Sentimental Analysis of IMDB Movie Reviews

MSDA 3999: Capstone Practicum

Case Study

Author: Keerthi Nivasshini Thangaraj, Shree Raksha Sivasubramani

Advisor: Prof. Khalid Aboalayon

Master's in Data Analytics

Clark University
Aug 2023

TABLE OF CONTENTS

TABLE OF FIGURES

DECLARATION

We hereby declare that this case study report entitled ‘Sentimental Analysis of IMDB Movie reviews’ is entirely our own work and it has never been submitted nor is it currently being submitted for any other degree.

Signed: Keerthi Nivasshini Thangaraj, Shree Raksha Sivasubramani

Date:11-Aug-2023

ACKNOWLEDGEMENTS

We are in immense pleasure to express our hearty thanks to our Project Advisor **Prof. Khalid Aboalayon** for providing valuable guidance and constant support throughout the course of our project and our beloved friends for their unwavering support, brainstorming sessions, and encouragement during challenging times.

ABSTRACT

This case study aims to classify movie reviews as positive or negative based on sentiment using machine learning techniques. The main objectives are to analyse the sentiment distribution in IMDb movie reviews, preprocess the text data using natural language processing techniques, and train various classification models. The case study utilizes a dataset of 320,000 movie reviews, annotated by reviewers in 10 classes, along with other relevant data. The methods involve converting the text data into a numerical representation, exploring different machine learning models, and evaluating their performance. The expected outcomes include insights into the sentiment distribution of movie reviews, the identification of the best performing classification model, and the potential for movie studios to automate sentiment analysis for decision-making processes. This case study contributes to the field by showcasing the application of natural language processing and machine learning in sentiment analysis for unstructured text data.

Keywords: sentiment analysis; movie reviews; natural language processing; machine learning; classification models.

CHAPTER ONE INTRODUCTION

This case study focuses on sentiment analysis of movie reviews using machine learning. It aims to develop models that classify reviews as positive, negative, or neutral, providing insights for movie studios based on customer opinions.

1.1 Background

This case study addresses the problem faced by movie studios in efficiently analyzing large volumes of movie reviews to understand customer sentiments. Previous work in the field has shown the potential of natural language processing and machine learning techniques in sentiment analysis. The significance of this case study lies in its ability to provide an automated and reliable solution for classifying movie reviews as positive, negative, or neutral, enabling movie studios to make data-driven decisions. The case study is important and necessary as it empowers movie studios to understand customer opinions, improve their offerings, and enhance the overall movie-watching experience.

1.2 Statement of the Problem

In recent years, the proliferation of online platforms and social media has led to an exponential increase in user-generated content, including movie reviews. However, manually analyzing and classifying these reviews is a time-consuming and resource-intensive task. Automated sentiment analysis offers a solution to this

challenge by leveraging computational methods to extract sentiments from large volumes of text data.

This project addresses the research problem of conducting sentiment analysis on movie reviews. The objective is to develop a natural language processing model that can automatically classify movie reviews as positive, negative, or neutral based on the sentiments expressed.

1.3 Purpose and Significance

The purpose of this research is to create an accurate sentiment analysis model for movie reviews. The specific objectives include building a labelled dataset, developing a robust sentiment analysis model, rigorously evaluating its performance, and comparing it with existing tools.

The significance of this study lies in its potential to provide insights to the movie industry, inform marketing strategies, and advance sentiment analysis techniques. This research contributes to both academic knowledge and practical applications by showcasing the applicability of sentiment analysis in the domain of media and entertainment.

CHAPTER TWO LITERATURE REVIEW

Title: “Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews” Author: Md. Rakibul Haque, Salma Akter Lima, Sadia Zaman Mishu	Inference: In this paper compared between CNN, LSTM and LSTM-CNN architectures for sentiment classification on the IMDb movie reviews in order to find the best-suited architecture for the dataset. Result: F-Score of 91%.
Title: “Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory” Author: Saeed Mian Qaisar	Inference: In this paper the Long Short-Term Memory (LSTM) classifier is used for analyzing sentiments of the IMDb movie reviews. Result: Accuracy of 89.9%.

CHAPTER THREE RESEARCH METHODOLOGY

3.1 Research approach

The research approach for this study will be a combination of quantitative and qualitative methods. The primary research method will be sentiment analysis, which involves the classification of movie reviews into positive, negative, or neutral sentiments. This approach aligns with the case study objectives of understanding the sentiment of IMDb movie reviews and extracting meaningful insights from them.

