VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI- 590 018, KARNATAKA

An Internship Report
on

Sentiment Analysis of IMDB Dataset

of 50k Movie Reviews

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

Akshay P 1RN19IS018

Under the Guidance of

Mr. Pramoda R
Assistant Professor
Department of ISE

ESTD : 2001
An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

2022 - 2023

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

ESTD : 2001
An Institute with a Difference

CERTIFICATE

Certified that the internship work entitled ***Sentiment Analysis of IMDB Dataset of 50k Movie Reviews*** has been successfully completed by **Akshay P (1RN19IS018)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2022-2023**. The internship report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mr. Pramoda R	**Dr. R Rajkumar / Mr. Pramoda R**	**Dr. Suresh L**
Internship Guide	Internship Coordinators	Professor & HoD
Assistant Professor	Associate Professor	Department of ISE
Department of ISE	Department of ISE	RNSIT

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

Inflow
Information Integrity

PARTICIPATION CERTIFICATE

Start Date: 19th March 2022
End Date: 25th May 2022
Learning Partner: Inflow Technologies Pvt. Ltd.

This Certificate Is Presented To

AKSHAY P

On successfully completing a knowledge transfer session of :

" Data Science & Analytics "

ARIB NAWAL
Trainer

DECLARATION

I, **AKSHAY P [USN: 1RN19IS018]**, student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship project work entitled ***Sentiment Analysis of IMDB Dataset of 50k Movie Reviews*** has been carried out and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2022-2023.

Place: Bengaluru

Date:

AKSHAY P
1RN19IS018

ABSTRACT

Sentiment analysis refers to the task of natural language processing to determine whether a piece of text contains some subjective information and what subjective information it expresses, i.e., whether the attitude behind this text is positive, negative or neutral. Understanding the opinions behind user-generated content automatically is of great help for commercial and political use, among others. The task can be conducted on different levels, classifying the polarity of words, sentences or entire documents.

Sentiment analysis and opinion mining have been acquiring a crucial role in both commercial and research applications because of their possible applicability to several different fields. Therefore, a large number of companies have included the analysis of opinions and sentiments of customers as part of their mission. One of the most interesting applications of these approaches involves the automatic analysis of social network messages or customer reviews, on the basis of the feelings and emotions conveyed. The main result consists of a review of the most interesting methods employed to compare and classify customer reviews of IMDB Movies and a description of advanced tools in this area.

To support sentiment analysis, various toolkits such as the Natural language toolkit (NLTK), open CV, Pattern, and SK Learn packages NLTK support pre-processing of text contents, and also offer the Naïve Bayes supervisor to implement frequency of terms analysis.

ACKNOWLEDGEMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all his wisdom.

We place our heartfelt thanks to **Mr. Pramoda R** Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Arib Nawal, Trainer, Inflow Technologies**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinators **Dr. R Rajkumar,** Associate Professor, and **Mr. Pramoda R**, Assistant Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

Place: Bengaluru

Date:

AKSHAY P

1RN19IS018

TABLE OF CONTENTS

LIST OF FIGURES

Figures:

LIST OF TABLES

Table:

Chapter 1

INTRODUCTION

1.1 ORGANIZATION/INDUSTRY

1.1.1 COMPANY PROFILE

Inflow Technologies is a niche player in the Distribution Services industry providing value added distribution in Cyber Security, Networking, Unified Communications and Collaboration, AIDC & POS, Infrastructure & Application Software, Storage Management and Electronic Security products related Services in South Asia. Inflow Technologies enables system integrators & resellers to design, deploy and adopt IT technologies to facilitate their customer needs.

1.1.2 DOMAIN/TECHNOLOGY

Data analytics is the process of examining data sets in order to find trends and draw conclusions about the information they contain. This technology is widely used in commercial industries to enable organizations to make more-informed business decisions. They help businesses increase revenue, improve operational efficiency, optimize marketing campaigns and bolster customer service efforts. This also enables organizations to respond quickly to emerging market trends and gain a competitive edge over business rivals. Scientists and researchers make use of analytics tools to verify or disprovescientific models, theories and hypotheses. Depending on the particular application, the data that's analyzed can consist of either historical records or new information that has been processed for the real-time analytics.

In our increasingly data-driven world, it's more important than ever to have accessible ways to view and understand data. After all, the demand for data skills in employees is steadily increasing each year. Employees and business owners at every level need to have an understanding of data and of its impact. That's where data visualization comes in handy. With the goal of making data more accessible and understandable, data visualization in the form of dashboards is the go-to tool for manybusinesses to analyze and share information.

1.1.3 DEPARTMENT

R.N.Shetty Institute of Technology (RNSIT) established in the year 2001, is the brain-child of the Group Chairman, Dr. R. N. Shetty. The Murudeshwar Group of Companies headed by Sri.

R. N. Shetty is a leading player in many industries viz construction, manufacturing, hotel, automobile, power & IT services and education. The group has contributed significantly to thefield of education. A number of educational institutions are run by the

R. N. Shetty Trust, RNSIT being one amongst them. With a continuous desire to provide quality education to the society, the group has established RNSIT, an institution to nourish andproduce the best of engineering talents in the country. RNSIT is one of the best and top accredited engineering colleges in Bengaluru.

