

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**Jnana Sangama, Belagavi-590018**



**AI AND ML APPLICATION DEVELOPMENT MINI PROJECT**  
**REPORT ON**  
**“ SENTIMENT ANALYSIS FOR MOVIE REVIVES ”**

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

**BACHELOR OF ENGINEERING**  
**In**  
**Department of Artificial Intelligence and Machine Learning**

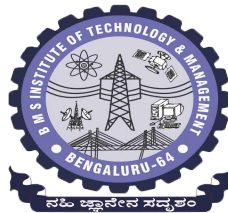
**Submitted By**

**Mr.Nagaabhishekha R [1BY19AI035]**  
**Ms.Pratiksha Rao [1BY19AI039]**  
**Ms.Puja S [1BY19AI040]**

**Under The Guidance Of**

**Dr. Anupama H S**  
**Associate Professor**  
**Dept. of AI&ML**

**Dr. Pradeep K R**  
**Assistant Professor**  
**Dept. of AI&ML**



**Department of Artificial Intelligence and Machine Learning**

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(Autonomous Institute Affiliated to VTU)**

**Avalahalli, Yelahanka, Bengaluru – 560064.**

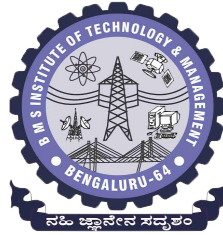
**2022-2023**

---

**VISVESWARAYA TECHNOLOGICAL UNIVERSITY**  
**Jnana Sangama, Belagavi-590018**

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**  
**(Autonomous Institute Affiliated to VTU)**

**Department of Artificial Intelligence and Machine Learning**  
**Avalahalli, Yelahanka, Bengaluru – 560064.**



**CERTIFICATE**

This is to certify that the **AI AND ML APPLICATION DEVELOPMENT MINI PROJECT** entitled “**Semantic Analysis For Movie Revives**” has been carried out by **Mr. Nagaabhishekha R (1BY19AI035), Ms. Pratiksha Rao (1BY19AI039) and Ms. Puja S (1BY19AI040)** a bonafide student of BMS Institute of Technology and Management, Autonomous Institute Affiliated to VTU, in partial fulfillment of the award of **Bachelor of Engineering** degree in **Department of Artificial Intelligence and Machine Learning** in **Visvesvaraya Technological University, Belagavi**, during the academic year 2022-2023. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of **AI And ML Application Development Laboratory with Mini Project (18AIL76)** work prescribed for the said degree.

-----  
**Dr. Anupama H S**  
Associate Professor,  
AI&ML, BMSIT&M

-----  
**Dr. Pradeep K R**  
Assistant Professor  
AI&ML, BMSIT&M

-----  
**Dr. Bharathi Malakreddy A**  
Prof. & Head,  
AI&ML, BMSIT&M

**Name of the Student**

**USN:**

**Signature of the Student:**

---

Signature of the Internal Examiner: \_\_\_\_\_

Signature of the External Examiner: \_\_\_\_\_

## DECLARATION

I, **Mr.Nagaabhishekha R (1BY19AI035), Ms.Pratiksha Rao (1BY19AI039), Ms. Puja S (1BY19AI040)**, student of VII Semester BE, in **Artificial Intelligence and Machine Learning**, BMS Institute of Technology and Management, Autonomous Institute Affiliated to VTU, hereby declare that the **AI And ML Application Development Laboratory with Mini Project (18AIL76)** entitled “**Semantic Analysis For Movie Revives**” has been carried out by us and submitted in partial fulfillment of the requirements for the *Bachelor of Engineering in the Department of Artificial Intelligence and Machine Learning* during the academic year 2022-23.

Date:

**Student Name:**

Place: Yelahanka Bangalore

USN:\_\_\_\_\_

## ACKNOWLEDGMENT

We are happy to present this **AI And ML Application Development Laboratory Mini Project (18AIL76)** report after completing it successfully. This Mini Project would not have been possible without the guidance, assistance, and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us make this a success.

We heartily thank our Principal, **Dr. MOHAN BABU. G.N**, BMS Institute of Technology & Management, Autonomous Institute Affiliated to VTU for his constant encouragement and inspiration in taking up this AI And ML Application Development Laboratory Mini Project work.

We heartily thank our Professor & Head of the Department, **Dr. Bharathi Malakreddy A**, Department of Artificial Intelligence and Machine Learning, BMS Institute of Technology & Management, Autonomous Institute Affiliated to VTU, for her constant encouragement and inspiration in taking up this AI And ML Application Development Laboratory Mini Project work.

We gratefully thank our AI And ML Application Development Laboratory Mini Project Guide, **Dr. Anupama H S**, Associate Professor, and **Dr. Pradeep K R**, Assistant Professor, Dept. of AI&ML for his/her guidance, support and advice.