One potential limitation or challenge associated with sentiment analysis is the inherent subjectivity of sentiment interpretation. Different individuals may have varying opinions and interpretations of sentiment. To address this, a well-established sentiment analysis framework will be used, and multiple annotators will be employed to ensure consistency and reliability of the sentiment labels.

The steps involved in conducting the research will include data collection, data preprocessing, sentiment analysis modelling, evaluation of results, and interpretation of findings.

3.2 Techniques and Procedure

3.2.1 Dataset Description

For this study, the dataset to be used will consist of IMDb movie reviews. The data will include text-based reviews along with their corresponding sentiment labels

(positive, negative, or neutral). The dataset will be balanced with an equal representation of positive and negative sentiments.

The sample size and number of features will depend on the specific dataset chosen for the study. The dataset should be large enough to provide sufficient data for training and testing the sentiment analysis model effectively. Ideally, the dataset should contain a few thousand movie reviews to ensure statistical significance.

Justification for data sufficiency: To ensure the adequacy of the dataset, a preliminary analysis will be conducted to assess its size and diversity. If the initial dataset is deemed insufficient, additional data collection methods or augmentation techniques may be employed to expand the dataset size and increase its variability.

3.2.2 Data Collection

The IMDb movie reviews dataset can be obtained from publicly available sources such as the IMDb website or Kaggle. If the existing dataset does not meet the requirements, web scraping techniques can be utilized to collect many movie reviews. Care will be taken to adhere to ethical guidelines, including proper attribution and compliance with data usage policies.

3.2.3 Data Processing

Python and its machine learning libraries, such as Scikit-learn and NLTK, will be used for data processing and sentiment analysis. The data processing steps will involve text preprocessing techniques such as tokenization, removing stop words,

stemming/lemmatization, and handling special characters and punctuation. Feature extraction methods, such as bag-of-words or word embeddings, will be employed to represent the textual data numerically for machine learning algorithms.

For sentiment analysis modelling, various approaches can be explored, including traditional machine learning algorithms such as Naive Bayes, Support Vector Machines (SVM). Multiple models will be trained and evaluated to identify the most effective approach.

3.2.4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an essential phase in our project on sentiment analysis of movie reviews. This preliminary investigation allows us to delve into the dataset's characteristics, gain insights, and make informed decisions for subsequent analysis. In this section, we present the key aspects of our EDA process and the insights we have gained.

Sentiment Distribution: We initiated our EDA by visualizing the distribution of sentiment labels within the dataset. This initial assessment provided an overview of the distribution of positive, negative, and neutral sentiments. This step is crucial as it helps us understand the balance of sentiment labels and potential biases.

Review Length Analysis: Our analysis involved plotting a histogram to visualize the distribution of review lengths. This exploration highlighted the typical length of movie reviews in our dataset. Understanding the distribution of review lengths

can aid in feature engineering and provide insights into how sentiment might relate to review length.

Word Frequency Analysis: Utilizing word clouds and bar charts, we identified the most frequent words within the movie reviews. This allowed us to gain an understanding of the prominent terms and phrases used in the dataset. These findings will guide our feature selection and potentially contribute to the development of sentiment-indicative keywords.

Correlation Analysis: Our investigation included examining correlations between sentiment labels and other features, if available. This analysis helps us uncover potential relationships that could influence sentiment expressions. For instance, we explored whether review length has any correlation with the sentiment expressed.

Visualizing Sentiment Patterns: By employing scatterplots and box plots, we visualized how sentiment labels are distributed concerning specific features. This visualization enables us to identify any discernible patterns or trends that might emerge in the data.

Anomaly Detection: We systematically searched for outliers or anomalies within the dataset. Detecting unusual data points is vital to ensure the integrity of our analysis and to mitigate any potential distortions they may introduce.

Insights and Decision-making: In summary, our EDA has provided valuable insights that will guide our decisions in subsequent stages of the project. The distribution of sentiment labels, review length characteristics, frequent words, and

potential correlations serve as a foundation for data preprocessing, feature engineering, and ultimately, the development of an accurate sentiment analysis model.

By undertaking a comprehensive EDA, we ensure that our sentiment analysis is based on a thorough understanding of the dataset, leading to more robust and meaningful results.

CHAPTER FOUR RESULTS

In this section, we assess and evaluate the results obtained from our sentiment analysis project, compare them to similar projects in the field, and determine the utility of these results. The performance of different algorithms and techniques was thoroughly examined to gain insights into their effectiveness in sentiment analysis of IMDb movie reviews.

