Project Presentation

Text Classification

Date: 19/05/2023

Presented by: Vivek Sharma , Vishal Negi

Project Mentor: Mrs. Himani

Education & Training Division (ETD)
Centre for Development of Advanced Computing (C-DAC)
(Ministry of Electronics & Information Technology, Govt. of India)
A-34, Phase-VIII, Industrial Area, Mohali (160071)

TABLE OF CONTENTS

- Introduction (2 slides)
- Problem statement (1 slide)
- Objective or Problem formulation (1 slide)
- Libraries Used (1 slide)
- Design Methodology (Block Diagram or Work flow) – (2 slides)
- Dataset Explanation – (1 slide)
- Results – Front end clips, back end design clips, data base etc
- Conclusion (few lines)
- References

INTRODUCTION

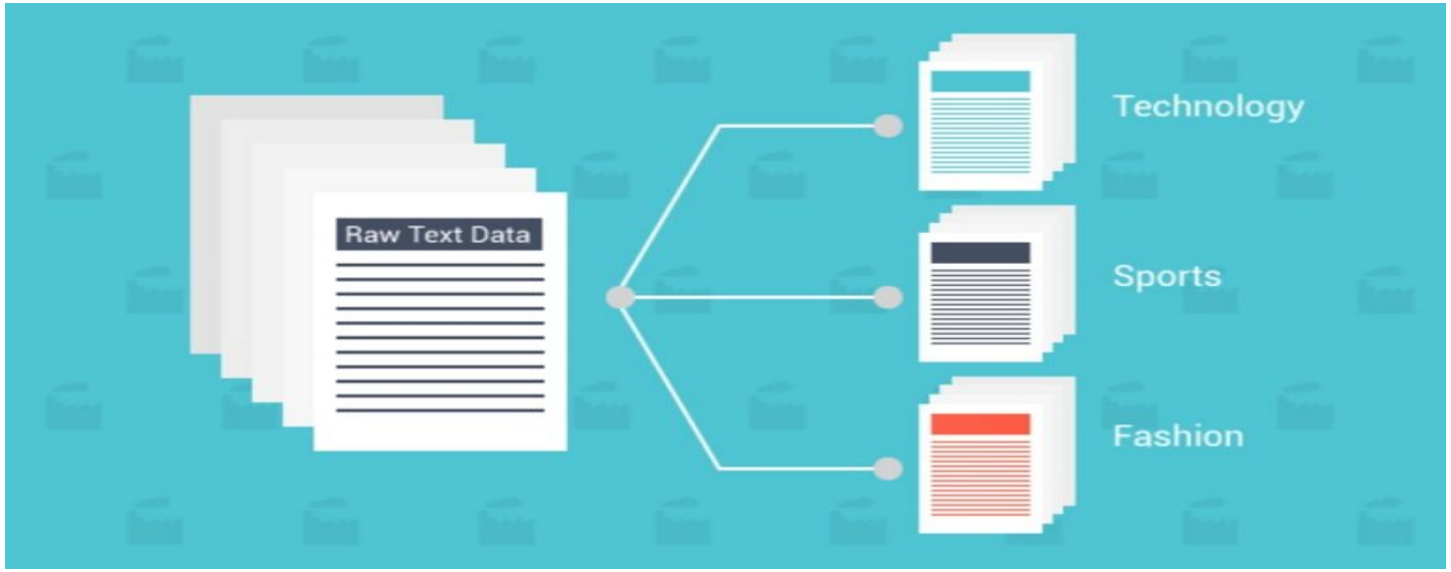
- Text classification is a natural language processing (NLP) task that involves assigning a category to a piece of text. In the case of movie reviews, the categories would be positive, negative, or neutral.

- The goal of this report is to demonstrate how text classification can be used to analyze movie reviews. By classifying movie reviews, we can gain insights into the public's perception of movies. This information can be used by movie studios to make better decisions about which movies to produce and market.

- This report aims to explore the application of text classification techniques in analyzing movie reviews and more specifically sentiment analysis.

INTRODUCTION

Movie reviews play a crucial role in providing insights and recommendations to potential viewers.
With the exponential growth of online platforms and social media, the volume of movie reviews has skyrocketed, making it challenging for individuals to manually process and analyze the vast amount of textual data.
This report aims to explore the application of text classification techniques in analyzing movie reviews and, more specifically, sentiment analysis.



PROBLEM STATEMENT

- Problem Statement: The project that the proposal infers to is called "**Movie Review Sentiment Analysis**". The main goal is to classify the sentiment of reviews from "IMBD 50K reviews" dataset. The movie review dataset "IMBD 50K reviews" is a corpus of movie reviews used for sentiment analysis.