1.2 PROBLEM STATEMENT

1.2.1 Existing System and their Limitations

The Current system uses linear regression and CNN which are less accurate but analyzing a customer review for sentiment is not easy the readings may be inaccurate as well.

1.2.2 Proposed Solution

To eliminate the drawbacks stated above, four algorithms namely Logistic Regression, Multinomial NB, Linear SVC, Tuned Linear SVC were used to predict the sentiment Tuned Linear SVC is giving higher and accurate results which is better than the previous results without any under fitting or over fitting.

1.2.3 Problem formulation

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses,
though you can use this for many other situations.

Chapter 2

LITERATURE SURVEY

In his work, Pang Lee et al. (2002, 2004), compared the performance of Naïve Bayes, Maximum Entropy and Support Vector Machines in SA on different features like considering only unigrams, bigrams, combination of both, incorporating parts of speech and position information, taking only adjectives *etc*. The result has been summarized in the Table 1.3.
It is observed from the results that:

a. Feature presence is more important than feature frequency.
b. Using Bigrams, the accuracy actually falls.
c. Accuracy improves if all the frequently occurring words from all parts of speech are taken, not only Adjectives.
d. Incorporating position information increases accuracy.
e. When the feature space is small, Naïve Bayes performs better than SVM. But SVM's perform better when feature space is increased.

Bikel et al. (2007) implemented a Subsequence Kernel based Voted Perceptron and compares its performance with standard Support Vector Machines. It is observed that as the number of true positives increase, the increase in the number of false positives is much less in Subsequence Kernel based voted Perceptrons compared to the bag-of-words based SVM's where the increase in false positives with true positives is almost linear. Their model, despite being trained only on the extreme one- and five-star reviews, formed an excellent continuum over reviews with intermediate star ratings, as shown in the figure below. The authorscomment that "It is rare that we see such behavior associated with lexical features which are typically regarded as discrete and combinatorial. Finally, we note that the voted perceptron is making distinctions that humans found difficult…".

Pande, Iyer et al. performs a detailed comparison of the different classifiers in two phases under two settings. In phase 1, the classifiers are made to distinguish between subjective and objective documents. In phase 2, the classifiers are made to classify positive from negative documents filtered by phase 1. In each phase classifiers have been tested without and with boosting to enhance performance. The classifiers tested are Bayesian Logistic Regression (BLR) with Gaussian and Laplacian prior, Naïve Bayes, Support Vector Machines with linear, polynomial and radial basis functions kernels and Voted Perceptrons.

Classifier	Avg F1	Avg Acc.	Max Acc.
Naïve Bayes	0.882	83.91	84.80
SVM(Linear)	0.880	83.29	85.736
SVM(Poly)	0.8571	80.80	82.946
SVM(RBF)	0.757	68.22	74.573
Voted Perceptron	0.875	82.96	85.43

Table 2.1: Accuracy Comparison of Different Classifiers (balanced Dataset)

Mullen et al. (2004) used Support Vector Machines with diverse information measures by using features like the PMI, Lemma, Turney, Osgood values along with other topic-oriented features.

It is observed that:

1. Using only Turney values, a high accuracy can be achieved.
2. The addition of Osgood values does not seem to yield improvement in any of the models.
3. Using only Lemmas instead of Unigrams result in a much better performance.
4. The inclusion of all PMI values with lemmas outperforms the use of only the Turney values, suggesting that the incorporation of the available topic relations is helpful.

The best performance is achieved by using Lemmas and PMI or Osgood Values.

Chapter 3

REQUIREMENT ANALYSIS, TOOLS &TECHNOLOGIES

3.1 Hardware and Software Requirements

3.1.1 Hardware Requirements:

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	Intel Core i3 processor
Processor Speed	1.70 GHz
RAM	4 GB
Storage Space	40 GB
Monitor Resolution	1024*768 or 1336*768 or 1280*1024

Table 3.1: Hardware Requirements

3.1.2 Software Requirements:

The software requirements are description of features and functionalities of the system.

Operating System	Windows 8.1
IDE	Anaconda
Libraries	Pandas, Matplotlib, Seaborn, Sklearn, NLTK, wordcloud

