SENTIMENT ANALYSIS OF IMDb REVIEWS USING NLP

An
Industrial Oriented Mini Project Report

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

ARATLA ANANYA	**(19SS1A0503)**
POTHUGANTI SAIKRISHNA	**(19SS1A0538)**
PUSULURI VARNITHA	**(19SS1A0546)**
GANKIDI SAI VIVEK REDDY	**(20SS5A0504)**

Under the guidance of

Dr. P. Krupa sagar
Assistant Professor (C)

Department of Computer Science and Engineering

JNTUH University College of Engineering Sultanpur

Sultanpur (V), Pulkal (M), Sangareddy (Dist), Telangana-502273

January 2023

JNTUH UNIVERSITY COLLEGE OF ENGINEERING SULTANPUR

Sultanpur(V),Pulkal(M),Sangareddy-502273 ,Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that the Industrial Oriented Mini Project Report work entitled **”SENTIMENT ANALYSIS OF IMDb REVIEWS USING NLP”** is a bonafide work carried out by a team consisting of **ARATLA ANANYA** bearing Roll no.**19SS1A0503**, **POTHUGANTI SAIKRISHNA** bearing Roll no.**19SS1A0538**, **PUSULURI VARNITHA** bearing Roll no.**19SS1A0546**, **GANKIDI SAI VIVEK REDDY** bearing Roll no.**20SS5A0504**, in partial fulfillment of the requirements for the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to Jawaharlal Nehru Technological University Hyderabad University College of Engineering Sultanpur during the academic year 2022-2023.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Guide
Dr. P. KRUPA SAGAR
Assistant Professor (C)

Head of the Department
Dr. B. V. RAM NARESH YADAV
Professor,& HOD of CSE

External Examiner

Declaration

We hereby declare that the Industrial Oriented Mini Project entitled "**SENTIMENT ANALYSIS OF IMDb REVIEWS USING NLP**" is a bonafide work carried out by a team consisting of **ARATLA ANANYA** bearing Roll no.**19SS1A0503**, **POTHUGANTI SAIKRISHNA** bearing Roll no.**19SS1A0538**, **PUSULURI VARNITHA** bearing Roll no.**19SS1A0546**, **GANKIDI SAI VIVEK REDDY** bearing Roll no.**20SS5A0504**, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering discipline to Jawaharlal Nehru Technological University Hyderabad College of Engineering Sultanpur during the academic year 2022-2023. The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

ARATLA ANANYA	**(19SS1A0503)**
POTHUGANTI SAIKRISHNA	**(19SS1A0538)**
PUSULURI VARNITHA	**(19SS1A0546)**
GANKIDI SAI VIVEK REDDY	**(20SS5A0504)**

Acknowledgment

We wish to take this opportunity to express our deep gratitude to all those who helped us in various ways during our Industrial Oriented Mini Project report work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our Industrial Oriented Mini Project report.

We express our sincere gratitude to **Dr. G. Narsimha, Professor and Principal**, JNTUHUCES for his support during the course period.

We express our sincere gratitude to **Dr. Y. Raghavender Rao, Associate Professor, Vice Principal and Head of the Department(ECE)**, JNTUHUCES for his support during the course period.

We are thankful to **Dr. B. V. Ram Naresh Yadav, Professor, and Head of the Department**, Computer Science and Engineering, for his support and encouragement throughout the course period.

We are thankful to our Guide **Dr. P. Krupa sagar, Assistant Professor (C)** for his effective suggestions during the course period.

Finally, we express our gratitude with great admiration and respect to our faculty for their moral support and encouragement throughout the course.

ARATLA ANANYA	**(19SS1A0503)**
POTHUGANTI SAIKRISHNA	**(19SS1A0538)**
PUSULURI VARNITHA	**(19SS1A0546)**
GANKIDI SAI VIVEK REDDY	**(20SS5A0504)**

Contents

Abstract

Huge textual data is available on sites like Amazon, IMDB, and Rotten Tomatoes on movies, and analyzing such massive data manually is a tedious task. So, to speed up the process, programmers use certain techniques to extract out public opinion. One of which is using sentiment analysis. Sentiment analysis or Opinion Mining tools are essential to detect and understand customer feelings. Companies that use these tools to understand how customers feel can use them to improve customer Satisfaction. This is largely an NLP technique used by companies to mine data on the reviews left on their website by their own customers. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs. The objective of our project is to accurately extract people's opinions from a large number of IMDB review texts and classify them into sentiment classes, i.e., positive or negative. Sentiment analysis is done on the IMDb data set and we will manipulate the dataset using python libraries.

List of Figures

Chapter 1

INTRODUCTION

1.1 Project Overview

The sentiment analysis is an emerging research area where vast amounts of data are being analyzed, to generate useful insights in regards to a specific topic. It is an effective tool that can serve governments, corporations, and even consumers. Text emotion recognition plays a key role in this framework.

A subfield of Natural Language Processing and Conversational AI, Sentiment Analysis focuses on extracting meaningful user sentiments and assigning them scores through Natural Language Processing techniques. It helps understand user sentiments and opinions for a particular service and product. Hence, it heavily impacts improving the business logic and overall profit of an organization by bringing what their customers prefer. In this article, we will have a look at the sentiment analysis of IMDB Reviews with NLP. Sentiment Analysis can be carried out by text preprocessing using the standard NLP procedures and applying Language Understanding Algorithms to predict user sentiments.

Movie reviews are an important way to gauge the performance of a movie. While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer.

Sentiment Analysis[1] is a major subject in machine learning which aims to extract subjective information from the textual reviews. The field of sentiment of analysis is closely tied to natural language processing and text mining. It can be used to determine the attitude of the reviewer with respect to various topics or the overall polarity of review. Using sentiment analysis, we can find the state of mind of the reviewer while providing the review and understand if the person was “happy”, “sad”, “angry” and so on.

1.2 Purpose

To analyze each and every movie review that is taken from Kaggle (IMDb reviews dataset) and classify them based on their sentiment strengths into 2 classes: Positive and Negative respectively.

1.3 Existing System

Existing approaches to sentiment analysis can be grouped into three main categories: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques classify text by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid, and bored. Some knowledge bases not only list obvious affect words but also assign arbitrary words a probable emotion. Statistical methods leverage elements from machine learning such as support vector machines.