- The problem at hand is to develop a system for movie review analysis. Given a set of movie reviews, the objective is to automatically classify them as **positive** or **negative** based on the sentiment expressed in the text.

OBJECTIVE OR PROBLEM FORMULATION

- The objective of movie review analysis is to develop a system that can automatically analyze and classify movie reviews based on their sentiment. The primary goal is to determine whether a review expresses a positive or negative sentiment towards the movie.
- **Formally, the problem can be defined as follows:**
- Given a dataset of movie reviews labeled with their corresponding sentiments (positive, negative), the objective is to build a predictive model that can accurately classify new, unseen movie reviews into one of these sentiment categories.
- The model should take into account the textual content of the reviews and learn the underlying patterns and features that differentiate positive negative, sentiments.

LIBRARIES USED

Pandas : Pandas is a powerful open-source data manipulation and analysis library for Python.

Numpy : NumPy (Numerical Python) is a powerful open-source library for numerical computing in Python.

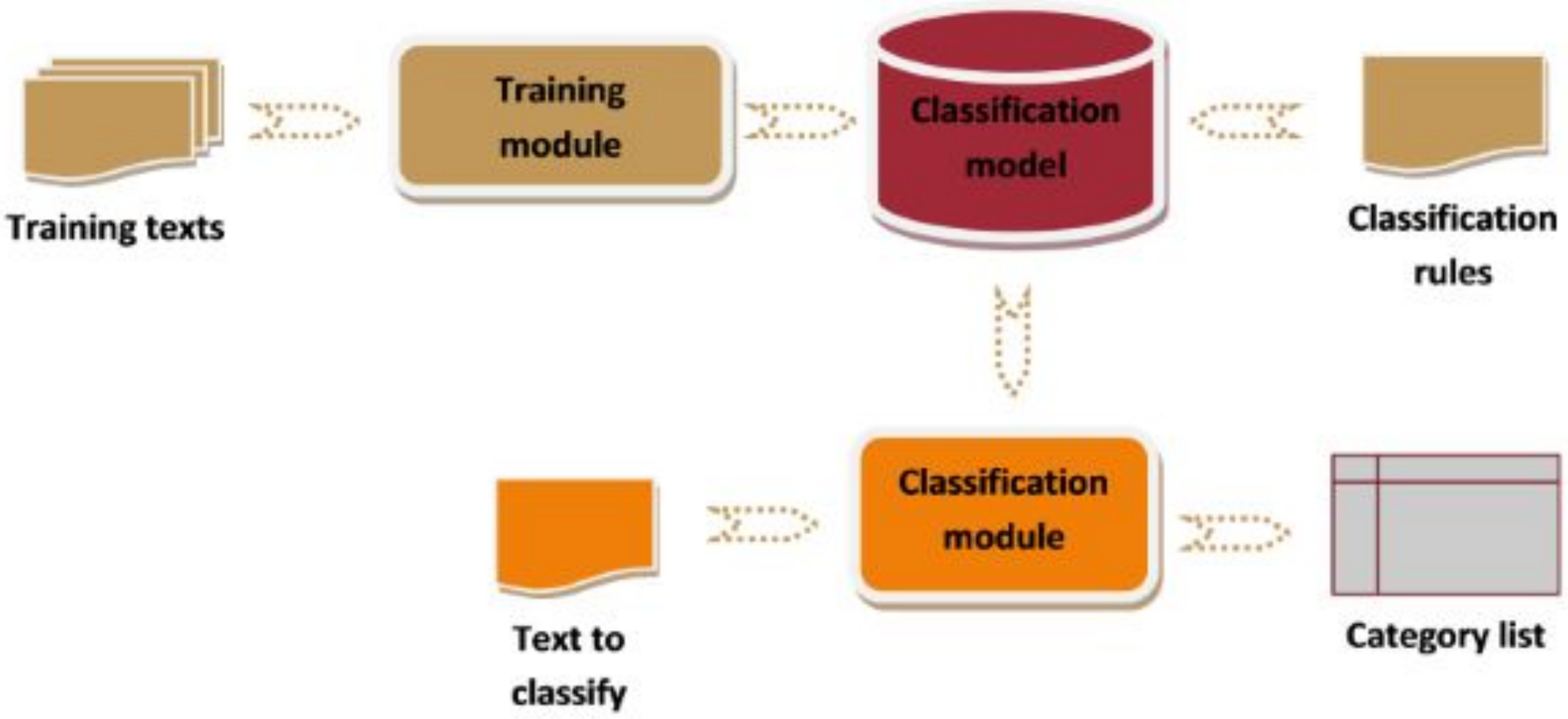
Re : This is the built-in regular expression library in Python used for pattern matching and manipulation of strings.