Special thanks to all the staff members of the Artificial Intelligence and Machine Learning department for their help and kind cooperation. Lastly, We thank our parents and

---

friends for the support and encouragement given to us in completing this AI And ML Application Development Laboratory Mini Project work successfully

**Student name:** \_\_\_\_\_

**USN:** \_\_\_\_\_

## II

### **ABSTRACT**

Sentiment Analysis is a new subject in Research and is useful in many other fields. In Modern World, A huge amount of textual data is collected using surveys, comments, and reviews over the web. All of the collected data is used to improve products and services provided by both private organizations and governments around the world. This project includes sentiment analysis of movie reviews using feature-based opinion mining and supervised machine learning. Here, the main focus is to determine the polarity of reviews using nouns, verbs, and adjectives as opinion words. Reviews will be Classified into two different categories positive and negative. Reviews of Open Movie Database is used as source data set and Natural Language Processing Toolkit for Part of Speech Tagging.

Movie reviews are an important way to scale the performance of a movie. While providing stars/numerical ratings 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 into different aspects of the movie. A textual movie review tells us about the strong and weak points of the movie and a deeper analysis of a movie review can tell us if the movie, in general, meets the expectations of the reviewer.

Sentiment Analysis is a major subject in machine learning which aims to extract subjective information from textual reviews. The field of the sentiment of analysis is closely tied to natural language processing and text mining. A fundamental task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect. It focuses on whether the expressed opinion in a document, a sentence, or

---

a feature/aspect is positive, negative, or neutral. Sometimes it goes beyond polarity and looks at emotional states such as "angry," "sad," and "happy."

---

## TABLE OF CONTENTS

<b>ACKNOWLEDGMENT</b>	<b>I</b>
<b>ABSTRACT</b>	<b>II</b>
<b>SL NO.</b>	<b>CHAPTER NAME</b>
<b>1. Introduction</b>	<b>9</b>
1.1 Overview of Project	9
1.2 Problem Statement	10
<b>2. Requirement Analysis</b>	<b>11</b>
2.1 Software Requirement	11
2.2 Hardware Requirement	12
<b>3. System Design</b>	<b>13</b>
3.1 Initialization	13
3.2 Flowchart	20
<b>4. Implementation</b>	<b>21</b>
4.1 Modules Present	21
<b>5. Snapshots</b>	<b>22</b>
5.1 Snapshots	22
<b>6. Snippets</b>	<b>24</b>
6.1 Loading of data	24
6.2 Model building	24
6.3 Model testing	25
6.4 Deployment	25
<b>CONCLUSION</b>	<b>26</b>
<b>FUTURE ENHANCEMENTS</b>	<b>27</b>
<b>REFERENCES</b>	<b>28</b>

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>FIGURE NAME</b>	<b>PAGE NO.</b>
Figure 3.3.1	Distribution of sentiment	14
Figure 3.2	Proposed methodology workflow	20
Figure 5.1.1	Input for positive review	22
Figure 5.1.2	Output for negative review	22
Figure 5.1.3	Input for negative review	23
Figure 5.1.4	Output for a negative review	23
Figure 6.1	Code snippet for data loading	24
Figure 6.2	Code Snippet for the building of mode	24
Figure 6.3	Code Snippet for testing the model	25
Figure 6.4	Code snippet for model deployment	25



## LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
Table 3.1	Dataset sample	13
Table 3.9.1	Classification report	19

---

# CHAPTER 1

## INTRODUCTION

### 1.1 OVERVIEW OF THE PROJECT

Every Human Being makes decisions based on past experiences, sentiments, or opinions passed by other human beings. Whenever an individual wants to buy a new item or product, they seek ideas from others about the item or product. Similarly, every organization wants to deliver its best product to the market so they gather customer opinions about their product using surveys. Sentiment Analysis is a study of someone's opinions, sentiments, or emotions expressed about a product or a movie.

Sentiment analysis is the process of using natural language processing and computational linguistics to identify and extract subjective information from text. In the context of movie reviews, sentiment analysis can be used to automatically determine the overall sentiment of a review as positive, negative, or neutral. This can be useful for aggregating and summarizing the collective sentiment of a group of reviews to get a general idea of how people feel about a movie. Sentiment analysis can also be used to identify specific aspects of a movie that people like or dislike, such as the acting, the storyline, or the special effects. This information can be valuable for movie studios and filmmakers as they consider how to improve their films and market them to audiences.

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.

---

## **1.2 PROBLEM STATEMENT**

Each review given as input to the model predicts a rating ranging from 0-10. For a binary classification of sentiment (positive/negative) we have considered any review below 5 as unfavorable (assigned as 0) and any review 5 and above as positive (assigned as 1).