1.3.1 Drawbacks of Existing System

- Knowledge-based techniques do not require any training which it does not provide an accurate result.
- Some knowledge bases not only list obvious affect words but also assign arbitrary words a probable emotion.

1.4 Proposed System

The objective of our project is to accurately extract people's opinions from a large number of IMDB review texts and classify them into sentiment classes, i.e., positive or negative. We can easily test this model using any appropriate dataset using Naive Bayes Classifier and Python Flask framework. Naive Bayes is used as it increase model accuracy while using large amounts of data. Flask is a web framework that provides libraries to build lightweight web applications in python. Our model takes input from user-provided reviews, then splits them into individual words and assigns each word a unique integer number. Then we summarise the assortment of words to find either a positive or negative sentiment of that specific review.

1.4.1 Advantages of Proposed System

- Our model is a generalized model as it directly splits a sentence into individual words.
- Hence, no matter the sequence, it is nearly guaranteed to give a correct result.

1.5 Scope

The scope of the project is to estimate the sentiment of many movie reviews from the Internet Movie Database (IMDb) based on the content of the reviews, the dataset that has been pre-labeled with "positive" and "negative" sentiment class labels.

1.6 Conclusion

This project discusses the detailed approach to Sentiment Analysis, mainly using NLP. It provides a detailed view of the different applications and potential challenges of Sentiment Analysis that make it a difficult task. Sentiment analysis is becoming a crucial tool for monitoring and understanding audience sentiment as they share their opinions and emotions more openly than ever before. Movie makers can know what makes the audience satisfied or frustrated by automatically evaluating audience feedback.

Chapter 2

LITERATURE SURVEY

2.1 Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method

Authors: H M Keerthi Kumar, B S Harish, H. K. Darshan

This paper was proposed in January 2018 International Journal of Interactive Multimedia and Artificial Intelligence InPress

Classification, Hybrid Social Networking sites have become popular and common places for sharing a wide range of emotions through Features, Short texts, and short texts. These emotions include happiness, sadness, anxiety, fear, etc. Analyzing short texts helps in identifying Sentiment Analysis. the sentiment expressed by the crowd. Sentiment Analysis on IMDb movie reviews identifies the overall sentiment or opinion expressed by a reviewer towards a movie. Many researchers are working on pruning the sentiment analysis model that clearly identifies and distinguishes between a positive review and a negative review. In the proposed work, we show that the use of Hybrid features obtained by concatenating Machine Learning features (TF, TF-IDF) with Lexicon features (Positive-Negative word count, Connotation) gives better results both in terms of accuracy and complexity when tested against classifiers like SVM, Naïve Bayes, KNN and Maximum Entropy. The proposed model clearly differentiates between a positive review and a negative review. Since understanding the context of the reviews plays an important role in classification, using hybrid features helps in capturing the context of the movie reviews and hence increases the accuracy of classification.

2.2 Sentiment Analysis for Movie Reviews

Authors: Ankit Goyal,Amey Parulekar

The main aim[5] of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. In this project, we try to classify whether a person liked the movie or not based on the review they give for the movie. This is particularly useful in cases when the creator of a movie wants to measure its overall performance using reviews that critics and viewers are providing for the movie. The outcome of this project can also be used to create a recommender by providing recommendation of movies to viewers on the basis of their previous reviews. Another application of this project would be to find a group of viewers with similar movie tastes (likes or dislikes).

As a part of this project, we aim to study several feature extraction techniques used in text mining e.g. keyword spotting, lexical affinity and statistical methods, and understand their relevance to our problem. In addition to feature extraction, we also look into different classification techniques and explore how well they perform for different kinds of feature representations. We finally draw a conclusion regarding which combination of feature representations and classification techniques are most accurate for the current predictive task.

2.3 Keyword Extraction for Film Reviews based on Social Network Analysis and Natural Language Technology

Authors: Quan Yanan, Tang Fuqiang
Conference: 2020 E3S Web of Conferences

At present[4], there are many movie reviews appear on main stream websites, and these evaluations are quite different to the same movie. As a customer, how to choose your favorite movie and television program? To solve this problem, this study attempts to use the semantic analysis of word vectors (Word2vec) semantic analysis in machine learning as a research tool to mine a large number of movie reviews. The research shows that most movie reviews have a certain theme cohesion and their semantic network has quite connected. Through the use of social network analysis and the use of Word2vec word vector technology in natural language processing, it is possible to present a streamlined movie review based on movie review network semantics and keyword extraction, thus helping to select the favorite movie review.

2.4 Conclusion

Existing literature is summarized based on the content of the relevant research papers. The problem of Sentiment Analysis is a complex problem. Finally, the findings are identified in some research papers. The gist of the studied research papers is presented in brief. In particular, this paper provided the direction to solve the classification problem through various approaches.

Chapter 3

REQUIREMENT SPECIFICATION

3.1 Hardware and Software Requirements

Computer Hardware: Hardware refers to the physical components of a computer. Computer Hardware is any part of the computer that we can touch these parts. These are the primary electronic devices used to build up the computer. Examples of hardware in a computer are the Processor, Memory Devices, Monitor, Printer, Keyboard, Mouse, and the Central Processing Unit.

Hardware Requirements:

- System: Intel i3 processor and above
- Input devices: Mouse, Keyboard
- RAM: 4GB and above
- Hard disk: 512 GB

Computer Software: Software is a collection of instructions, procedures, and documentation that performs different tasks on a computer system. we can say also Computer Software is a program- ming code executed on a computer processor. The code can be machine-level code or the code written for an operating system. Examples of software are Ms Word, Excel, PowerPoint, Google Chrome, Photoshop, MySQL, etc.

Software Requirements:

- IDE: Jupyter (or) Python Ide (or) Visual Studio code (or) Google Colab
- OS: Windows7 and above
- Coding languages: Python

3.2 Functional Requirements

- Data Collection
- Data Preprocessing
- Training and Testing
- Modeling
- Predicting

3.3 Conclusion

Requirements specification is a must when it comes to developing software. Some good practices lead to good documentation. Since RS is useful for both customers and software development team, it is essential to develop a complete and clear requirements document. RS helps the customers to define their needs with accuracy, while it helps the development team understand what the customers need in terms of development.

Chapter 4

SYSTEM ANALYSIS

4.1 Architecture of the System

The Architecture of the system depicts how the actual process of the system is working. Initially, data is taken as input and then it is pre-processed. The system is trained with a given dataset further it uses the NLP to test the new data and produces output as the polarity of the review.