Table 3.2: Software Requirements

Chapter 4

Design And Implementation

4.1 System Architecture

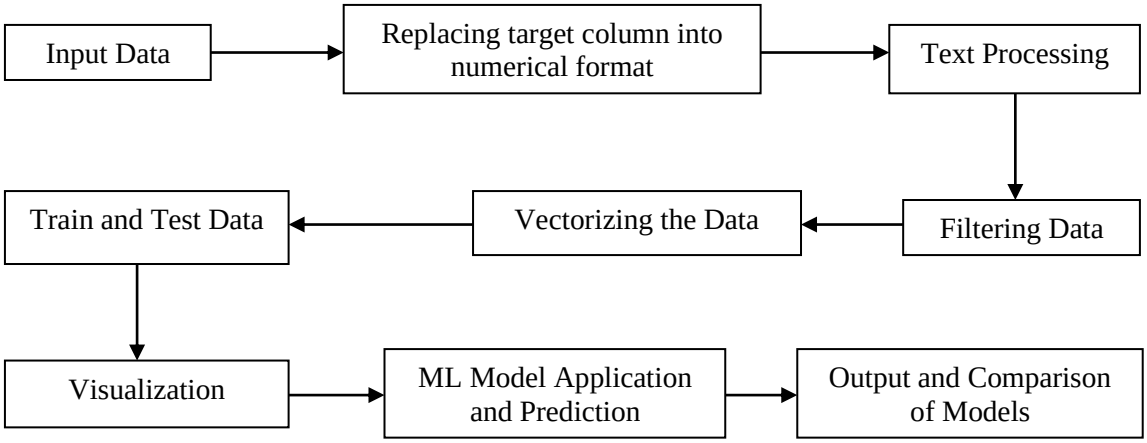

Figure 4.1: System Architecture

4.2 Data Analysis Life Cycle

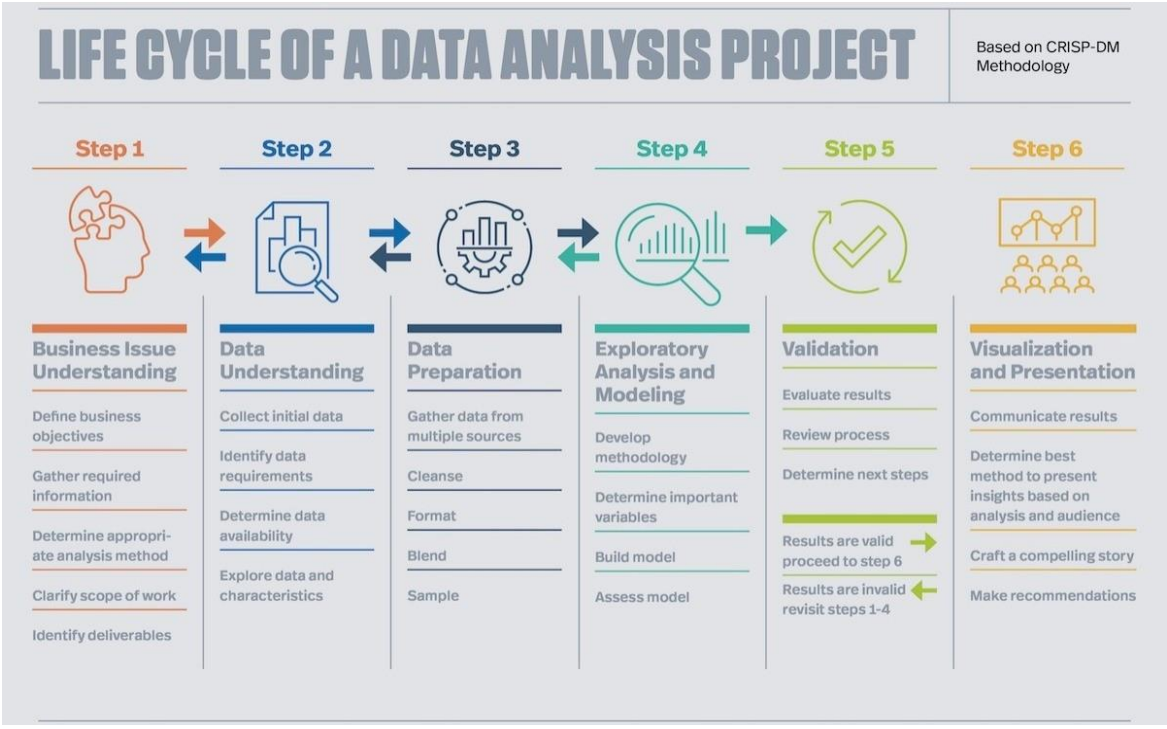

Figure 4.2: Data Analysis Life Cycle

The data analysis lifecycle describes the process of conducting a data analytics project, which consists of six key steps based on the CRISP-DM methodology. Data analysis is the process of examining data sets in order to find trends and draw conclusions about the information they contain. Increasingly, data analytics is done with the aid of specialized systems and software.

When presented with a data project, you will be given a brief outline of the expectations. From that outline, you should identify the key objectives that the business is trying to uncover. When presented with a small dataset, you can use tools like Excel, R, Python, Tableau Prep or Tableau Desktop to help prepare your data for its cleaning. Once you have organized and identified all the variables in your dataset, you can begin cleaning. Using different statistical modeling methods, you can determine which is the best. Interactive visualization tools like Tableau are tremendously useful in illustrating your conclusions to clients. Being able to tell a story with your data is essential.

4.3 Libraries

4.3.1 Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open-source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

Main Features:

- Easy handling of missing data (represented as NaN, NA, or NaT) in floating point as well as non-floating-point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensionalobjects.
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or theuser can simply ignore the labels and let Series, DataFrame, etc. automatically align the

data for you in computations.

- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data.
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets.
- Intuitive merging and joining data sets.
- Flexible reshaping and pivoting of data sets.
- Hierarchical labeling of axes (possible to have multiple labels per tick).
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, andsaving/loading data from the ultrafast HDF5 format.