String : The str type provides a range of built-in methods for manipulating and working with strings.

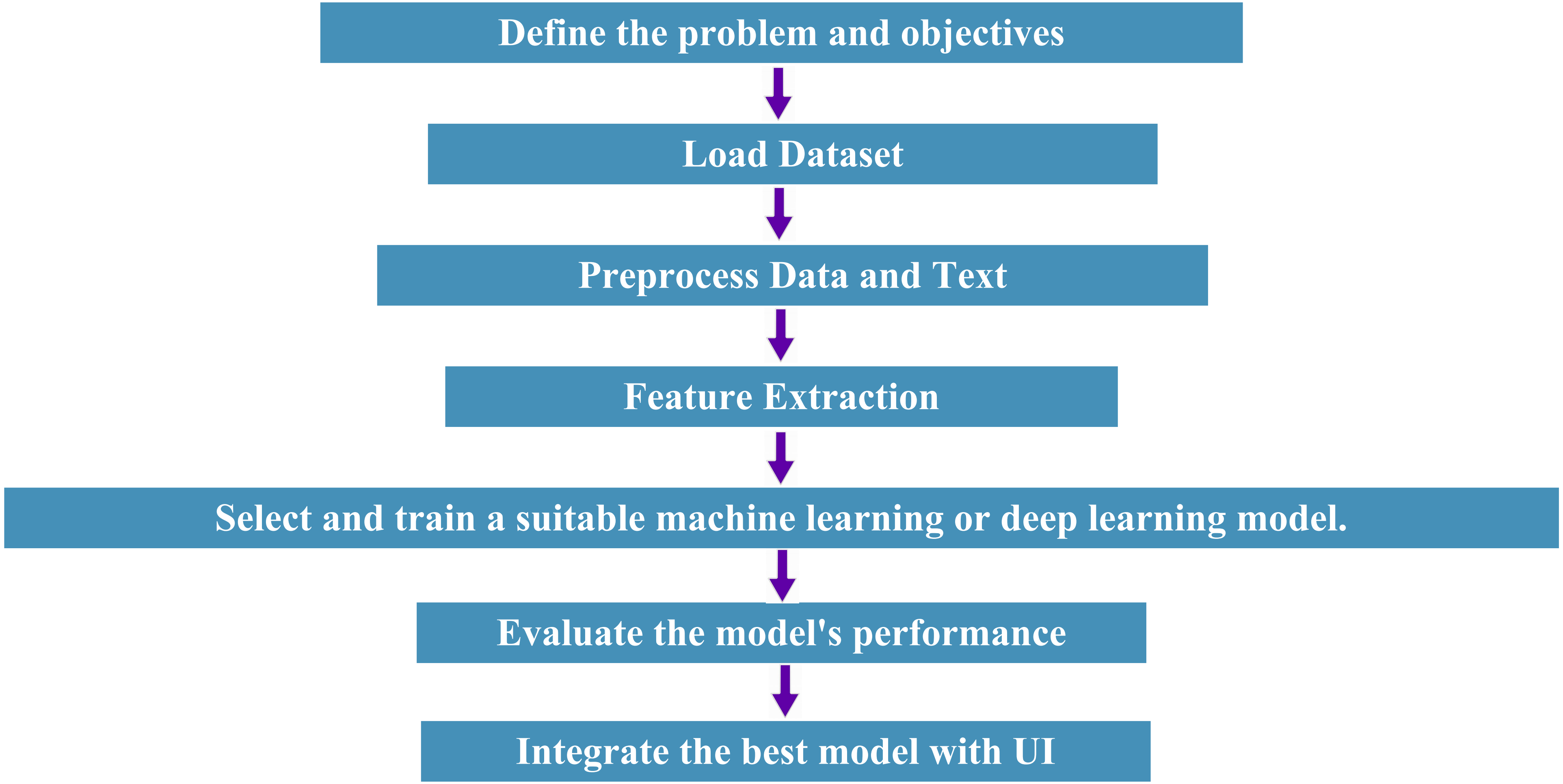
Scikit-learn : The scikit-learn library or sklearn is a widely used machine learning library in Python. It provides a rich set of tools for data preprocessing, feature engineering, model selection, and evaluation.

Pickle : Pickle allows you to convert Python objects, such as lists, dictionaries, and custom classes, into a byte stream (serialization), and later reconstruct the objects from the byte stream (deserialization).

DESIGN METHODOLOGY



DESIGN METHODOLOGY



Define the problem and objectives

Load Dataset

Preprocess Data and Text

Feature Extraction

Select and train a suitable machine learning or deep learning model.