---

## CHAPTER 2

### REQUIREMENT ANALYSIS

#### 2.1 SOFTWARE REQUIREMENTS

- ❖ Operating system : Windows 7 or More
- ❖ Programming Language : Backend-Python,  
Frontend-HTML,CSS
- ❖ Dataset : Kaggle
- ❖ Software : Google Colab

---

## 2.2 HARDWARE REQUIRMENTS

- ❖ RAM :Minimum 4GB
- ❖ Processor :Any Intel or AMD x86-64 processor
- ❖ Internet Connection : Required

---

## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 Dataset -

The dataset containing IMDB movie reviews was obtained from Kaggle. Movies reviews data set contains a total of 50,000 movie reviews with labels for Natural Language Processing, in this data 25000 reviews are positive and 25000 reviews are negative.

REVIEW	SENTIMENT
Loved today's show!!! It was a variety and not solely cooking (which would have been great too). Very stimulating and captivating, always keeping the viewer peeking around the corner to see what was coming up next. She is as down to earth and as personable as you get, like one of us which made the show all the more enjoyable. Special guests, who are friends as well made for a nice surprise too. Loved the 'first' theme and that the audience was invited to play along too. I must admit I was shocked to see her come in under her time limits on a few things, but she did it and by golly I'll be writing those recipes down. Saving time in the kitchen means more time with family. Those who haven't tuned in yet, find out what channel and the time, I assure you that you won't be disappointed.	positive
Nickelodeon has gone down the toilet. They have kids saying things like "Oh my God!" and "We're screwed". This show promotes hate for people who aren't good looking, or aren't in the in crowd. It say that sexual promiscuity is alright, by having girls slobbering over shirtless boys. Not to mention the overweight boy who takes off his shirt. The main characters basically shun anyone out of the ordinary. Carly's friend Sam, who may be a lesbian, beats the snot out of anybody that crosses her path, This show has so much negativity in it that nobody should watch it! I give it a 0 out of 10!!!	negative

Table 3.1 - Dataset sample

---

### 3.2 Exploratory data analysis (EDA) -

The `info()` method was called on the dataset to obtain a summary of the attributes in the dataset. It was found that there are 2 columns named `review` with 50000 values and `sentiment` with 50000 values. There were no missing values.

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

### 3.3 Distribution of sentiment -

The below plot shows the distribution of sentiment (positive and negative) over the entire dataset by taking the count of the number of times each of the sentiment has occurred. It was found that there are an equal number of both positive and negative reviews that is 25,000 for each.

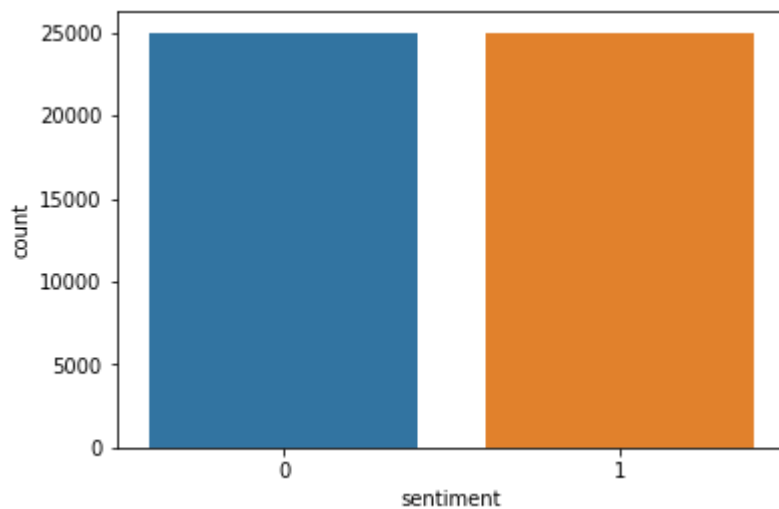


Figure 3.3.1 : Distribution of sentiment

### 3.4 Label Encoder -

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning. Label encoding converts the data in machine-readable form, but it assigns a unique number(starting from 0) to each class of data.

After applying label encoding, all negative reviews were given a label of 0 whereas all positive reviews were given a label of 1.

### 3.5 Natural language processing -

Natural language processing is the technique with which computers are able to understand human language as written or spoken. It is a subfield of Artificial Intelligence and linguistics. Sentiment analysis is one of the major applications of natural language processing. The following are some methods in NLP:

Tokenization - Is the process of segmenting running text into sentences and words. In essence, it's the task of cutting a text into pieces called tokens, and at the same time



---

throwing away certain characters, such as punctuation. Splitting on blank spaces may break up what should be considered as one token, as in the case of certain names.

Stopwords removal - Involves getting rid of common language articles, pronouns and prepositions such as “and”, “the” or “to” in English. In this process some very common words that appear to provide little or no value to the NLP objective are filtered and excluded from the text to be processed, hence removing widespread and frequent terms that are not informative about the corresponding text. Stop words can be safely ignored by carrying out a lookup in a predefined list of keywords, freeing up database space and improving processing time. We have used the predefined stopwords for english present in the nltk library.

Stemming - Refers to the process of slicing the end or the beginning of words with the intention of removing affixes (lexical additions to the root of the word).