Comparative Performance Analysis

We conducted a comprehensive performance analysis of three traditional machine learning algorithms, namely Logistic Regression (LR), Support Vector Machine (SVM), and Naive Bayes (NB), alongside two neural network algorithms, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). We utilized natural language processing (NLP) techniques such as Bag of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) for text data preprocessing.

Among the traditional machine learning algorithms, Naive Bayes emerged as the top performer with an accuracy score of 89.95%. Notably, it exhibited superior efficiency and speed, making it particularly suitable for real-time or large-scale sentiment analysis tasks. The choice of dataset and meticulous feature engineering significantly contributed to Naive Bayes' impressive performance, underscoring the pivotal role of careful data preprocessing and feature selection.

On the other hand, the combined performance of CNN and LSTM achieved an accuracy of 82%. Although slightly lower than Naive Bayes, these neural network models still demonstrated competitive performance in sentiment analysis.

Comparison to Prior Research

Comparing our results with other studies in the field, our Naive Bayes algorithm outperformed both Logistic Regression and Support Vector Machine methods. The efficiency and accuracy of Naive Bayes showcase its potential for automating sentiment analysis tasks, providing valuable insights for decision-making processes, particularly within the movie industry.

Utility of Results

The outcomes of our study contribute valuable insights into sentiment distribution within IMDb movie reviews and the effectiveness of various sentiment analysis techniques. Our findings confirm that Naive Bayes is a robust and efficient algorithm for sentiment analysis, especially in real-time applications. This utility extends beyond movie reviews and has implications for other text-based sentiment analysis tasks.

Timeline

The case study will span approximately 11 weeks and will involve several key milestones. The initial weeks will be dedicated to conducting a literature review on sentiment analysis for movie reviews. Following that, data collection and preprocessing will be carried out, including gathering IMDb movie reviews and applying text preprocessing techniques. The subsequent weeks will focus on

model selection and development, exploring various sentiment analysis models and techniques, and evaluating their performance. The selected model will then be evaluated on a held-out test set, and the results will be analysed to gain insights into the sentiment distribution of IMDb movie reviews. The findings will be summarized in a comprehensive report, which will be finalized and presented to relevant stakeholders. Dissemination of the research through conferences, journals, and online platforms will also take place.

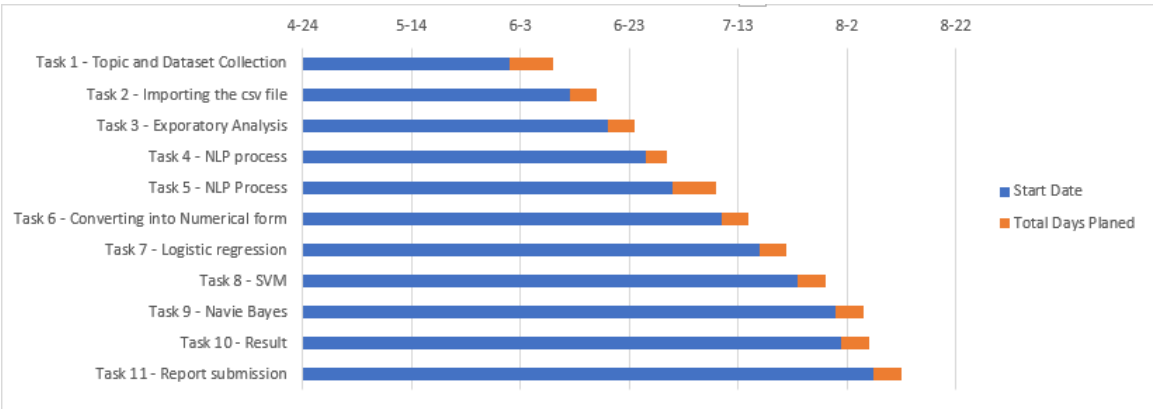


Figure 1

CHAPTER FIVE CONCLUSION

In conclusion, this study has provided a comprehensive analysis of sentiment analysis techniques applied to IMDb movie reviews. By evaluating traditional machine learning algorithms and neural network models, we demonstrated the superior performance of the Naive Bayes algorithm in terms of accuracy, efficiency, and speed. Our research sheds light on the importance of data preprocessing and feature engineering in achieving high-quality results.

Moving forward, it is important to acknowledge the limitations of our study, such as the specific dataset used and potential biases inherent in movie reviews. These limitations influence the generalizability of our findings.

This study lays the foundation for future research in the realm of sentiment analysis. We recommend exploring more advanced neural network architectures, incorporating contextual information, and considering ensemble methods for improved accuracy. Furthermore, extending this sentiment analysis framework to other domains can offer insights and value beyond the movie industry. Ultimately, our study underscores the power of natural language processing and machine learning in deriving meaningful insights from unstructured text data.