Evaluate the model's performance

Integrate the best model with UI

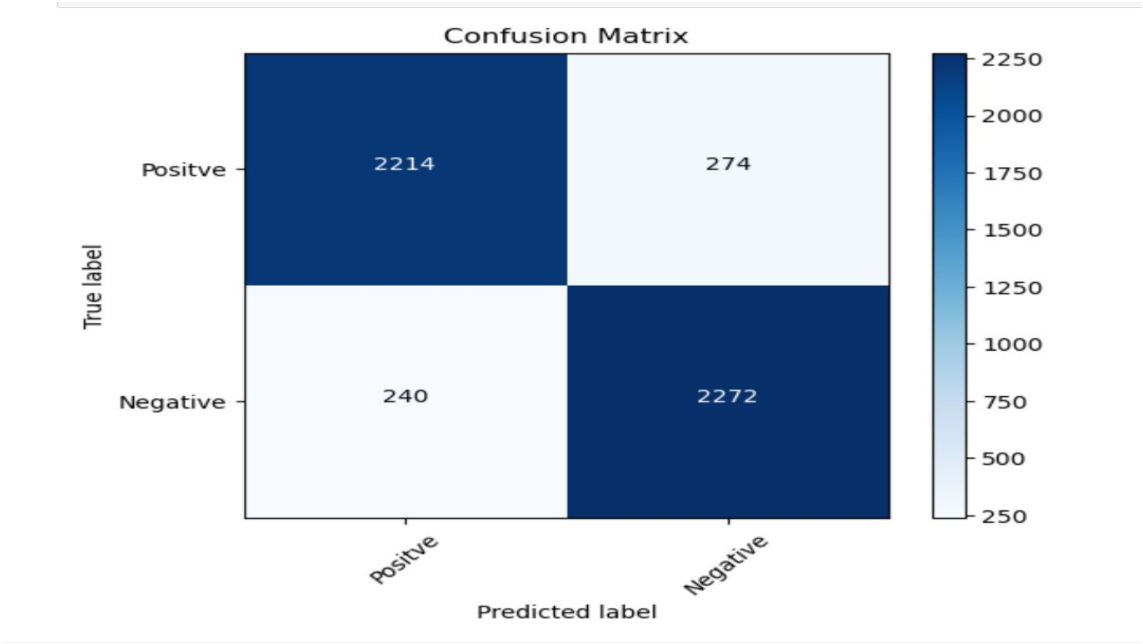
SOFTWARE

- **Jupyter Notebook** is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and explanatory text.
- **Anaconda** is a popular distribution of Python and other open-source packages commonly used for data science, machine learning, and scientific computing. It includes the Python programming language, along with a comprehensive collection of pre-installed packages and tools that simplify package management and environment setup.
- **VS Code**, short for Visual Studio Code, is a popular source code editor developed by Microsoft. It is highly extensible and supports various programming languages, making it a versatile choice for software development.

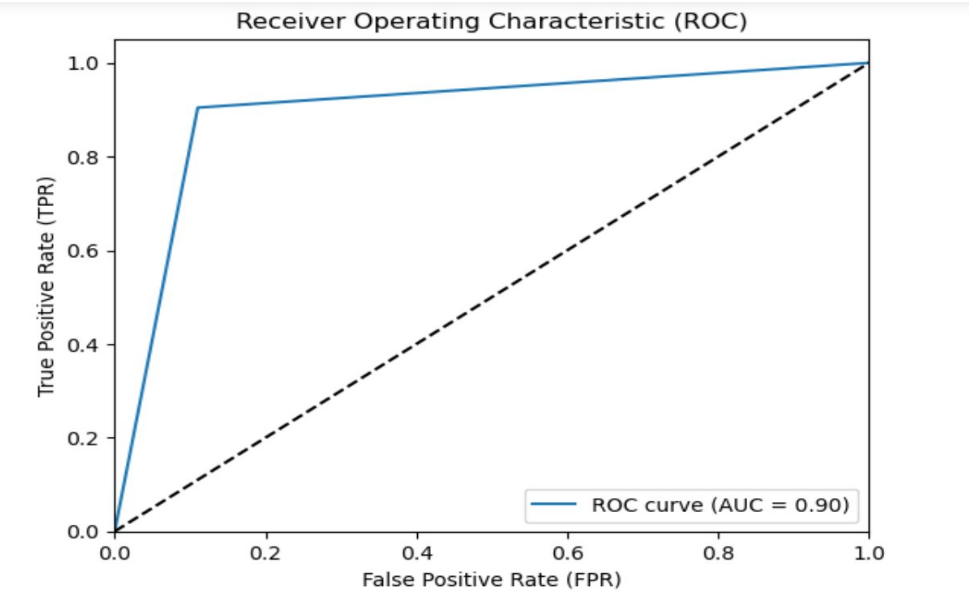
DATASET EXPLANATION

- The IMDb movie review dataset is a widely used dataset that consists of movie reviews and their corresponding sentiment labels. It is commonly used for sentiment analysis tasks, where the goal is to classify movie reviews as positive or negative based on the expressed sentiment.
- The IMDb movie review dataset typically includes the following components:
- **Movie Reviews:** The dataset contains a collection of movie reviews written by IMDb users. These reviews are typically in text format and represent the opinions and experiences of the users regarding specific movies.
- **Sentiment Labels:** Each movie review in the dataset is associated with a sentiment label indicating whether the review expresses a positive or negative sentiment. This label is assigned based on the overall sentiment conveyed in the text of the review.