4.3.2 Matplotlib

Matplotlib is a low-level graph plotting library in python that serves as a visualization utility. It was created by John D. Hunter. It is open source and we can use it freely. Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

Main Features:

- Creates publication quality plots.
- Makes interactive figures that can zoom, pan, update.
- Customizes visual style and layout.
- Export to many file formats.
- Can be embedded in JupyterLab and Graphical User Interfaces.
- Uses a rich array of third-party packages built on Matplotlib.

4.3.3 Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. It provides dataset- oriented APIs, so that we can switch between different visual representations for same variables.

Main Features:

- Built in themes for styling matplotlib graphics.
- Visualizing univariate and bivariate data.
- Fitting in and visualizing linear regression models.
- Seaborn works well with NumPy and Pandas data structures.
- It comes with built in themes for styling Matplotlib graphics.

4.3.4 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Main Features:

- Provides functionality to split dataset for training and testing. Splitting the data is essential for an unbiased evaluation of prediction performance.
- Provides numbers supervised ML models which can be used for analysis and prediction.
- Helps in clustering and feature extraction.

4.3.5 WordCloud

A word cloud is a simple yet powerful visual representation object for text processing, which shows the most frequent word with bigger and bolder letters, and with different colors. The smaller the the size of the word the lesser it’s important.

4.3.6 NLTK

The Natural Language Toolkit, or more commonly NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc… In this article, we will go through how we can set up NLTK in our system and use them for performing various NLP tasks during the text processing step.

Main Features:

- Tokenization
- Stopword removal
- Helps in sentiment analysis

4.4 Dataset

IMDB dataset having 50K 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. We provide a set of 25,000 highly polar movie reviews for training and 25,000 for testing. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.
For more dataset information, please go through the following link,
http://ai.stanford.edu/~amaas/data/sentiment/

	Review	Sentiment
2	One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hc	positive
3	A wonderful little production. The filming technique is very unassuming- very ol	positive
4	I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in t	positive
5	Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his	negative
6	Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei c	positive
7	Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a no	positive
8	I sure would like to see a resurrection of a up dated Seahunt series with the tech they have t	positive
9	This show was an amazing, fresh & innovative idea in the 70's when it first aired. The first 7 c	negative
10	Encouraged by the positive comments about this film on here I was looking forward to watcl	negative
11	If you like original gut wrenching laughter you will like this movie. If you are young or old th	positive
12	Phil the Alien is one of those quirky films where the humour is based around the oddness o	negative
13	I saw this movie when I was about 12 when it came out. I recall the scariest scene was the bi	negative
14	So im not a big fan of Boll's work but then again not many are. I enjoyed his movie Postal (m	negative
15	The cast played Shakespeare. Shakespeare lost. I appreciate that this	negative
16	This a fantastic movie of three prisoners who become famous. One of the actors is george cl	positive
17	Kind of drawn in by the erotic scenes, only to realize this was one of the most amateurish ar	negative
18	Some films just simply should not be remade. This is one of them. In and of itself it is not a b	positive
19	This movie made it into one of my top 10 most awful movies. Horrible. There wa	negative
20	I remember this film,it was the first film i had watched at the cinema the picture was dark in	positive

Figure 4.3: IMDB dataset of 50k Movie Review

4.5 Code Segment:

Importing Necessary Libraries:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from matplotlib import style
style.use('ggplot')
import re
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
```

Uploading and reading the dataset:

```
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('IMDB Dataset.csv')
df.head()
df.info()
```

Plotting the Sentiment Distribution:

```
sns.countplot(x='sentiment',data=df)
plt.title("Sentiment Distribution")
```

```
# Function to find number of words in each review


def no_of_words(text):
        words= text.split()
        word_count=len(words)
        return word_count
df['word count'] = df['review'].apply(no_of_words)
df.head()
```

```
#Replacing target column into numerical format i.e., +ve -> 1 and -ve -> 2


df.sentiment.replace("positive", 1, inplace=True)
df.sentiment.replace("negative", 2, inplace=True)
df.head()
```

Function for Text Processing:

```
# Text Processing
def data processing(text):
    text = text.lower()
```

```
# Removing Break Tags
    text = re.sub('<br />', '', text)
# Removing URLs
    text = re.sub(r"https\S+|www\S+|http\S+", '', text, flags = re.MULTILINE)
# Removing #s and @
    text = re.sub(r'\@w+|\#', '', text)
# Removing Punctuations
    text = re.sub(r'[^\w\s]', '', text)
text_tokens = word_tokenize(text)
# Removing Stopwords
    filtered_text = [w for w in text_tokens if not w in stop_words]
    return " ".join(filtered_text)


df.review = df['review'].apply(data_processing)

# Checking for Duplicates
duplicated_count = df.duplicated().sum()
print("Number of duplicate entries : ", duplicated_count)
```

Output: Number of duplicate entries: 421

```
# Dropping Duplicate data
df = df.drop_duplicates('review')

# Performing Stemming
stemmer = PorterStemmer()
def stemming(data):
    text = [stemmer.stem(word) for word in data]
    return data
df.review = df['review'].apply(lambda x: stemming(x))
df['word count'] = df['review'].apply(no_of_words)
df.head()
```

Visualization:

Code for word cloud visualization:

```
# Separating +ve reviews
pos_reviews = df[df.sentiment == 1]

#use word cloud to visualize +ve reviews
text = ' '.join([word for word in pos_reviews['review']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most frequent words in positive reviews', fontsize = 19)
plt.show()
```

Displays Wordcloud for Positive reviews

```
#Visualize most frequent words in the positive reviews
from collections import Counter
count = Counter()
for text in pos_reviews['review'].values:
  for word in text.split():
    count[word] += 1
count.most_common(15)
```

Outputs 15 most occurred word in positive review along with its frequency

```
pos_words = pd.DataFrame(count.most_common(15))
pos words.columns = ['word', 'count']
pos_words.head()
px.bar(pos_words, x='count', y='word', title='Common words in positive reviews', color = '
word')
```

Displays a bar chart containing common words in positive reviews

```
# Separating -ve reviews
neg_reviews = df[df.sentiment == 2]

# Use word cloud to visualize -ve reviews
text = ' '.join([word for word in neg_reviews['review']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most frequent words in negative reviews', fontsize = 19)
plt.show()
```

Displays Wordcloud for Negative reviews

```
#visualize most frequent words in the negative reviews
count = Counter()
for text in neg_reviews['review'].values:
  for word in text.split():
    count[word] += 1
count.most_common(15)
```

Outputs 15 most occurred word in negative review along with its frequency

```
neg_words = pd.DataFrame(count.most_common(15))
neg_words.columns = ['word', 'count']
neg_words.head()
px.bar(pos_words, x='count', y='word', title='Common words in negative reviews', color = '
word')
```

Displays a bar chart containing common words in negative reviews

Splitting data as X and Y for further processing:

```
X = df['review']
Y = df['sentiment']
```

```
# Vectorizing data
vect = TfidfVectorizer()
X = vect.fit_transform(df['review'])
```

Splitting into train and test dataset using sklearn:

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
```

Printing Size of the train and test dataset:

```
print("Size of x_train : ", (x_train.shape))
print("Size of x_test : ", (x_test.shape))
print("Size of y_train : ", (y_train.shape))
print("Size of y_test : ", (y_test.shape))
```

Output:

Size of x_train : (34704, 221707)
Size of x_test : (14874, 221707)
Size of y_train : (34704,)
Size of y_test : (14874,)

4.6 Algorithms:

4.6.1 Logistic Regression:

- Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- Logistic Regression is similar to linear regression, it is also used to make a prediction about a categorical variable versus a continuous one.

```
#Trainig Data with Logistic Regression
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_pred = logreg.predict(x_test)
logreg_acc = accuracy_score(logreg_pred, y_test)
print("Test Accuracy : {:.2f}%".format(logreg_acc*100))
```

Output: Test Accuracy: 89.00%

4.6.2 Multinomial Naïve Bayes Classifier:

- The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.
- When we have frequency as a feature, we can use Multinomial Naïve Bayes Classifier.

```
#Training data on a Multinomial Naive Bayes model
mnb = MultinomialNB()
mnb.fit(x_train, y_train)
mnb_pred = mnb.predict(x_test)
mnb_acc = accuracy_score(mnb_pred, y_test)
```

```
print("Test Accuracy : {:.2f}%".format(mnb_acc*100))
```

Output: Test Accuracy: 86.44%

4.6.3 Support Vector Classifier:

- Linear Support Vector Classification.
- Similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.
- This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.
- SVM is a fast and dependable classification algorithm that performs very well with a limited amount of data to analyze.

```
#training the data on a Support Vector Classifier Model

svc = LinearSVC()
svc.fit(x_train, y_train)
svc_pred = svc.predict(x_test)
svc_acc = accuracy_score(svc_pred, y_test)
print("Test Accuracy : {:.2f}%".format(svc_acc*100))
```

Output: Test Accuracy: 89.22%

```
#trying to increase accuraccy by performing hyper parameter tuning on the svc
# using GridSearchCV to loop through different parameters
from sklearn.model_selection import GridSearchCV
param_grid = {'C':[0.1, 1, 10, 100], 'loss':['hinge', 'squared_hinge']}
grid = GridSearchCV(svc, param_grid, refit=True, verbose = 3)
grid.fit(x_train, y_train)
#Printing the best cross violation and best parameter
print("Best cross validation score: {:.2f}".format(grid.best_score_))
print("Best Parameter: ", grid.best_params_)
```

Output:

Best cross validation score: 0.89
Best Parameter: {'C': 1, 'loss': 'hinge'}

```
# Applying obtained best parameter to SVC Model
svc = LinearSVC(C = 1, loss='hinge')
svc.fit(x_train, y_train)
svc_pred = svc.predict(x_test)
svc_acc = accuracy_score(svc_pred, y_test)
print("Test Accuraccy : {:.2f}%".format(svc_acc*100))
```

