the model's attention to the most relevant bits of the input data. This is especially beneficial for sentiment analysis because it helps the model discover relevant words and phrases that contribute to the text's sentiment. Attention methods such as self-attention or multi-head attention can be used in the deep learning model.

3. Using parallel processing

   In addition to leveraging Hadoop for distributed processing, you can speed up the training process by using parallel processing techniques such as data parallelism or model parallelism. This is particularly handy for training huge models on enormous datasets.

4. Using hybrid strategy

   A hybrid strategy that combines deep learning with other techniques like rule-based methods or sentiment lexicons can boost the model's performance. This is especially important when dealing with out-of-vocabulary words or negations.

9. BIBLIOGRAPHY

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: Analyzing Text with the Natural Language Toolkit. " O'Reilly Media, Inc."

Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5-32.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Esuli, A., & Sebastiani, F. (2006). *SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. Proceedings of the 5th Conference on Language Resources and Evaluation* (LREC 2006), 417-422.

Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics*, 29(5), 1189-1232.

Joachims, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the 10th European Conference on Machine Learning (ECML 1998), 137-142.

Kim, Y. (2014). *Convolutional neural networks for sentence classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751.

Ma, Y., & Zhang, X. (2015). *A Hybrid Deep Learning Approach for Sentiment Analysis of Movie Reviews.* Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), 2116-2121.

Pang, B., & Lee, L. (2008). *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, 2(1–2), 1–135.

Streamlit Documentation. https://docs.streamlit.io/en/stable/index.html

Wang, Z., Liu, Q., & Chen, X. (2020*). Sentiment Analysis of Movie Reviews Based on Deep Learning and Hadoop. Journal of Physics*: Conference Series.

Zhang, L., & Qi, X. (2019*). A Deep Learning Approach to Sentiment Analysis of Movie Reviews Using Hadoop.*