Lemmatization - Lemmatization resolves words to their dictionary form (known as lemma) for which it requires detailed dictionaries in which the algorithm can look into and link words to their corresponding lemmas. It has the objective of reducing a word to its base form and grouping together different forms of the same word. For example, verbs in past tense are changed into present (e.g. “went” is changed to “go”) and synonyms are unified (e.g. “best” is changed to “good”), hence standardizing words with similar meaning to their root. Although it seems closely related to the stemming process, lemmatization uses a different approach to reach the root forms of words.

### 3.6 Feature Extraction -

Machine Learning algorithms learn from a predefined set of features from the training data to produce output for the test data. But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to convert text into a matrix(or vector) of features. The technique we have opted for is TFIDF vectorizer.

---

TF-IDF Vectorizer : TF-IDF stands for term frequency-inverse document frequency. It highlights a specific issue which might not be too frequent in our corpus but holds great importance. The TF-IDF value increases proportionally to the number of times a word appears in the document and decreases with the number of documents in the corpus that contain the word. It is composed of 2 sub-parts, which are :

Term Frequency(TF) : Term frequency specifies how frequently a term appears in the entire document. It can be thought of as the probability of finding a word within the document.

Inverse Document Frequency(IDF) : The inverse document frequency is a measure of whether a term is rare or frequent across the documents in the entire corpus. It highlights those words which occur in very few documents across the corpus, or in simple language, the words that are rare have high IDF score.

---

### 3.7 Data splitting -

The generated data must be appropriately split into training and validation/test sets to ensure that the models being trained are accurate. The splitting of data can be done mainly in 3 ways -

3.7.1 Random Sampling - Instances in the data are chosen randomly and added to the training or test sets. It protects the model from getting biased towards any aspect of the data, however it faces problems when the data is unevenly distributed

3.7.2 Stratified Random Sampling - This method also selects data instances at random but the selection is done within specific parameters. It ensures all the stratas or classes in the data are represented properly in the training and testing data

3.7.3 Nonrandom Sampling - This method is used when there is a different criteria to satisfy while splitting the data, like giving more weightage to more recent data

The data is split in the ratio of 8:2. 80% is used for training the model while the other 20% is used for testing and evaluating performance. This is done using the in-built `train_test_split` function in sklearn library.

### 3.8 Building Naive Bayes Model -

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.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

### 3.9 Performance evaluation -

Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model. To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics. These performance metrics help us understand how well our model has performed for the given data.

For classification problems the following metrics can be used -

3.9.1 Accuracy - The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

3.9.2 Confusion matrix - A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.

3.9.3 Precision - The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive or predictions that are actually true to the total positive predictions.

3.9.4 Recall or sensitivity - Recall aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive or predictions that are

---

actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

Accuracy - 0.8518

Confusion matrix -  $\begin{bmatrix} 4215 & 744 \\ 738 & 4303 \end{bmatrix}$

Classification report - (precision, recall, F1 score)

	precision	recall	F1 score	support
0	0.85	0.85	0.85	4959
1	0.85	0.85	0.85	5041
micro avg	0.85	0.85	0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	1000

Table 3.9.1 : Classification report

### 3.10 Saving model (Pickling) -

The trained model must be saved into drive so it can be called when required by the frontend. To do so we have used the pickle module in python. After you create a machine learning model, you train it. The training process usually takes some time. After the model is trained, what you get is the model object. To use the model, you can either