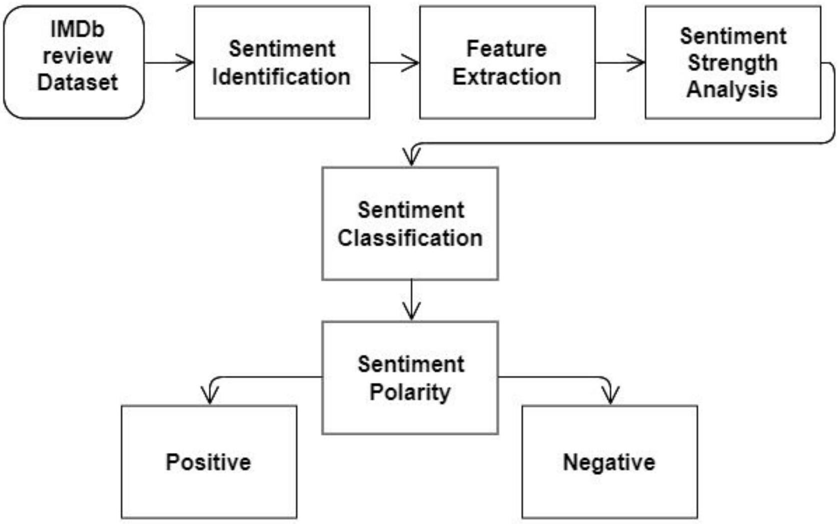


Figure 4.1: Architecture of the System

4.2 Feasibility Study

The next step in analysis is to verify the feasibility of the proposed system. ”All projects are feasible given unlimited resources and infinite time”. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in there consumption of resources. This places a constant are approval of any project. These feasibility study are 3 types:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

4.2.1 Technical Feasibility

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

- Technical Feasibility is the process of figuring out how you’re going to produce your product or service to determine whether it’s possible for your company.
- Before launching your offerings, you must plan every part of your operations, from first sourcing your production materials all the way to tracking your sales.
- By looking at all the logistics of this process, you can determine potential challenges and figure out ways to overcome them.
- Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

4.2.2 Operational Feasibility

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

4.2.3 Economic Feasibility

To decide whether a project is economically feasible, we have to consider various factors as:

- Cost-benefit analysis
- Long-term returns candidates appearing
- Maintenance costs

It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn't incur additional economic overheads, which renders the system economically feasible. The examination system being an online system should be available anytime.

4.3 Algorithm

4.3.1 Naive Bayes Classifier

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

4.3.2 Python Flask

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

Figure 4.2: Flask

4.4 Conclusion

We can say that system analysis is a problem solving strategy that includes glimpsing at the more extensive system, breaking the separated parts, and sorting out how it works to accomplish a specific objective. There are several definitions of system analysis like another definition is its examination of a specific system to observe the sectors of modifications and prepare any essential enhancements, if required.

Chapter 5

SYSTEM DESIGN

5.1 UML

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system. A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

UML Concepts: The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software intensive system

The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

Characteristics of UML:

The UML has the following features:

- It is a generalized modeling language.
- It is distinct from other programming languages like C++, Python, etc.
- It is interrelated to object-oriented analysis and design.
- It is used to visualize the workflow of the system.
- It is a pictorial language, used to generate powerful modeling artifacts.

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Building Blocks of the UML: The vocabulary of the UML encompasses three kinds of building blocks.

- **Things:** Things are the abstractions that are first-class citizens in a model
- **Relationships:** relationships tie these things together
- **Diagrams:** diagrams group interesting collections of things

5.2 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors. In this project, faculty and student are the actors.