RESULTS



Confusion Matrix
True label
Positve
Negative
2214
274
240
2272
2250
2000
1750
1500
1250
1000
750
500
250
Positve
Negative
Predicted label



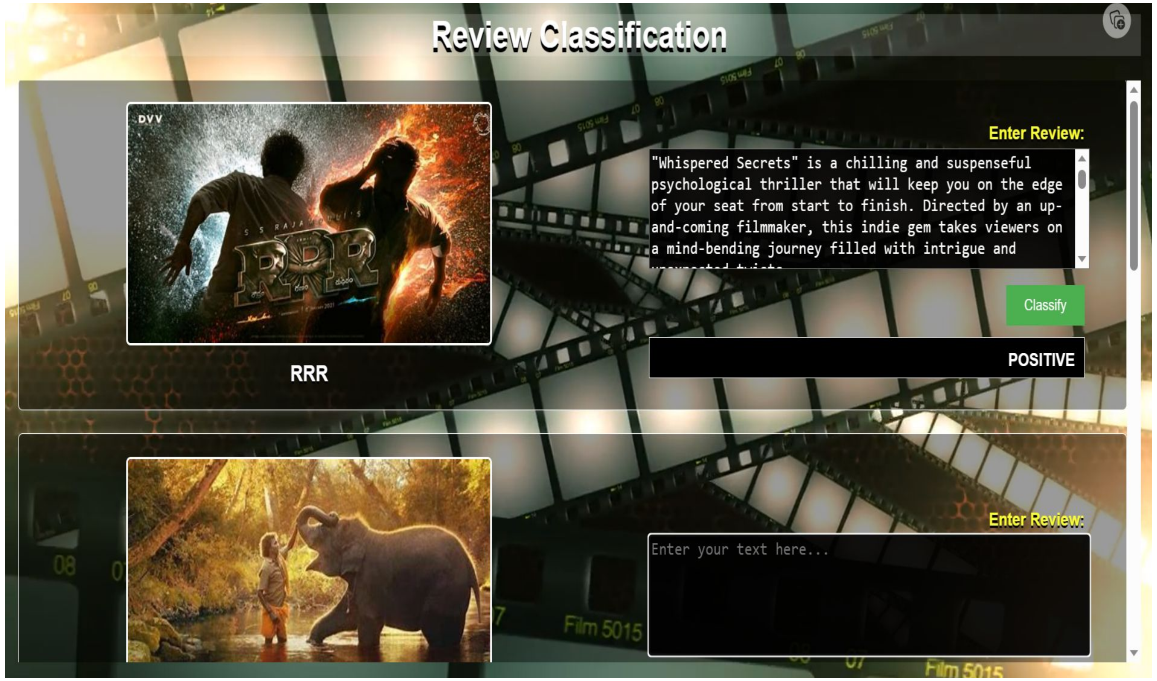
Receiver Operating Characteristic (ROC)
True Positive Rate (TPR)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (FPR)
ROC curve (AUC = 0.90)

Accuracy Table

	Tf-Idf Vectorizer	Count Vectorizer
Logistic Regression	89.72	89.62
K-Neighbors Classifier	73.7	61.8
Decision Tree	70.58	72.4
LSTM	86	-
MLP	87.84	-
CNN	89	-