Output: Test Accuracy: 89.41%

Chapter 5

OBSERVATION AND RESULTS

5.1 Results:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Figure 5.1: output of df.head()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   review     50000 non-null  object
 1   sentiment  50000 non-null  object
dtypes: object(2)
memory usage: 781.4+ KB
```

Figure 5.2: output for df.info()

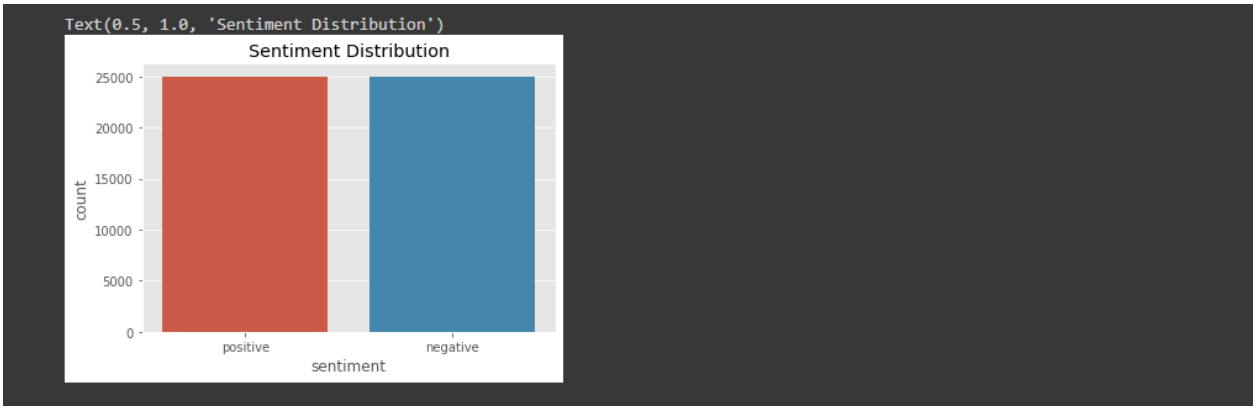

Figure 5.3: Sentiment distribution output

	review	sentiment	word count
0	One of the other reviewers has mentioned that ...	positive	307
1	A wonderful little production. The...	positive	162
2	I thought this was a wonderful way to spend ti...	positive	166
3	Basically there's a family where a little boy ...	negative	138
4	Petter Mattei's "Love in the Time of Money" is...	positive	230

Figure 5.4: Output for df.head() after applying no_of_words function to calculate the word count of each review

	review	sentiment	word count
0	One of the other reviewers has mentioned that ...	1	307
1	A wonderful little production. The...	1	162
2	I thought this was a wonderful way to spend ti...	1	166
3	Basically there's a family where a little boy ...	2	138
4	Petter Mattei's "Love in the Time of Money" is...	1	230

Figure 5.5: df.head() after replacing positive and negative with 1 and 2

	review	sentiment	word count
0	one reviewers mentioned watching 1 oz episode ...	1	168
1	wonderful little production filming technique ...	1	84
2	thought wonderful way spend time hot summer we...	1	86
3	basically theres family little boy jake thinks...	2	67
4	petter matteis love time money visually stunni...	1	125

Figure 5.6: df.head() after dropping duplicate entries and applying stemming

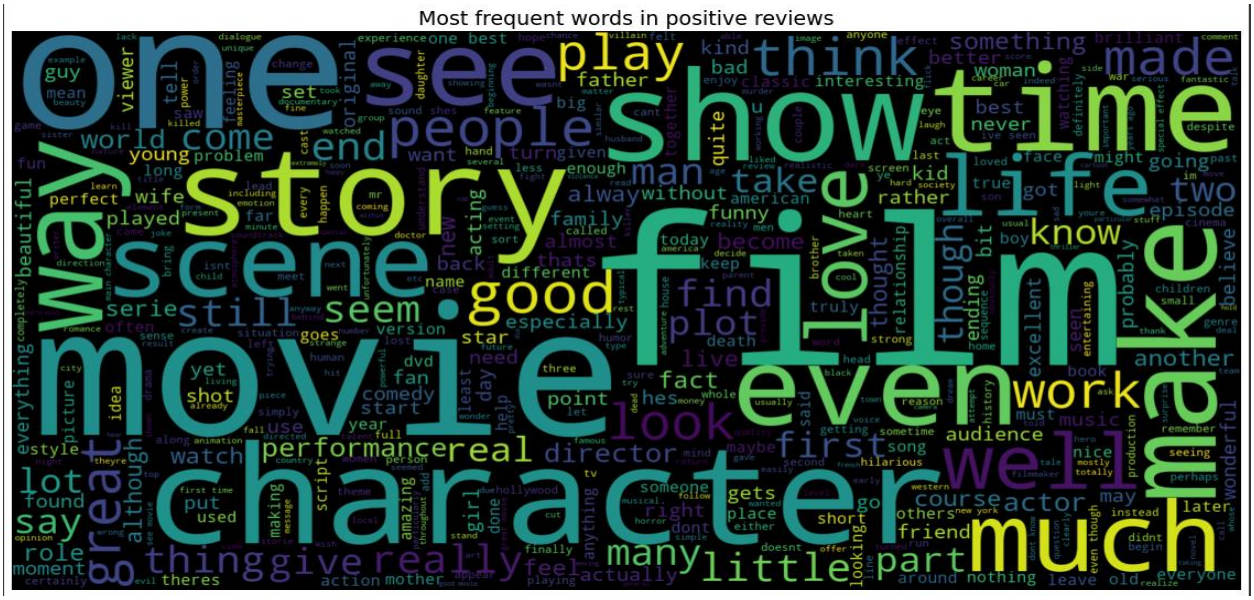

Figure 5.7: Wordcloud for most frequent words in positive review

```
[('film', 39285),
 ('movie', 35830),
 ('one', 25621),
 ('like', 16998),
 ('good', 14281),
 ('great', 12568),
 ('story', 12338),
 ('see', 11814),
 ('time', 11724),
 ('well', 10930),
 ('really', 10638),
 ('also', 10516),
 ('would', 10320),
 ('even', 9318),
 ('much', 8971)]
```

Figure 5.8: 15 most occurred word along with its frequency

	word	count
0	film	39285
1	movie	35830
2	one	25621
3	like	16998
4	good	14281

Figure 5.9: converting the above comma separated values into a data frame

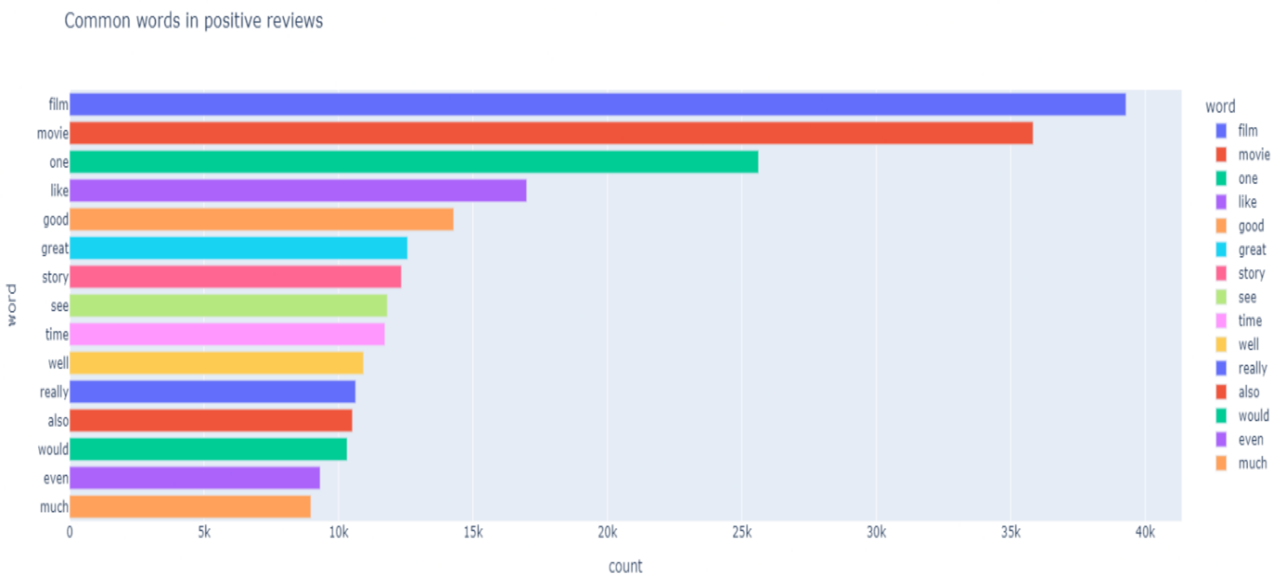

Figure 5.10: bar chart of common words in positive reviews

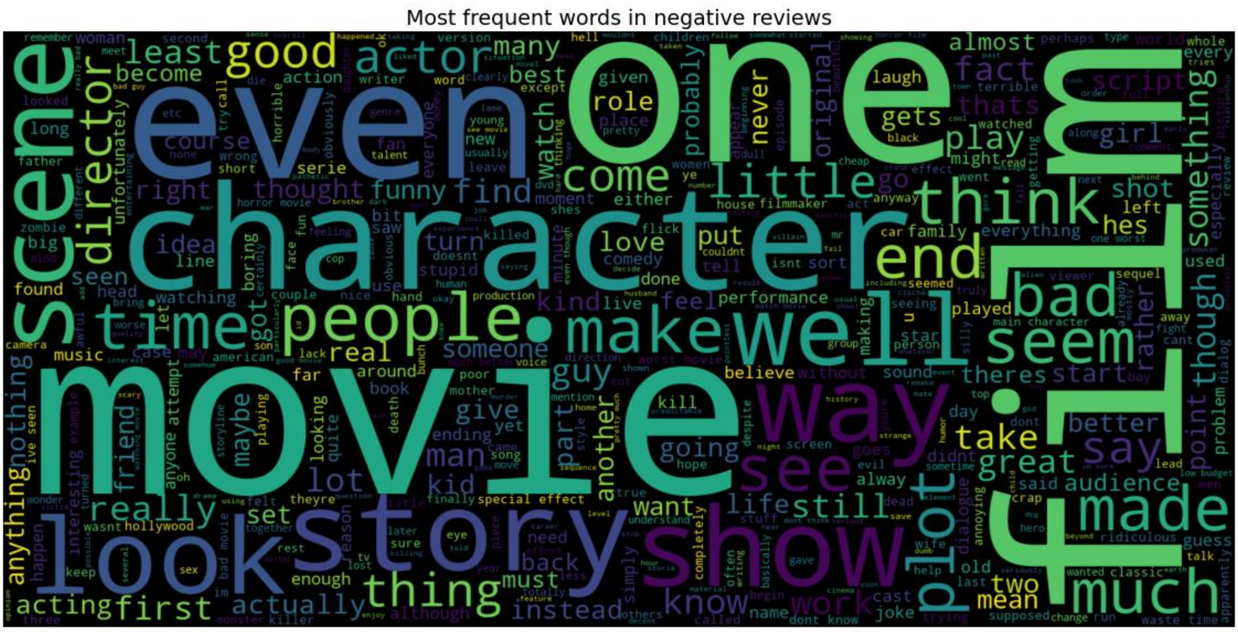

Figure 5.11: Wordcloud for most frequent words in negative review

```
[('movie', 47001),
 ('film', 34651),
 ('one', 24361),
 ('like', 21508),
 ('even', 14759),
 ('good', 13995),
 ('bad', 13903),
 ('would', 13482),
 ('really', 12084),
 ('time', 11349),
 ('see', 10412),
 ('dont', 9912),
 ('get', 9884),
 ('much', 9758),
 ('story', 9563)]
```

Figure 5.12: 15 most occurred word along with its frequency

	word	count
0	movie	47001
1	film	34651
2	one	24361
3	like	21508
4	even	14759

Figure 5.13: converting the above comma separated values into a data frame

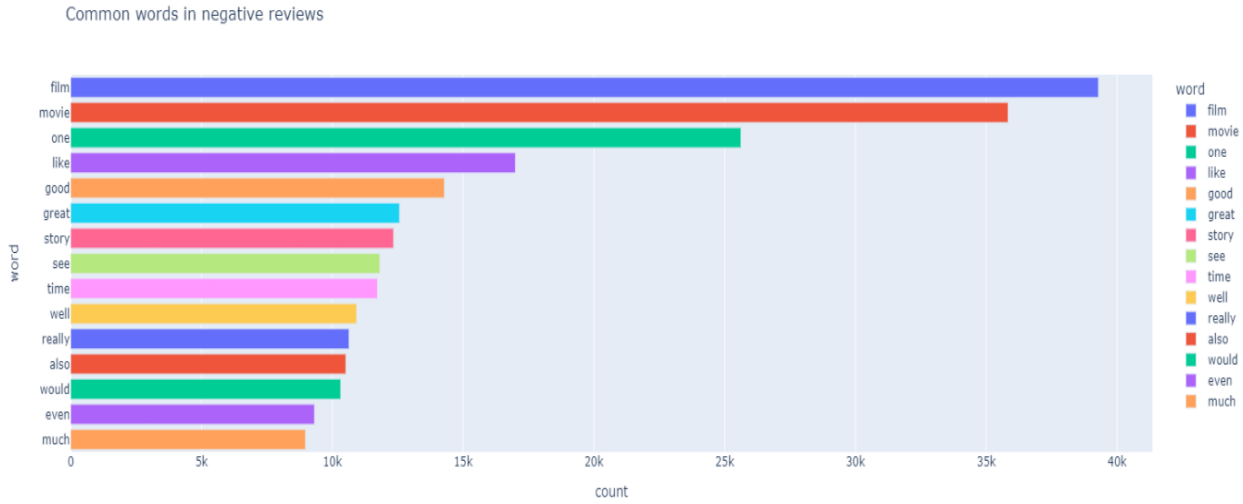

Figure 5.14: bar chart of common words in negative reviews

MODEL	ACCURACY
Logistic Regression	89.00%
Multinominal NB	86.44%
Linear SVC	89.22%
Tuned Linear SVC	89.41%

Table 5.1: Comparison of models and Accuracy Scores

5.2 Observations:

- On comparing 5.4 and 5.6 we get to see that on dropping duplicate entries there is a huge drop in word count.
- On observing the table 5.1, we can see than on tuning linear SVC, there is slight increase in the accuracy.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

- The dataset was trained with 4 models namely Logistic Regression, Multinomial NB, Linear SVC and Tuned Linear SVC successfully with highest accuracy of 89.41% which is much better than the previous models.
- The Tuned Linear SVC model returned a high precision, f1 score and recall score which are 0.90,0.89,0.88 respectively and predicted the results with high accuracy

6.2 Future Enhancement

- The model can be improved using LSTM, BERT models
- Model can also be improved using neural networks
- Size of dataset can be enhanced
- Real life data can be collected

REFERENCES

[1] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews?resource=download
[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[3] https://ml-course.github.io/master/intro.html
[4] https://matplotlib.org/stable/tutorials/index
[5] https://seaborn.pydata.org/tutorial.html
[6] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[7] https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/#:~:text=SVMs%20are%20used%20in%20applications,linear%20and%20non%2Dlinear%20data
[8] https://www.nltk.org/_modules/nltk/stem/porter.html
[9] https://pypi.org/project/wordcloud/
[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html