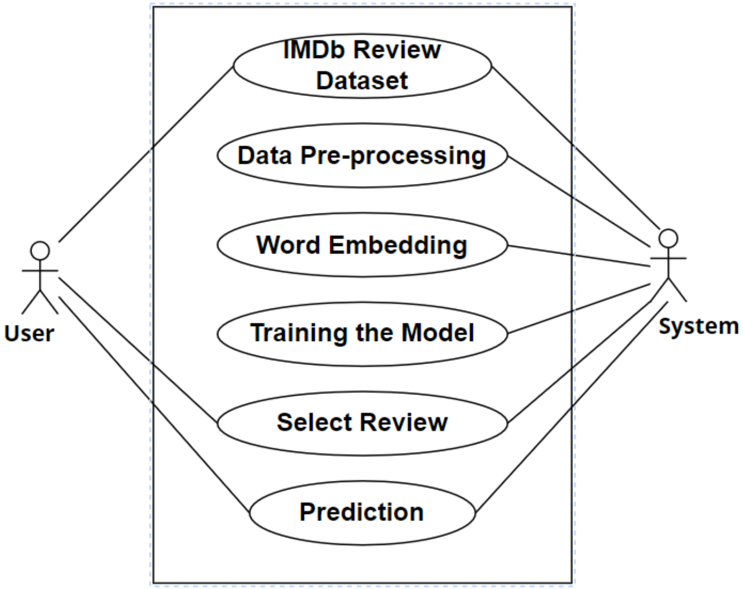


Figure 5.1: Use Case Diagram of System

5.3 Activity Diagram

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

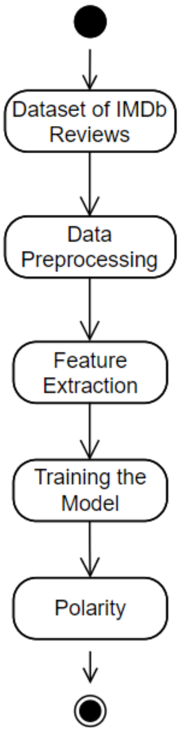


Figure 5.2: Activity Diagram of System

5.4 Sequence Diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

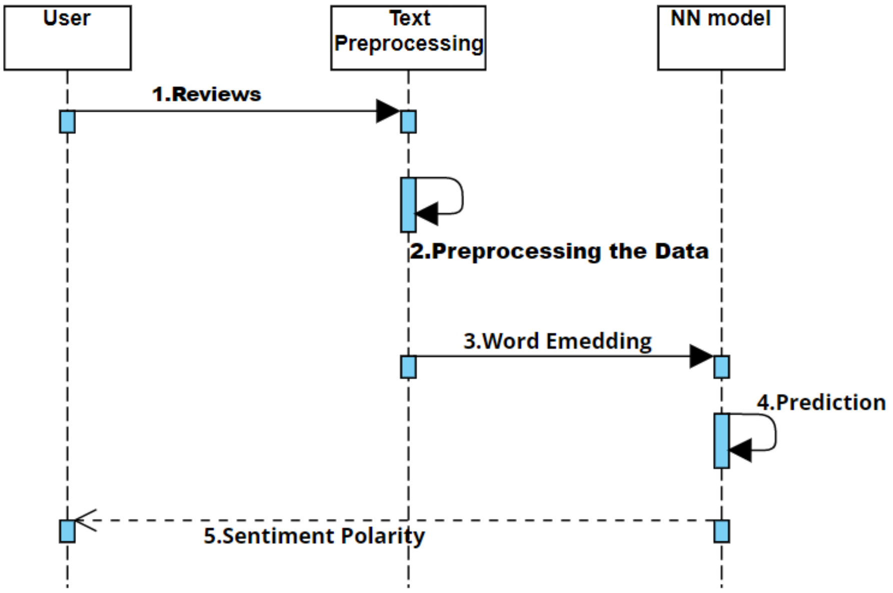


Figure 5.3: Sequence Diagram of System

5.5 Conclusion

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Chapter 6

IMPLEMENTATION

6.1 Coding

6.1.1 mini.ipynb

Import required Libraries:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score ,
classification_report , confusion_matrix
import pickle
import seaborn as sns
nltk.download('stopwords')
```

Load movies review data and show top 5 rows:

```
df = pd.read_csv("IMDB Dataset.csv")
df.head()
```

Perform EDA:

```
df.shape
df.isnull().sum()
df.describe()
df.info()
df['sentiment'].unique()
df['sentiment'].value_counts()
sns.countplot(df['sentiment'])
```

Apply LabelEncoding to make target feature into numerical (Positive : 1 , Negative : 0):

```
label = LabelEncoder()
df['sentiment'] = label.fit_transform(df['sentiment'])
df.head()
```

Divide data into independent and dependent:

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

Remove all special and numeric character from data and also remove stopwards and apply stemming:

```
ps = PorterStemmer()
corpus = []

for i in range(len(X)):
    print(i)
    review = re.sub("[^a-zA-Z]"," ", X[i])
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if word not in set(stopwords.word
    review = " ".join(review)
    corpus.append(review)
    corpus
```

Apply TfidfVectorizer to make text data into vectors:

```
from sklearn.feature_extraction.text import TfidfVectorizer
cv = TfidfVectorizer(max_features=5000)
X = cv.fit_transform(corpus).toarray()
X.shape
```

Split data into train and test:

```
X_train , X_test , Y_train , Y_test = train_test_split(X , y , test_size=0.2 ,
X_train.shape , X_test.shape , Y_train.shape , Y_test.shape
```

Define naive-bayes model:

```
mnb = MultinomialNB()
mnb.fit(X_train , Y_train)
```

test model using test data:

```
pred = mnb.predict(X_test)
```

Check accuracy_score, confusion_matrix and classification_report:

```
print(accuracy_score(Y_test , pred))
print(confusion_matrix(Y_test , pred))
print(classification_report(Y_test , pred))
```

Save my trained naive-bayes model and TfidfVectorizer:

```
pickle.dump(cv , open("count-Vectorizer.pkl" , "wb"))
pickle.dump(mnb , open("Movies_Review_Classification.pkl" , "wb"))


\textbf{\large Load my naive-bayes model and TfidfVectorizer:}
save_cv = pickle.load(open(’count-Vectorizer.pkl’,’rb’))
model = pickle.load(open(’Movies_Review_Classification.pkl’,’rb’))
```

Define my function to test model:

```
def test_model(sentence):
    sen = save_cv.transform([sentence]).toarray()
    res = model.predict(sen)[0]
    if res == 1:
        return ’Positive review’
    else:
        return ’Negative review’
```

Test review and check that what does model predicts

```
sen = ’This is the wonderful movie of my life’
res = test_model(sen)
print(res)
sen = ’This is the worst movie, I have ever seen in my life’
res = test_model(sen)
print(res)
```

6.1.2 app.py

```
# Importing essential libraries
from flask import Flask, render_template, request
import pickle

# Load the Naive Bayes model and TfidfVectorizer object from
disk
filename = 'Movies_Review_Classification.pkl'
classifier = pickle.load(open(filename, 'rb'))
cv = pickle.load(open('count-Vectorizer.pkl','rb'))
app = Flask(__name__)

@app.route('/')
def home():

    return render_template('home.html')

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        message = request.form['message']
        data = [message]
        vect = cv.transform(data).toarray()
        my_prediction = classifier.predict(vect)
        return render_template('result.html', prediction=my_prediction)

if __name__ == '__main__':
    app.run(debug=True)
```

6.1.3 Home.html

```
    <!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>Movies Reviews Classifier</title>
        <link rel="shortcut icon" href="{{ url_for
        ('static', filename='spam-favicon.ico') }}">
        <link rel="stylesheet" type="text/css" href="{{ url_for
        ('static', filename='styles.css') }}">
        <script src="https://kit.fontawesome.com/5f3f547070.js"
        crossorigin="anonymous"></script>
    </head>

    <body>

        <!-- Website Title -->
     <div class="container">
     <h2 class='container-heading'><span>Movies
                Reviews Classifier</span> </h2>
            <div class='description'>
     <p>A Machine Learning Web App which is implemented using Flask</p>
     </div>
     </div>

        <!-- Text Area -->
     <div class="ml-container">
     <form action="{{ url_for('predict') }}" method="POST">
         <textarea class='message-box' name="message" rows="15" cols="75"
                placeholder="Enter Your Message Here...
                (ex: This is the worst movie,
                i have ever seen in my life  OR  This is the wonderful movie,
                i have ever seen in my life)"></textarea><br/>
         <input type="submit" class="my-cta-button" value="Predict">
         </form>
     </div>
    </body>
</html>
```

6.1.4 result.html

```
  <!DOCTYPE html>

<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>Movies Reviews Classifier</title>
        <link rel="shortcut icon" href="{{ url_for('static',
        filename='spam-favicon.ico') }}">
        <link rel="stylesheet" type="text/css" href="{{
        url_for('static', filename='styles.css') }}">
        <script src="https://kit.fontawesome.com/5f3f547070.js"
        crossorigin="anonymous"></script>
    </head>