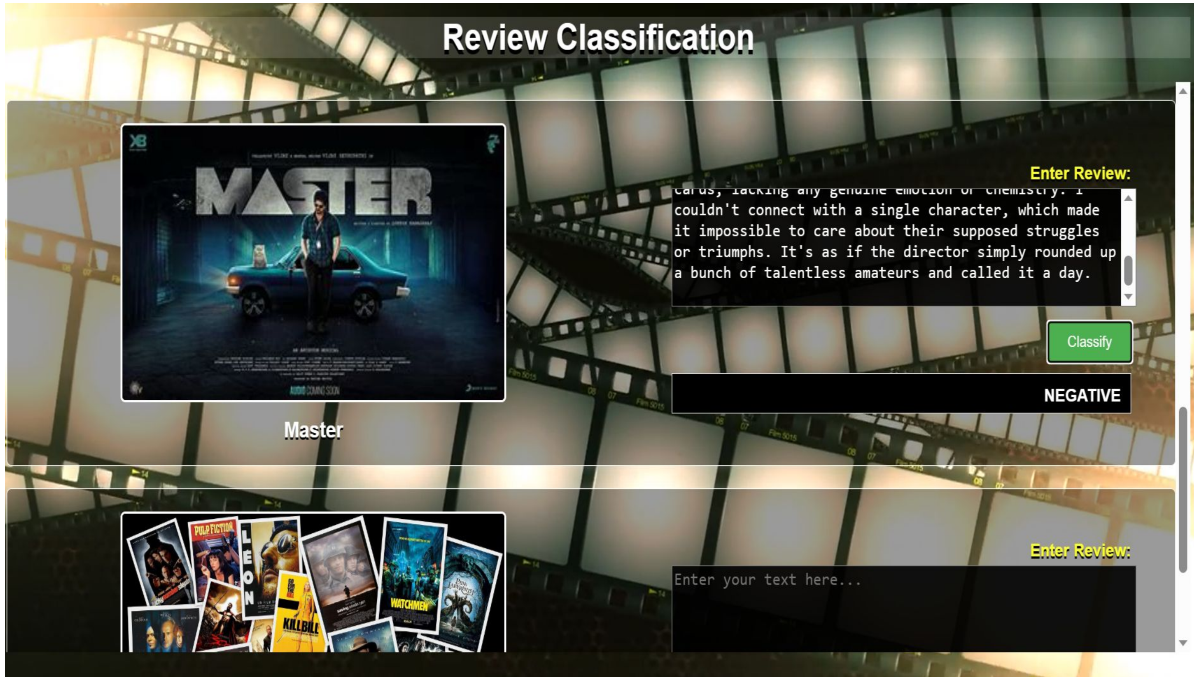
We have used logistic regression as classifier and tf-idf as vectorizer to train the model as it has shown the best accuracy amoung other combinations applied.

Front end clips:



"Whispered Secrets" is a chilling and suspenseful psychological thriller that will keep you on the edge of your seat from start to finish. Directed by an up-and-coming filmmaker, this indie gem takes viewers on a mind-bending journey filled with intrigue and