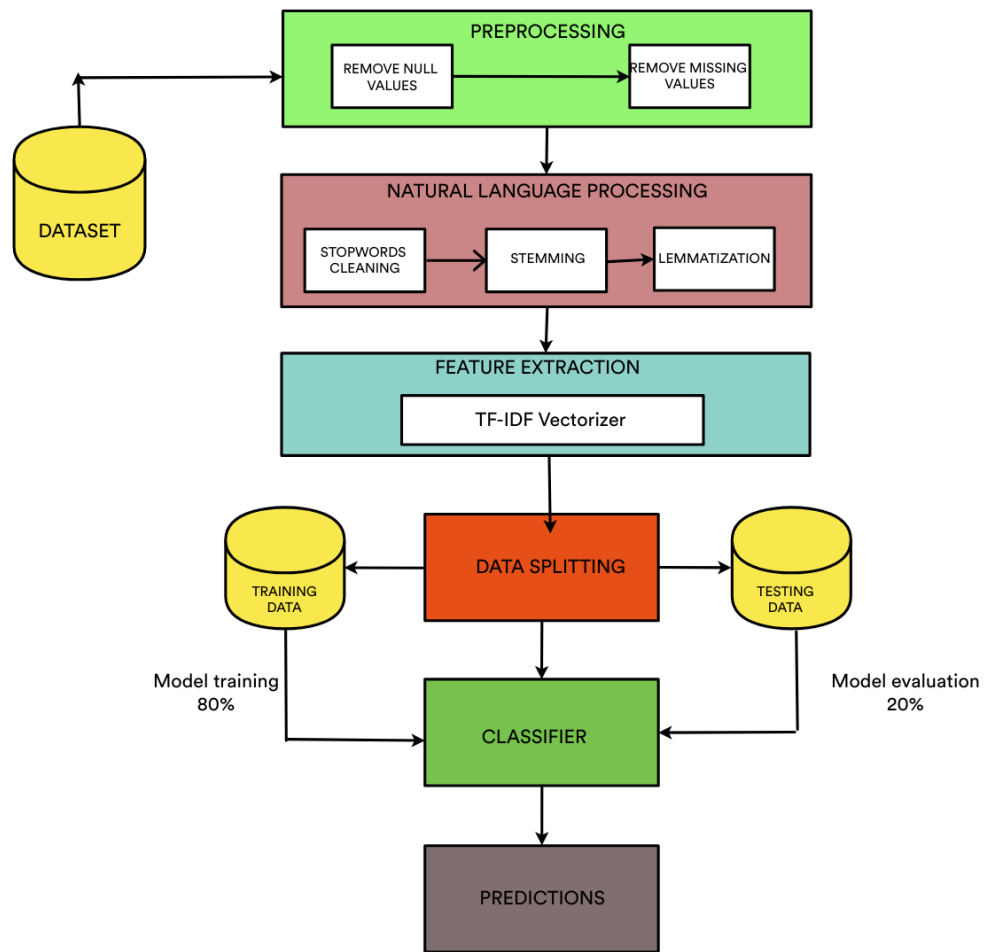
---

somehow save that model or you can retrain the model every time you use it, which is inefficient.

Pickle is a module in Python used for serializing and de-serializing Python objects. This converts Python objects like lists, dictionaries, etc. into byte streams (zeroes and ones). You can convert the byte streams back into Python objects through a process called unpickling. Pickling is also known as serialization, flattening, or marshalling.

### 3.11 Flask -

Flask is a web application framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[2] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools



**Figure 3.2 : Proposed methodology workflow**

---

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 Modules Present

- Sklearn
- nltk
- pandas
- numpy
- pickel
- seaborn
- matplotlib
- flask
- gunicorn.



## CHAPTER 5

### SNAPSHOTS

#### 5.1 SCREENSHOTS OF INPUT AND OUTPUT

Figure 5.1.1 shows the screenshot of the input for predicting a review which can be a positive review.



Figure 5.1.1: Input for positive review

Figure 5.1.2 shows the screenshot of the output for predicting a review which can be a positive review

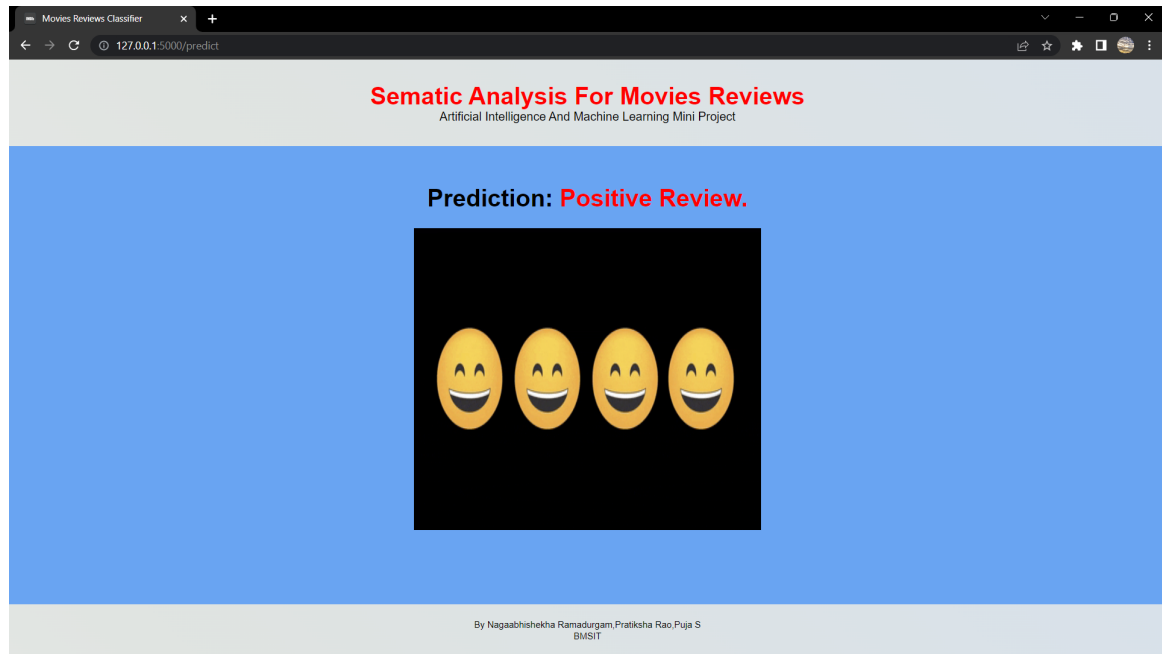


Figure 5.1.2: Output for positive review

Figure 5.1.3 shows the screenshot of the input for predicting a review which can be a negative review