    <body>

        <!-- Website Title -->
     <div class="container">
     <h2 class='container-heading'><span>Movies
                Reviews Classifier</span> </h2>
            <div class='description'>
     <p>A Machine Learning Web App which is
                    implemented using Flask</p>
     </div>
     </div>

        <!-- Text Area -->
     <div class="ml-container">
     <form action="{{ url_for('predict') }}" method="POST">
         <textarea class='message-box' name="message"
          rows="15" cols="75" placeholder="Enter Your
```

```
}


          Message Here... (ex: This is the worst movie, i
          have ever seen in my life  OR  This is the wonderful movie, i have eve
          </textarea><br/>
         <input type="submit" class="my-cta-button"
          value="Predict">
         </form>
     </div>



    </body>
</html>
```

6.1.5 styles.css

```
html{
height: 100%;
margin: 0;
}

body{
font-family: Arial, Helvetica,sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
    width: 100%;
height: 100%;
display: flex;
   flex-direction: column;
}

/* Website Title */
.container{
```

```
padding: 30px;
position: relative;
background: linear-gradient(45deg, #ffffff, #ffffff, #f9f9f9, #eeeeee, #e0e4e1,
background-size: 500% 500%;
animation: change-gradient 10s ease-in-out infinite;
}
@keyframes change-gradient {
0%{
background-position: 0 50%;
}
50%{
background-position: 100% 50%;
}
100%{
background-position: 0 50%;
}
}

.container-heading{
    margin: 0;
}

.container span{
    color: #ff0000;
}

.description p{
    font-style: italic;
    font-size: 14px;
    margin: 3px 0 0;
}

/* Text Area */
.ml-container{
    margin: 30px 0;
flex: 1 0 auto;
}
```

```
.message-box{
    margin-bottom: 20px;
}

/* Predict Button */
.my-cta-button{
    background: #f9f9f9;
    border: 2px solid #000000;
    border-radius: 1000px;
    box-shadow: 3px 3px #8c8c8c;
    padding: 10px 36px;
    color: #000000;
    display: inline-block;
    font: italic bold 20px/1 "Calibri", sans-serif;
    text-align: center;
}

.my-cta-button:hover{
    color: #ff0000;
    border: 2px solid #ff0000;
}

.my-cta-button:active{
    box-shadow: 0 0;
}

.contact-icon{
color: #000000;
padding: 7px;
}



.contact-icon:hover{
color: #8c8c8c;
}
/* Result */
.results{
    padding: 30px 0 0;
flex: 1 0 auto;
}
.danger{
    color: #ff0000;
}

.safe{
    color: green;
}

.gif{
width: 30%;
}
```

6.2 Conclusion

This chapter constitutes the code of our project. The above way is way of our project implementation. We have written code for every page of the project.

Chapter 7

TESTING

7.1 White-Box Testing

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

7.2 Black-Box Testing

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

7.3 Unit Testing

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the[5] module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

7.4 Integration Testing

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

7.5 Validation Testing

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

7.6 System Testing

This testing is a series of different tests whose primary is to fully exercise the computerbased system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

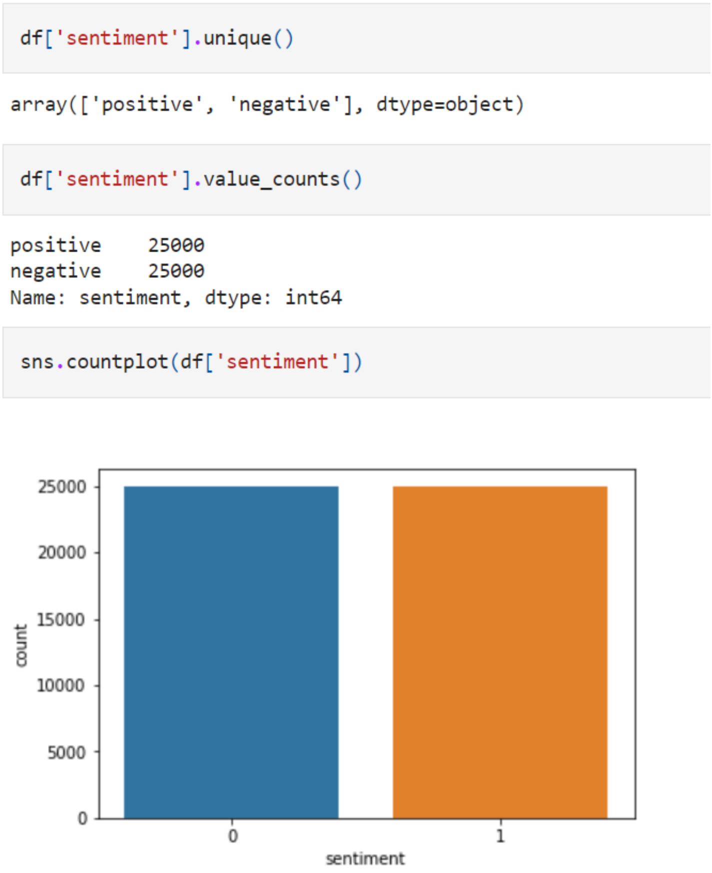
7.7 Conclusion

Software testing is an important part of the software development process. It is not a single activity that takes place after code implementation, but is part of each stage of the lifecycle. A Successful test strategy will begin with consideration during requirements specification.

Chapter 8

SCREENSHOTS

8.1 Exploratory Data Analysis



```
df['sentiment'].unique()
```

```
array(['positive', 'negative'], dtype=object)
```

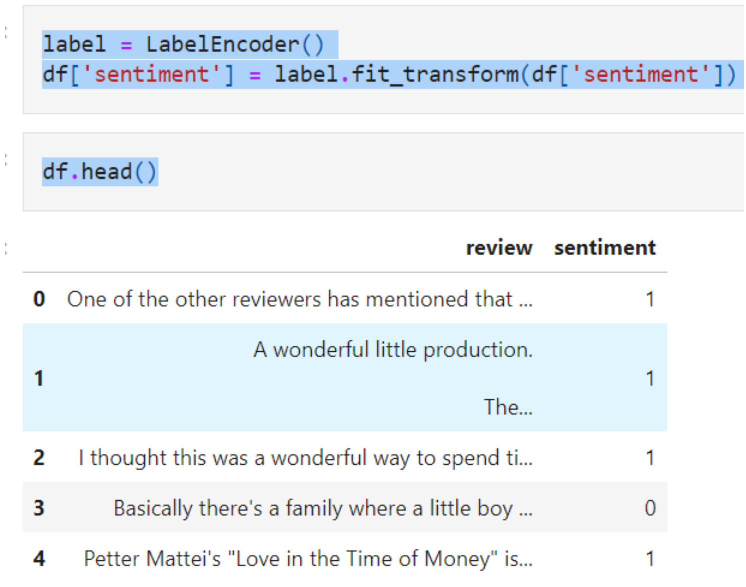
```
df['sentiment'].value_counts()
```