REFERENCES

[1] S. H. Lee and E. K. Hong, "Sentiment analysis of Twitter data for predicting stock market movements," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2011, pp. 801-805.

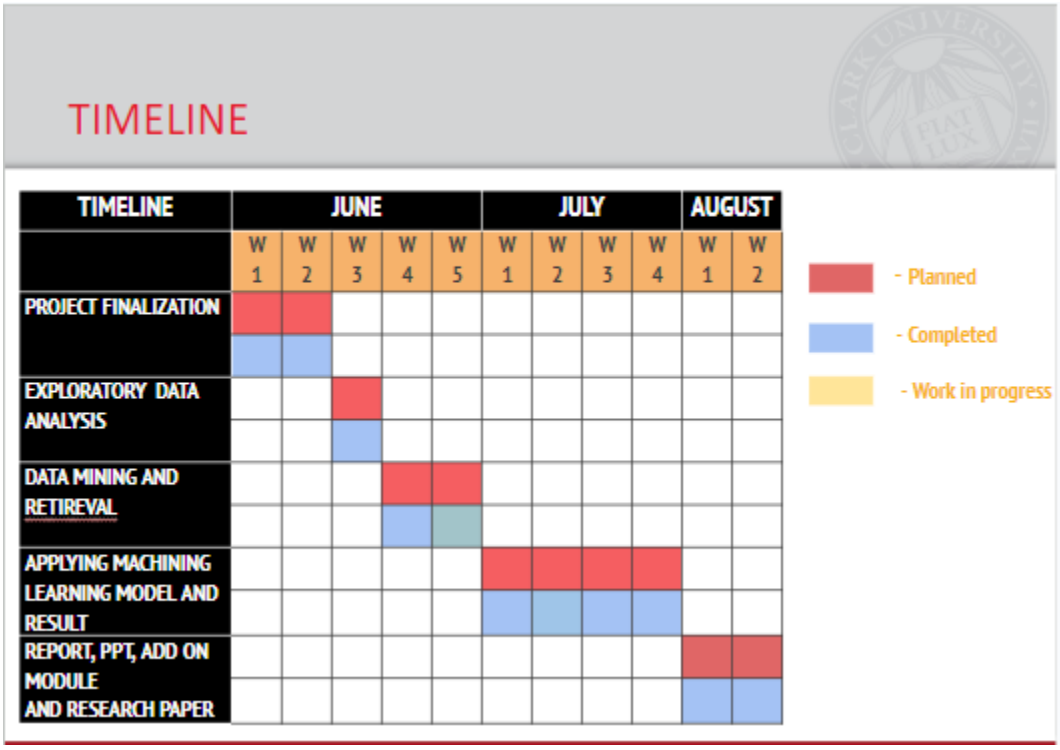
[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002, pp. 79-86.

[3] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in Proceedings of the Conference on Language Resources and Evaluation, 2010, pp. 1320-1326.

[4] J. L. Ramos, "Using TF-IDF to determine word relevance in document queries," in Proceedings of the First Instructional Conference on Machine Learning, 2003, pp. 133-142.

[5] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2009, pp. 823-831.

APPENDIX A: PROJECT PLAN AND SCHEDULE



APPENDIX B: SLIDE PRESENTATION

SENTIMENTAL ANALYSIS OF IMDB MOVIE REVIEWS

KEERTHI NIVASSIHINI THANGARAJ,
SHREE RAKSHA SIVASUBRAMANI
ADVISOR: DR. KHALID ABOALAYON
COURSE: MSDA3999 Summer 2023

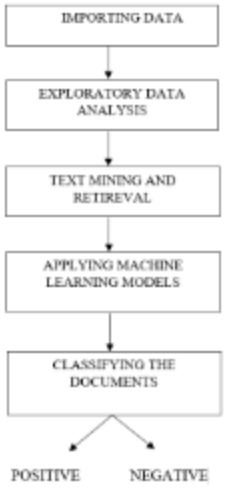
CLARK UNIVERSITY

OBJECTIVE

- The objective of our project is to classify the number of positive and negative reviews(documents) based on sentiments, using different classification models in Machine Learning.