Figure 5.1.3: Input for negative review

Figure 5.1.4 shows the screenshot of the output for predicting a review which can be a negative review



Figure 5.1.4: Output for negative review

# CHAPTER 6

## CODE SNIPPET

### 1. Loading of data

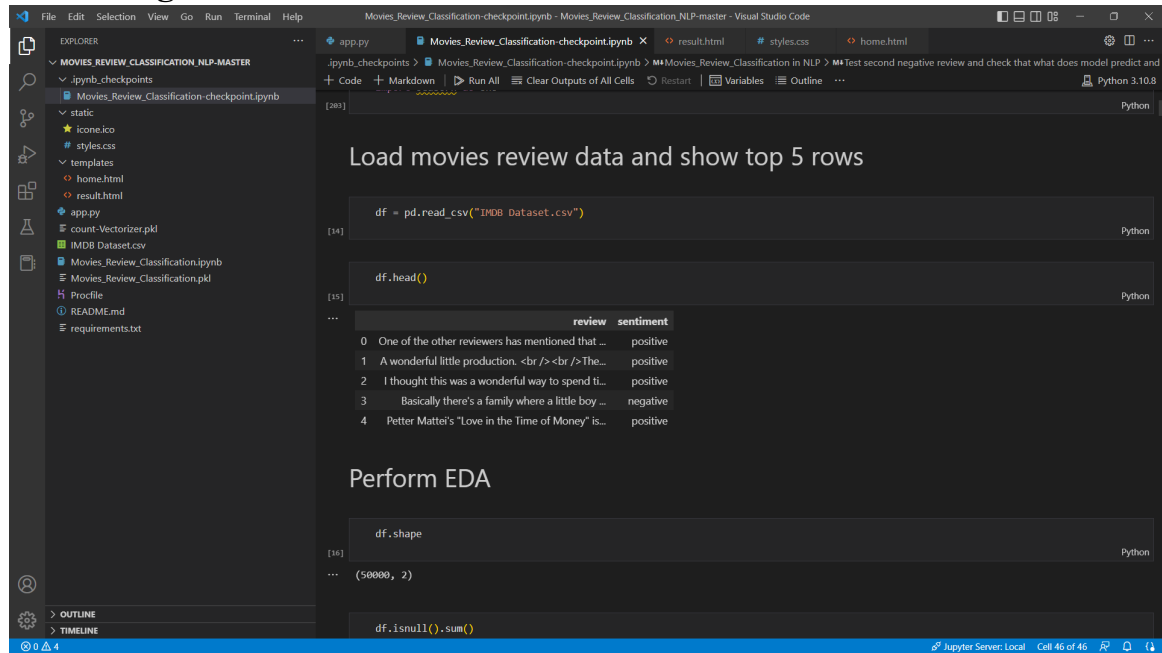
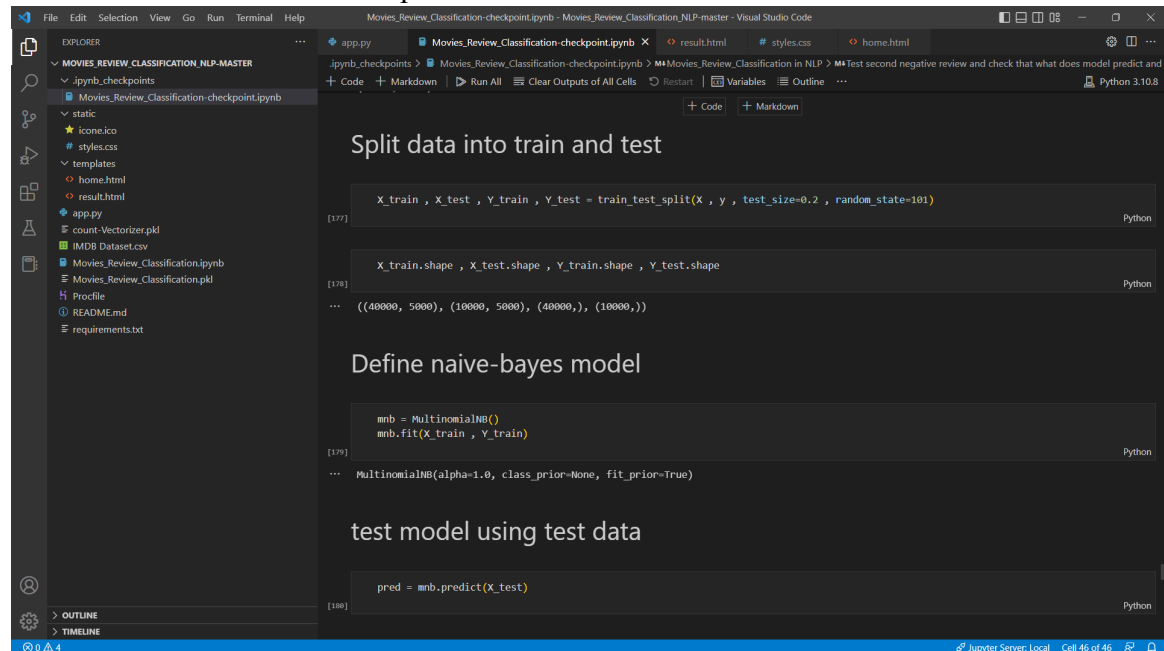


Figure.6.1: Code snippet for data loading

## 2. Model Building

Fig 5.2 shows a code snippet for model training and evaluation for all the defined classifiers as mentioned in the previous section.



```
EXPLORER
MOVIES_REVIEW_CLASSIFICATION_NLP-MASTER
  .ipynb_checkpoints
  Movies_Review_Classification-checkpoint.ipynb
  static
    icons.ico
    styles.css
  templates
    home.html
    result.html
  app.py
  count-Vectorizer.pkl
  IMDb8 Dataset.csv
  Movies_Review_Classification.ipynb
  Movies_Review_Classification.pkl
  Profile
  README.md
  requirements.txt

Split data into train and test

X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2, random_state=101)

X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
((40000, 5000), (10000, 5000), (40000,), (10000,))

Define naive-bayes model

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

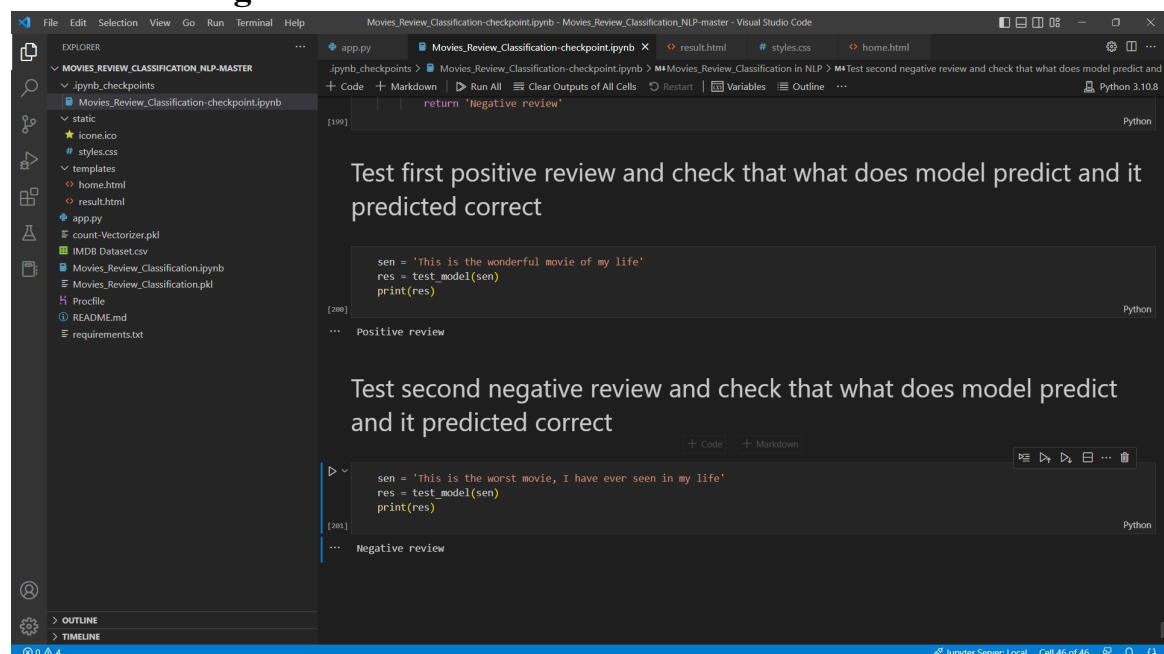
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

test model using test data

pred = mnb.predict(X_test)
```

Figure.6.2:Code Snippet for the building of model

## 3.Model testing



```
EXPLORER
MOVIES_REVIEW_CLASSIFICATION_NLP-MASTER
  .ipynb_checkpoints
  Movies_Review_Classification-checkpoint.ipynb
  static
    icons.ico
    styles.css
  templates
    home.html
    result.html
  app.py
  count-Vectorizer.pkl
  IMDb8 Dataset.csv
  Movies_Review_Classification.ipynb
  Movies_Review_Classification.pkl
  Profile
  README.md
  requirements.txt

return 'Negative review'

Test first positive review and check that what does model predict and it predicted correct

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

Positive review

Test second negative review and check that what does model predict and it predicted correct

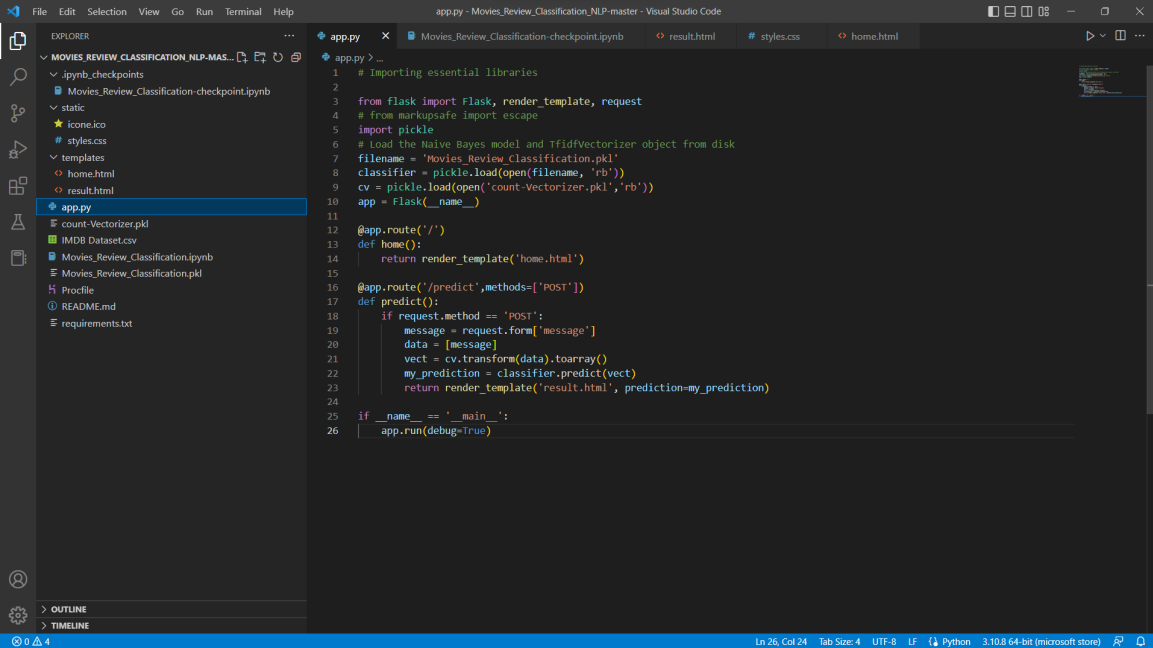
sen = 'This is the worst movie, I have ever seen in my life'
res = test_model(sen)
print(res)

Negative review
```

Figure.6.3:Code Snippet for testing model

## 4. Deployment

Fig 5.4 shows a code snippet for deployment of the best trained model as a web app using the Streamlit Python package.



```
1 # Importing essential libraries
2
3 from flask import Flask, render_template, request
4 # from marupsafe import escape
5 import pickle
6 # load the Naive Bayes model and TfidfVectorizer object from disk
7 filename = 'Movies_Review_Classification.pkl'
8 classifier = pickle.load(open(filename, 'rb'))
9 cv = pickle.load(open('count-Vectorizer.pkl', 'rb'))
10 app = Flask(__name__)
11
12 @app.route('/')
13 def home():
14     return render_template('home.html')
15
16 @app.route('/predict', methods=['POST'])
17 def predict():
18     if request.method == 'POST':
19         message = request.form['message']
20         data = [message]
21         vect = cv.transform(data).toarray()
22         my_prediction = classifier.predict(vect)
23         return render_template('result.html', prediction=my_prediction)
24
25 if __name__ == '__main__':
26     app.run(debug=True)
```

Figure.5.4:Code snippet for model deployment

---

## **CONCLUSION**

We have successfully developed a more accurate Semantic Analysis For Movie Reviews with neural networks in Python, with its accuracy being close to 87%. We have deployed the best running model using the Flask framework, which will enable the end user to easily input a given query message or email into a text box and consequently get the results Positive Review or Negative Review. The following results of a query can be obtained without any real-time delay.

---

## **FUTURE ENHANCEMENT**

For future enhancements, we can develop a model with still more accuracy in the classifier. Develop a more interactive, user-friendly interface (UI) for the end user so that they can see the analytics of why a review is good or bad in addition to receiving the classification tag of a message.

## **REFERENCES**

1. <https://acadpubl.eu/hub/2018-118-21/articles/21d/44.pdf>



- 
2. [https://www.ripublication.com/ijaer18/ijaerv13n16\\_53.pdf](https://www.ripublication.com/ijaer18/ijaerv13n16_53.pdf)
  3. [https://www.ripublication.com/ijaer18/ijaerv13n16\\_53.pdf](https://www.ripublication.com/ijaer18/ijaerv13n16_53.pdf)
  4. [Kia Dashtipour - Lecturer at Edinburgh Napier University - LinkedIn](#)
  5. [Opinion Mining and Information Fusion: A Survey - Academia.edu](#)
  6. <https://api.themoviedb.org/3/> - Movie Reviews Input API