```
positive    25000
negative    25000
Name: sentiment, dtype: int64
```

```
sns.countplot(df['sentiment'])
```

Figure 8.1: EDA

8.2 Label Encoder of Reviews



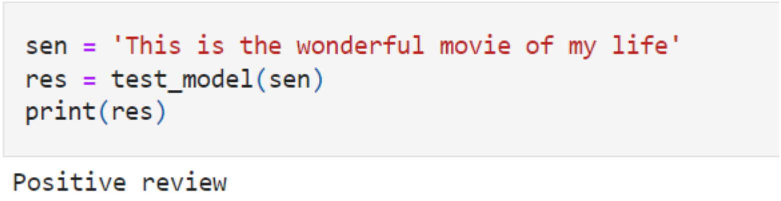
```
label = LabelEncoder()
df['sentiment'] = label.fit_transform(df['sentiment'])
```

```
df.head()
```

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

Figure 8.2: Label Encoding of Reviews

8.3 Review Testing for Positive

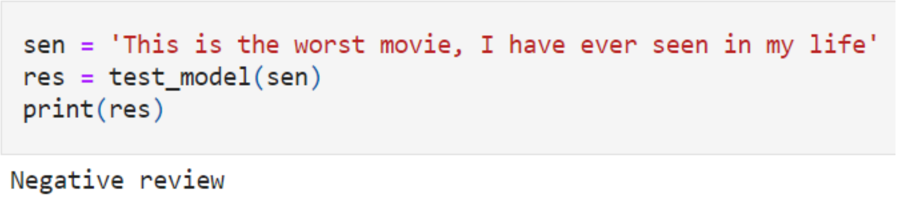


```
sen = 'This is the wonderful movie of my life'
res = test_model(sen)
print(res)
```

```
Positive review
```

Figure 8.3: Testing of Positive Review

8.4 Review Testing for Negative

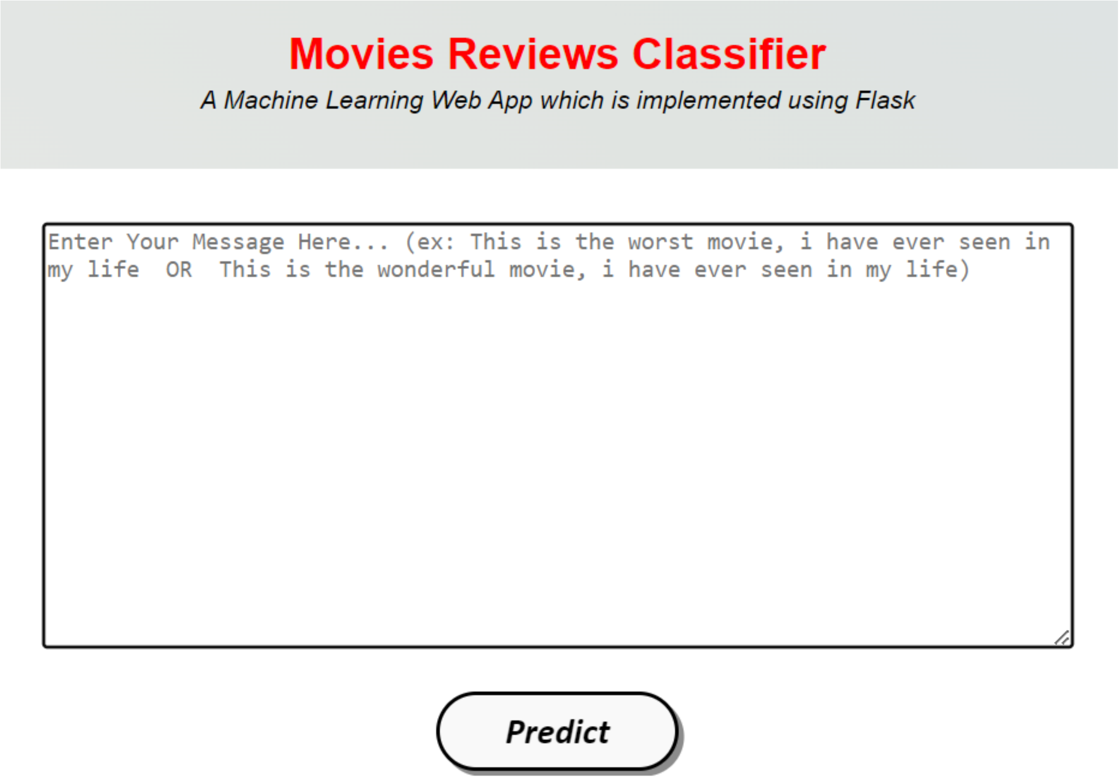


```
sen = 'This is the worst movie, I have ever seen in my life'
res = test_model(sen)
print(res)
```