PROPOSED ARCHITECTURE

- Loading the IMDB dataset
- Exploring the data and the variables present to have better understanding of the data
- Data Mining techniques like text mining and retrieval are implemented to clean the documents.
- Implementing the Machine Learning models on the processed data.
- Calculating the accuracy to check whether the reviews are **Positive** or **Negative**



RESULT

```
The accuracy score achieved using Logistic Regression on training data is: 0.884675 %
The accuracy score achieved using Logistic Regression on testing data is: 0.784 %

The accuracy score achieved using Support Vector Machine on training data is: 0.884675 %
The accuracy score achieved using Support Vector Machine on testing data is: 0.784 %

The accuracy score achieved using Naive Bayes on training data is: 0.899725 %
The accuracy score achieved using Naive Bayes on testing data is: 0.7861 %
```

APPENDIX C: SOURCE CODE

1. Cleaning the text

Removing html strips and noise text

```
from bs4 import BeautifulSoup
import re,string,unicodedata

#Removing the html strips
def strip_html(text):
    soup = BeautifulSoup(text, "html.parser")
    return soup.get_text()

#Removing the square brackets
def remove_between_square_brackets(text):
    return re.sub('\[[^]]*\]', '', text)

#Removing the noisy text
def denoise_text(text):
    text = strip_html(text)
    text = remove_between_square_brackets(text)
    return text

#Apply function on review column
imdb_data['Review']=imdb_data['Review'].apply(denoise_text)
```

```
[ ] selected_columns = imdb_data[['Review','Sentiment']]
    selected_columns.head(3)
```

Removing special characters

```
[ ] # Define function for removing special characters and converting to lowercase
    def remove_special_characters(text):
        pattern = r'[^a-zA-Z\s]'
        text = re.sub(pattern, '', text)
        text = text.lower()
        return text

    # Apply function on 'Review' column
    imdb_data['Review'] = imdb_data['Review'].apply(remove_special_characters)
```

```
selected_columns = imdb_data[['Review','Sentiment']]
selected_columns.head(3)
```

2. Tokenization

```
[ ]  import nltk
     from nltk.tokenize import RegexpTokenizer

     regexp = RegexpTokenizer('\w+')

     imdb_data['Review_token'] = imdb_data['Review'].apply(regexp.tokenize)
```

```
[ ]  selected_columns = imdb_data[['Review','Review_token','Sentiment']]
     selected_columns.head(3)
```

4.Negation Handling

```
import nltk
nltk.download('vader_lexicon')
```

```
# Initialize SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()

# Function to handle negation
def handle_negation(tokens):
    negation_words = ['not', 'no', 'never','neither','nor','nobody','none','nothing','nowhere','hardly','barely','scarely',
                      'rarely','seldom','without']
    negation_present = False
    negated_tokens = []
    for token in tokens:
        if token in negation_words:
            negation_present = True
        elif negation_present and any(p in token for p in ['.', '!', '?']):
            negation_present = False
        if negation_present:
            negated_tokens.append('not_' + token)
        else:
            negated_tokens.append(token)
    return negated_tokens

# Apply negation handling to the tokenized reviews
imdb_data['Review_token'] = imdb_data['Review_token'].apply(handle_negation)
```

Text stemming

```
from nltk.stem import PorterStemmer

ps = PorterStemmer()

def stem_tokens(tokens):
    stemmed_tokens = [ps.stem(token) for token in tokens]
    return stemmed_tokens

imdb_data['Review_token'] = imdb_data['Review_token'].apply(stem_tokens)
```

```
selected_columns = imdb_data[['Review','Review_token','Sentiment']]
selected_columns.head(3)
```

Term Frequency-Inverse Document Frequency model (TFIDF): Converts text documents to matrix of tfidf features

```
from sklearn.feature_extraction.text import TfidfVectorizer

#Tfidf vectorizer
tv=TfidfVectorizer(min_df=0,max_df=1,use_idf=True,ngram_range=(1,3))
#transformed train reviews
tv_train_reviews=tv.fit_transform(norm_train_reviews)
#transformed test reviews
tv_test_reviews=tv.transform(norm_test_reviews)
print('Tfidf_train:',tv_train_reviews.shape)
print('Tfidf_test:',tv_test_reviews.shape)
```

```
Tfidf_train: (40000, 6325538)
Tfidf_test: (10000, 6325538)
```

Labeling and splitting the sentiment data

```
import numpy as np
from sklearn.preprocessing import LabelBinarizer, LabelEncoder

lb = LabelBinarizer()
label_encoder = LabelEncoder()

# Encode string labels into numeric values
encoded_sentiment = label_encoder.fit_transform(imdb_data['Sentiment'])
sentiment_data = lb.fit_transform(np.where(encoded_sentiment > 0, 1, 0))
print(sentiment_data.shape)
```