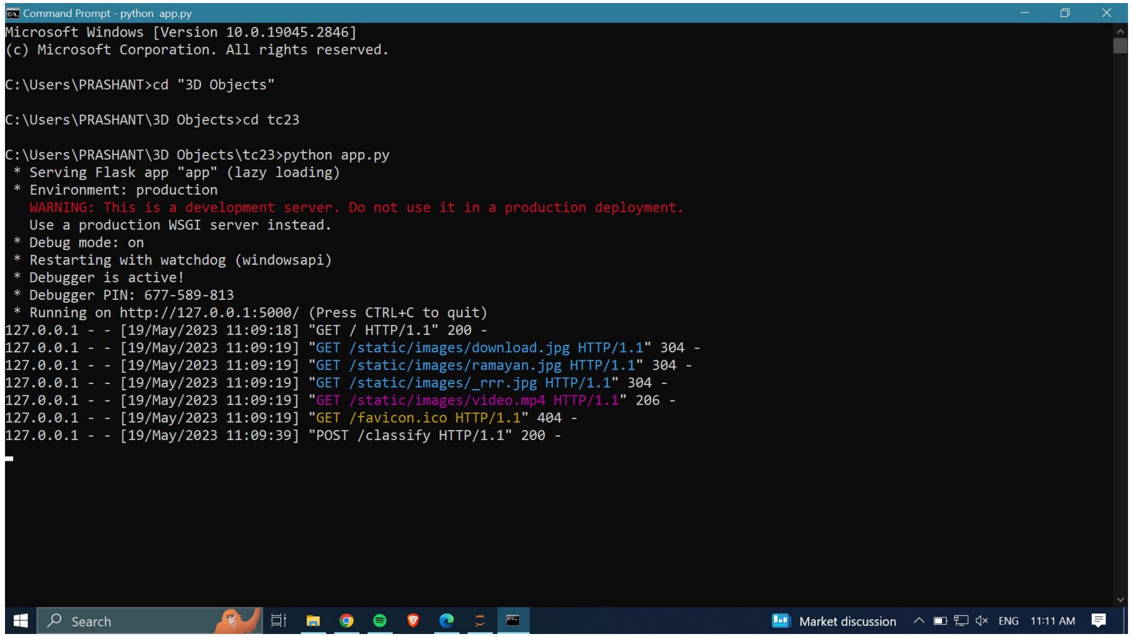
POSITIVE



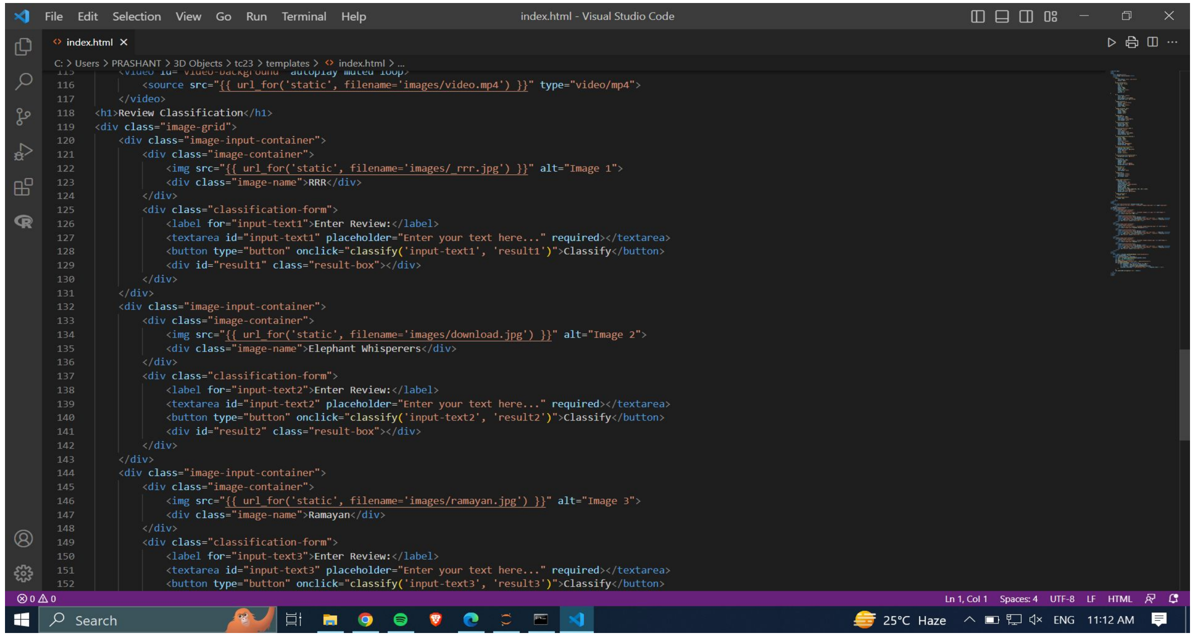
cards, lacking any genuine emotion or chemistry. I couldn't connect with a single character, which made it impossible to care about their supposed struggles or triumphs. It's as if the director simply rounded up a bunch of talentless amateurs and called it a day.

NEGATIVE

Back end design clips :

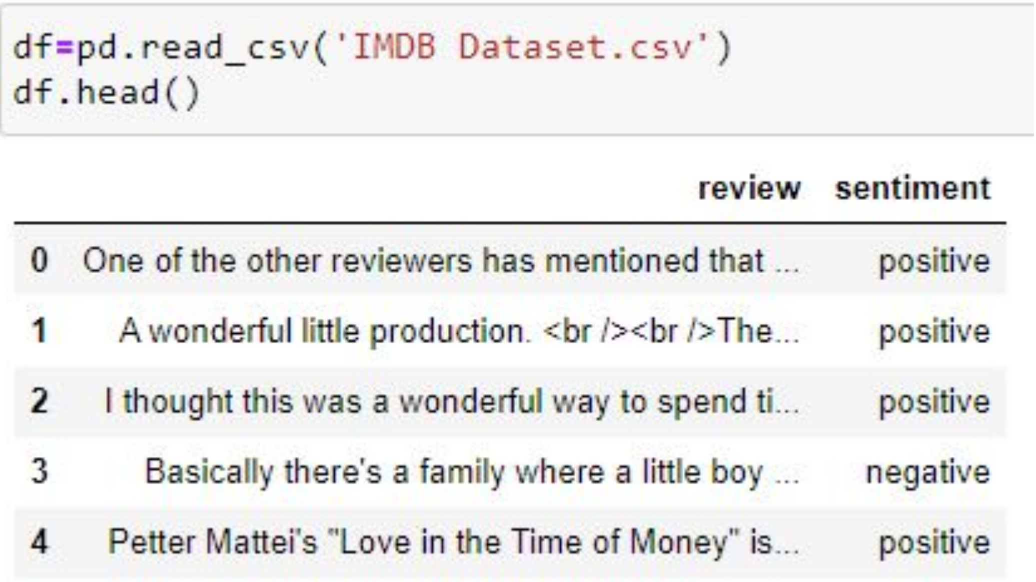


```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PRASHANT>cd "3D Objects"

C:\Users\PRASHANT\3D Objects>cd tc23

C:\Users\PRASHANT\3D Objects\tc23>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 677-589-813
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/May/2023 11:09:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/download.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/ramayan.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/_rrr.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/video.mp4 HTTP/1.1" 206 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/May/2023 11:09:39] "POST /classify HTTP/1.1" 200 -
```



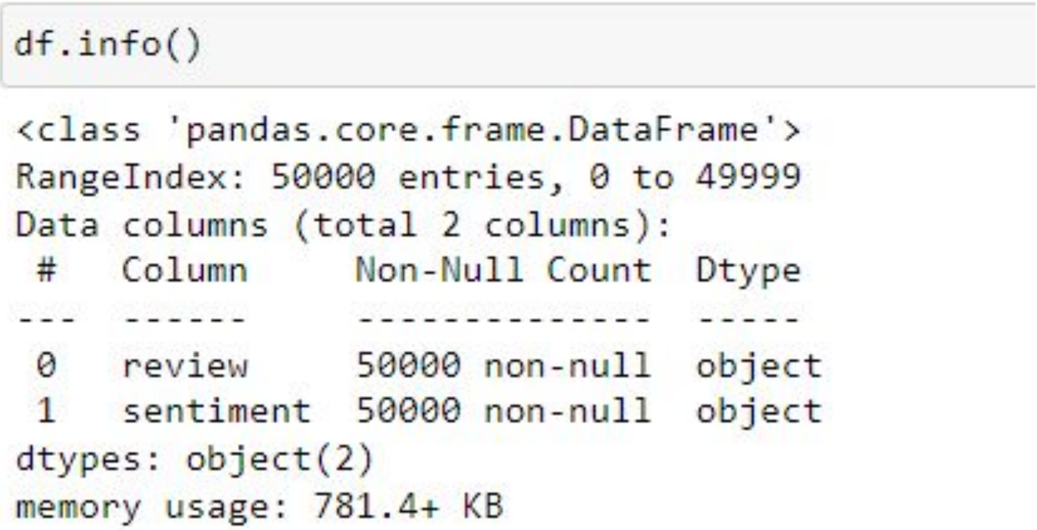
```
    <video id="video-background" autoplay muted loop>
        <source src="{{ url_for('static', filename='images/video.mp4') }}" type="video/mp4">
    </video>
<h1>Review Classification</h1>
<div class="image-grid">
    <div class="image-input-container">
        <div class="image-container">
            <img src="{{ url_for('static', filename='images/_rrr.jpg') }}" alt="Image 1">
            <div class="image-name">RRR</div>
        </div>
        <div class="classification-form">
            <label for="input-text1">Enter Review:</label>
            <textarea id="input-text1" placeholder="Enter your text here..." required></textarea>
            <button type="button" onclick="classify('input-text1', 'result1')">Classify</button>
            <div id="result1" class="result-box"></div>
        </div>
    </div>
    <div class="image-input-container">
        <div class="image-container">
            <img src="{{ url_for('static', filename='images/download.jpg') }}" alt="Image 2">
            <div class="image-name">Elephant Whisperers</div>
        </div>
        <div class="classification-form">
            <label for="input-text2">Enter Review:</label>
            <textarea id="input-text2" placeholder="Enter your text here..." required></textarea>
            <button type="button" onclick="classify('input-text2', 'result2')">Classify</button>
            <div id="result2" class="result-box"></div>
        </div>
    </div>
    <div class="image-input-container">
        <div class="image-container">
            <img src="{{ url_for('static', filename='images/ramayan.jpg') }}" alt="Image 3">
            <div class="image-name">Ramayan</div>
        </div>
        <div class="classification-form">
            <label for="input-text3">Enter Review:</label>
            <textarea id="input-text3" placeholder="Enter your text here..." required></textarea>
            <button type="button" onclick="classify('input-text3', 'result3')">Classify</button>
```

Data base :



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

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />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



```
df.info()
```

```
<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
```

CONCLUSION

- **Movie sentiment analysis** is a powerful tool for understanding and analyzing the sentiments expressed in movie reviews. Through sentiment analysis, we can gain valuable insights into audience opinions and perceptions, which have various applications in the film industry. Here are key conclusions regarding movie sentiment analysis:
- Audience Sentiment Understanding.
- Marketing Insights.
- Recommendation Systems.
- Content Evaluation and Improvement.
- Market Research and Competitive.

REFERENCES

- https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
- https://developers.google.com/machine-learning/guides/text-classification
- https://towardsdatascience.com/binary-classification-of-imdb-movie-reviews-648342bc70dd