```
Negative review
```

Figure 8.4: Testing of Negative Review

8.5 Home Page



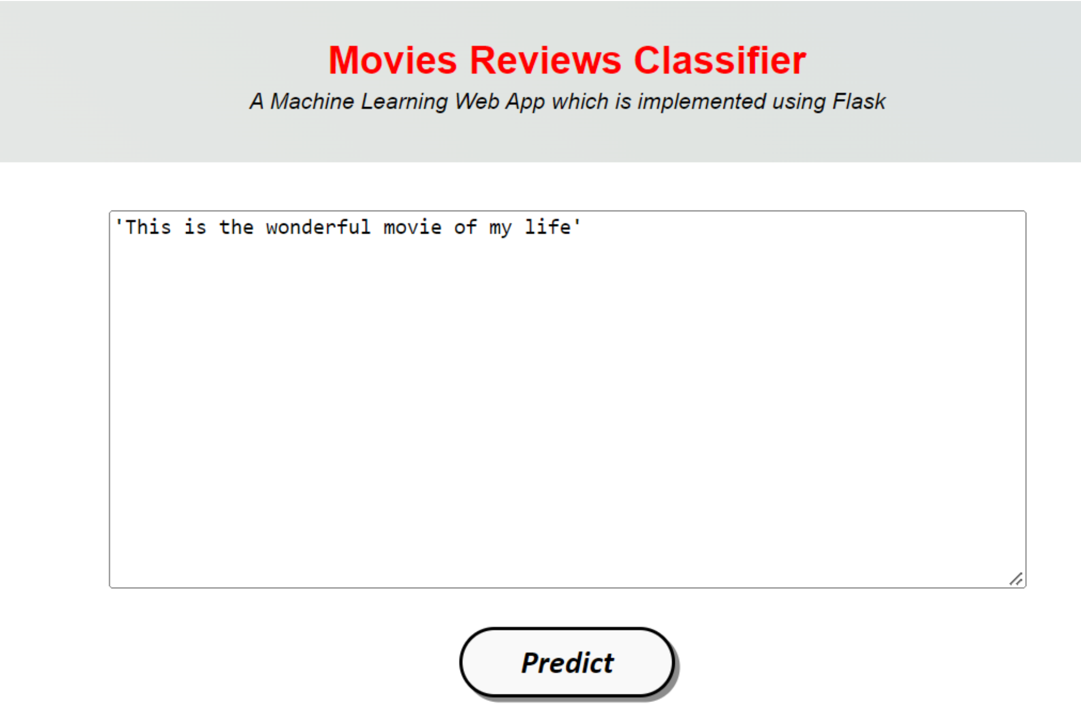
Movies Reviews Classifier

A Machine Learning Web App which is implemented using Flask

Enter Your Message Here... (ex: This is the worst movie, i have ever seen in my life OR This is the wonderful movie, i have ever seen in my life)

Figure 8.5: Home Page of Movie Review Classifier

8.6 Positive Review Input



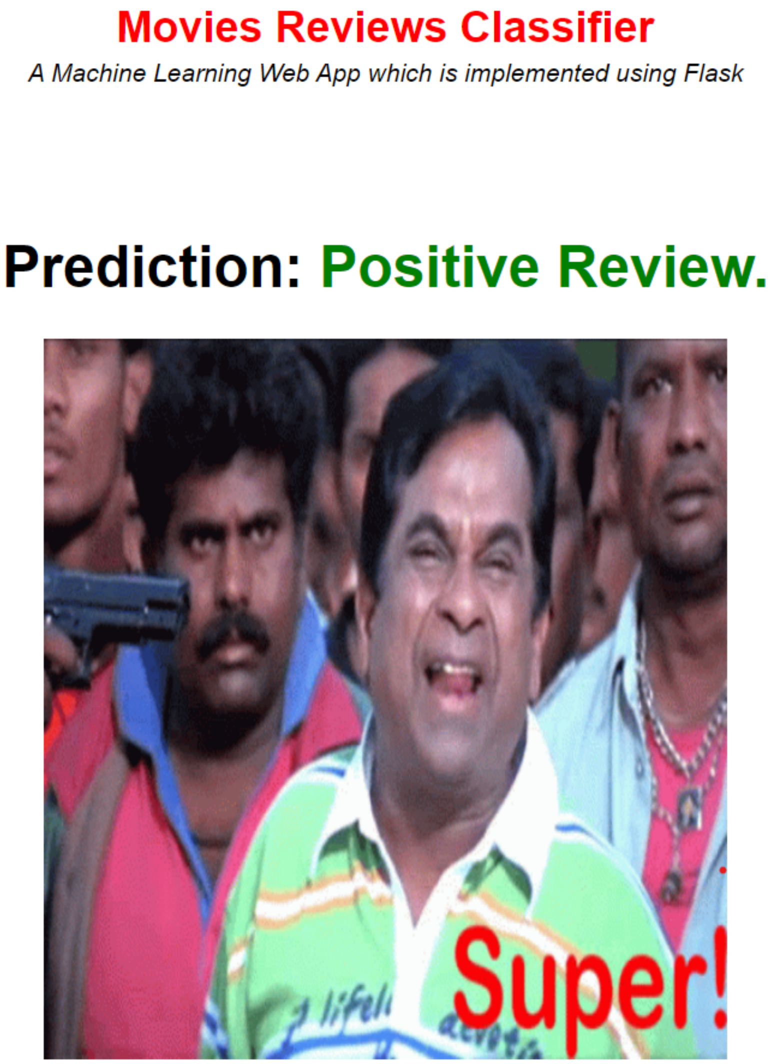
Movies Reviews Classifier

A Machine Learning Web App which is implemented using Flask

'This is the wonderful movie of my life'

Figure 8.6: Input of Positive Review

8.7 Positive Review Output



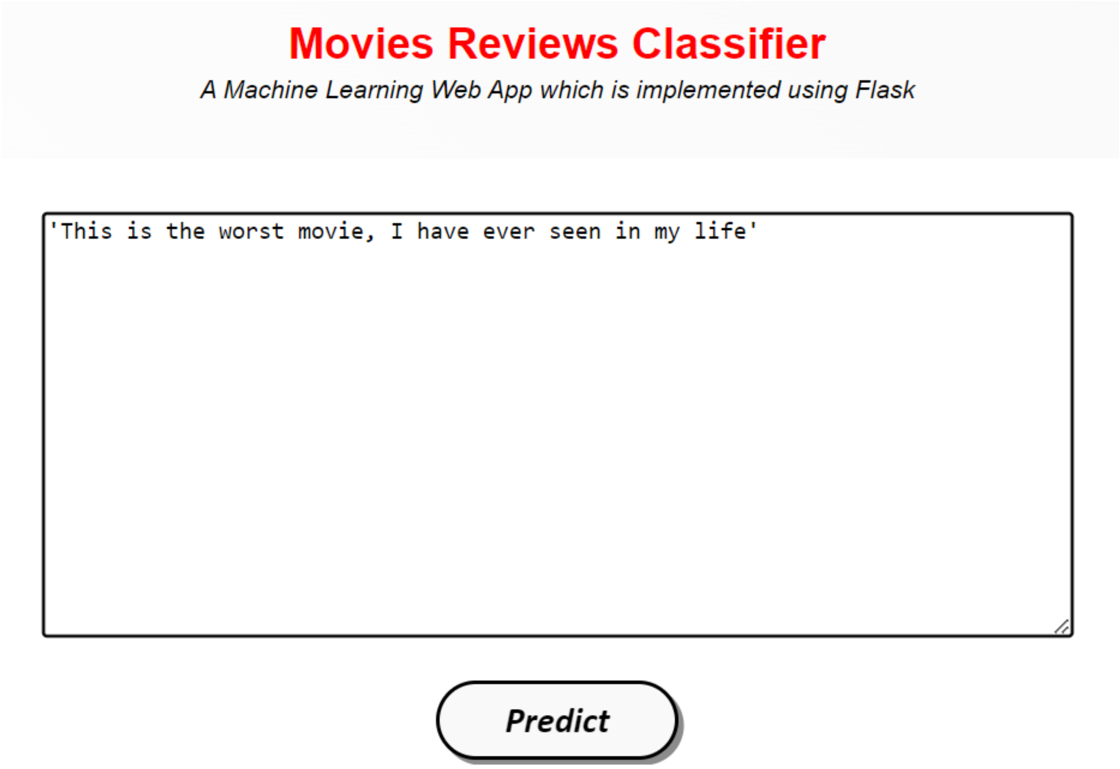
Movies Reviews Classifier

A Machine Learning Web App which is implemented using Flask

Prediction: Positive Review.

Figure 8.7: Output of Positive Review

8.8 Negative Review Input



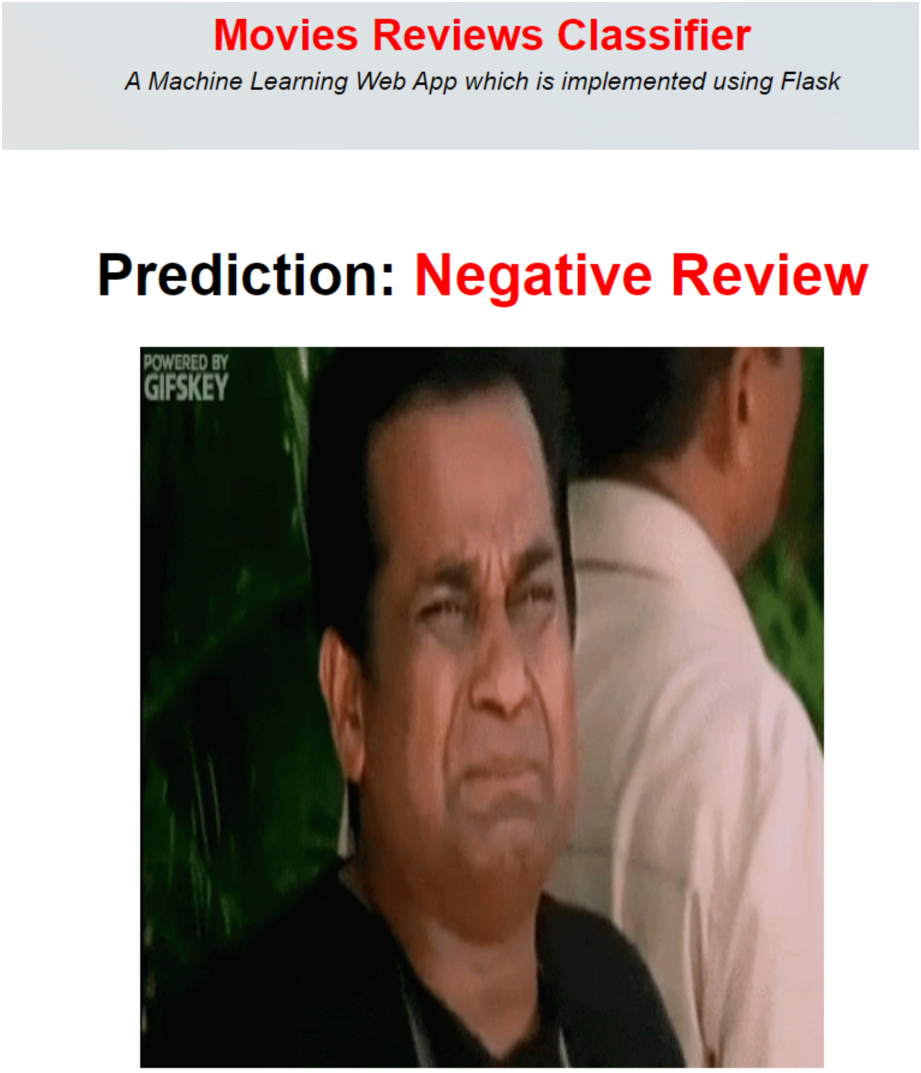
Movies Reviews Classifier

A Machine Learning Web App which is implemented using Flask

'This is the worst movie, I have ever seen in my life'

Figure 8.8: Input of Negative Review

8.9 Negative Review Output



Movies Reviews Classifier

A Machine Learning Web App which is implemented using Flask

Prediction: Negative Review

Figure 8.9: Output of Negative Review

8.10 Conclusion

The above images are the results of our project. Every page result is displayed in this-chapter. All the above figures show home page, result page, positive reviews and negative reviews.

CONCLUSION

We started with a brief introduction to the Sentiment Analysis and why it is required in the industries. Moving on, we applied a data pre-processing to our movie review dataset to remove the redundant expressions from the text .We implemented tokenization and Lemmatization to understand the context of those words used in the reviews and limit the recurring words .Further, we performed a feature extraction technique to convert text to number which is understandable to machine .Finally, we trained the Naive Bayes model to classify the reviews. In this project we aim to use Sentiment Analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. We aim to utilize the relationships of the words in the review to predict the overall polarity of the review.

REFERENCES

[1] *https://en.wikipedia.org/wiki/Sentiment_analysis*

[2] *https://www.ijimai.org/journal/bibcite/reference/2703/Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method*

[3] *https://ieeexplore.ieee.org/document/9257657S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 2020, pp. 1-4, doi: 10.1109/ICCIS49240.2020.9257657./*

[4] *https://www.semanticscholar.org/paper/Keyword-extraction-for-film-reviews-based-on-social-Yanan-Fuqiang/5da41fec8f9c8d0c99fae77b752df3e714d07867*

[5] *https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf*