AMAZON ECHO REVIEWS SENTIMENT ANALYSIS USING NLP

This project report is submitted in partial fulfilment of the requirement

for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

JILAGAM DHEERAJ

Reg.no: 208297601021

Under the esteemed guidance of

Dr B. Kezia Rani

Associate Professor & Head of the Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

2020-2024

ADIKAVI NANNAYA UNIVERSITY

RAJAMAHENDRAVARAM

UNIVERSITY COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this project report entitled, "**AMAZON ECHO REVIEWS SENTIMENT ANALYSIS USING NLP**" is a bonafide work of **JILAGAM DHEERAJ**, Reg.No: **208297601021** submitted in a partial fulfilment of the requirements for the award of Degree of BTech (CSE) during the period 2020-2024. This work carried out by him under my supervision and guidance and submitted to Department of Computer Science and Engineering, Adikavi Nannaya University.

INTERNAL GUIDE **HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINER

DECLARATION

I **JILAGAM DHEERAJ,** reg.no:**208297601021**, hereby declare that the project report entitled "Amazon Echo Reviews Sentiment Analysis Using NLP" done by me under the guidance of **Dr B.Kezia Rani, Associate Professor, Adikavi Nannaya University**, is submitted for the partial fulfilment of requirement of the award of the degree, Bachelor of Technology in Computer Science and Engineering in the academic year 2020-2024.

Signature of the student

ACKNOWLEDGEMENT

I feel fortunate to pursue my **Bachelor of Technologies** degree in the **Department of Computer Science and Engineering**, **University College of Engineering, Adikavi Nannaya University**. It provided all the facilities in the area of Computer Science and Engineering. It's a genuine pleasure to express my deep sense of thanks and gratitude to my mentor and project guide **Dr B.Kezia Rani, Associate Professor, Department of Computer Science and Engineering, Adikavi Nannaya University** for her excellent guidance right from the selection of the project and her valuable suggestions throughout the project work. Her constant and timely counsel has caused me to succeed in completing this project in college.

I profusely thank **Dr. B. Kezia Rani**, Head of the Department, of Computer Science and Engineering for the guidance.

I also thank **Dr. P. Venkateswara Rao**, Principal, University College of Engineering for all the encouragement and support.

I also thank **Dr. D. Latha**, Project Coordinator, Department, Computer Science and Engineering for the guidance.

A great deal of thanks goes to Review Committee Members and the entire Faculty of the Department of Computer Science and Engineering, University College of Engineering, Adikavi Nannaya University, Rajamahendravaram for their excellent supervision and valuable suggestions.

JILAGAM DHEERAJ
208297601021

ABSTRACT

The increasing volume of product reviews across online platforms presents a challenge for sellers to analyze customer sentiments. Examining thousands of customer reviews is time-consuming and tedious. This project proposes a sentiment analysis system utilizing machine learning (ML) and natural language processing (NLP) to address the challenge that benefits both the company and the customer. The system aims to extract valuable insights from large sets of product reviews, aiding sellers in enhancing product features and informing potential customers about product pros and cons. This project enhances the present scenario by offering NLP techniques to train ML models using XG Boost and Random Forest models, and advanced data visualizations providing insights to customers. Effective customer feedback analysis is crucial for any company, as it serves as a valuable resource for enhancing business opportunities and overall improvement.

Key-words: XG Boost, Random Forest, Sentiment Analysis, NLP

INDEX

LIST OF FIGURES

1. INTRODUCTION

The project includes several key components, such as Data Selection, Data Preprocessing, NLP techniques, Data Splitting, Feature Scaling, Classification, and Result Generation. NLP techniques are used to preprocess the text data. This involves tasks such as tokenization, lowercasing, removing stop words, stemming, and handling special characters. The preprocessed data is split into training and testing sets. Machine learning models, such as XGBoost and Random forest classifiers, are applied to the preprocessed and split data to perform sentiment analysis. The performance of the applied models is evaluated using metrics such as accuracy, precision, recall, and F1 score to assess their effectiveness in predicting sentiment.

The dataset is split into training and testing sets using the train_test_split function, where 70% of the data is used for training and 30% of the data is used for testing. Model Training as the sentiment analysis model is trained using different classifiers, such as XGBoost, Random Forest, and Decision Tree Classifier. Model Evaluation is the accuracy of the models on both the training and testing datasets to assess their performance. Saving Models and Transformers as **t**he trained models Random Forest, XGBoost, and preprocessing transformers CountVectorizer, and MinMaxScaler are saved using the pickle module. This allows the use of these models later without retraining.

The project also involves the development of a user interface, potentially using a web-based application built with Flask, to allow users to interact with the sentiment analysis system. Web Application Integration for user interaction, an interface using Flask is created, allowing users to upload a CSV file for bulk prediction or enter text for single predictions. The interface supports functionalities such as selecting the dataset, providing text input for analysis, and displaying the sentiment analysis results.

1.1 EXISTING SYSTEM

The existing system is a rule-based system that uses pre-defined rules to classify the sentiment of the reviews. This approach involves defining a set of rules based on keywords or phrases that indicate the sentiment. However, this approach may not be effective in capturing the nuances of language and may not be adaptable these days.

DRAWBACKS:

Nuance and Context issues

Low Scalability and High Maintenance

Limited Adaptability

1.2 PROPOSED SYSTEM

By adopting machine learning and NLP-based approaches, the project aimed to overcome the limitations of the rule-based approach, allowing for more adaptive, data-driven, and context-aware sentiment analysis of the Amazon Alexa reviews. The interface supports functionalities such as selecting the dataset, providing text input for analysis, and displaying the sentiment analysis results.

ADVANTAGES:

Better Nuance and Context

Efficient for larger data sets

Subjective

.

2. SYSTEM ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

1) **Prediction**: The sentiment analysis system must be able to utilize the trained model to make accurate predictions on new data, ensuring that it can consistently provide reliable and up-to-date results.

2) **Graph generation:** It is essential for the system to generate graphs and reports that summarize sentiment trends and patterns in a visually appealing and user-friendly way. This feature enables users to interpret the data quickly and make informed decisions based on the results.

3) **Data acquisition:** The system should be designed to allow users to interact with it and input their own sentences for sentiment analysis. This feature will enable the system to gather more data and improve its performance.

4) **Evaluation Metrics:** To ensure that the sentiment analysis model performs accurately, the system must use standard evaluation metrics such as precision, recall, and F1 score to evaluate its performance continually. This feature ensures that the system remains reliable and provides accurate results.

2.2 NON-FUNCTIONAL REQUIREMENTS

1) **Usability:** The user interface must be intuitive, with clear navigation and easy-to-understand features to ensure user adoption and satisfaction.

2) **Reliability:** The code should handle potential errors gracefully, providing informative error messages.

3) **Cross-Platform Compatibility:** The code should be portable across different operating systems like Windows, macOS, and Linux without significant modifications.

4) **Ease of Understanding:** Code and comments should be in a way that is easy to understand for both novice and experienced users.

2.3 SOFTWARE REQUIREMENTS

1)Programming Language: Python-3

2)Integrated Development Environment (IDE): Jupyter Notebook, Pycharm.

3)Libraries and Packages: pandas for data manipulation

wordcloud for creating word clouds

scikit-learn for machine learning tools

matplotlib and seaborn for data visualization

JupyterThemes for distinguishing between graphs

nltk for natural language processing tasks

XGBoost For implementing the XGBoost classifier for sentiment analysis.

2.4 HARDWARE REQUIREMENTS

1)**RAM**: 4GB

2)**Storage**: 32 GB

3)**Processor:** Intel i5

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

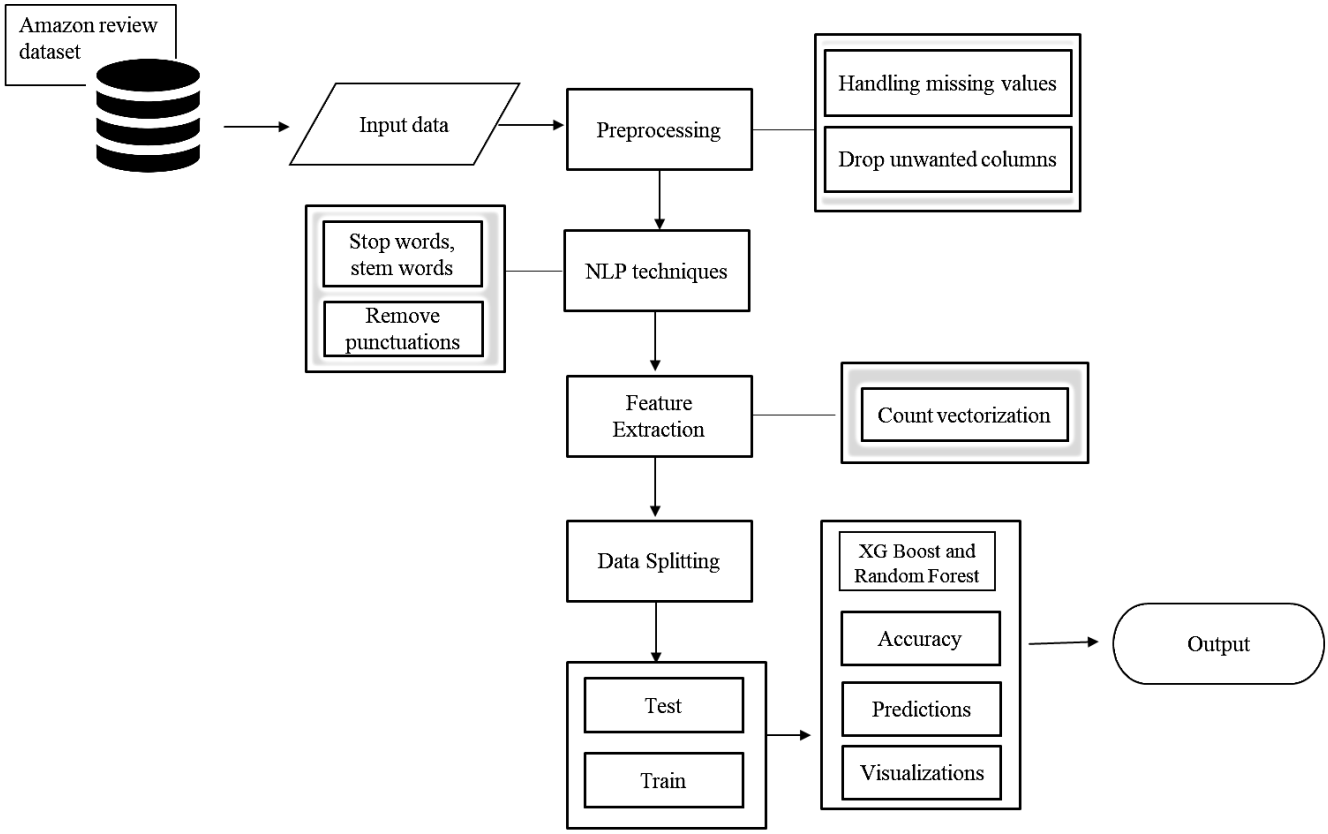

FIG 3.1 SYSTEM ARCHITECTURE

1. **Data Acquisition:** The process starts with acquiring data from an Amazon review dataset.
2. **Preprocessing:** The data is then preprocessed to prepare it for modeling. This includes:
    - Dropping unwanted columns: Irrelevant columns are removed from the dataset.
    - Text cleaning: This may involve removing punctuation, stop words, and stemming or lemmatization (converting words to their base form).
3. **Feature extraction:** Features are extracted from the text data. This could involve techniques like:

- Bag-of-words: This converts each document into a sparse vector where each element represents the number of times a particular word appears in the document.
- TF-IDF: This is a variant of bag-of-words that weights the importance of words based on their frequency in a document and rarity across the corpus.

4. **Model training:** The extracted features are then used to train a machine learning model to predict sentiment. The diagram shows two models being trained: XGBoost and Random Forest. These are both popular tree-based ensemble methods for classification.
5. **Testing and evaluation:** The evaluation process involves feeding the test set into the model and measuring its performance metrics, such as accuracy, precision, recall, and F1 score. The results of the evaluation will help determine whether the model is ready for deployment or if further improvements are needed.
6. **Model Predictions:** Once a machine learning model is trained and evaluated, it can be used to generate predictions on new and unseen data. This process involves applying the model to the new data, which could be in the form of text, images, or any other input format that the model was designed to handle. For example, if the model was trained on Amazon reviews and labeled them as positive or negative based on their sentiment, it can be used to predict the sentiment of new Amazon reviews. This way, the model can assist in automating tasks that would otherwise require human effort.
7. **Visualizations:** After conducting an in-depth analysis of the data, you can use various visualization techniques to represent the findings in a more meaningful way. These visualizations can provide a clear picture of the sentiment of the data and help you gain valuable insights into the patterns and trends that might have been missed otherwise. By using charts, graphs, and other visual tools, you can present the results in a more compelling and easy-to-understand manner, enabling you to make better-informed decisions based on the data analysis.

3.2 UML DIAGRAMS

3.2.1 CLASS DIAGRAM

The class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations or methods, and the relationships among objects.

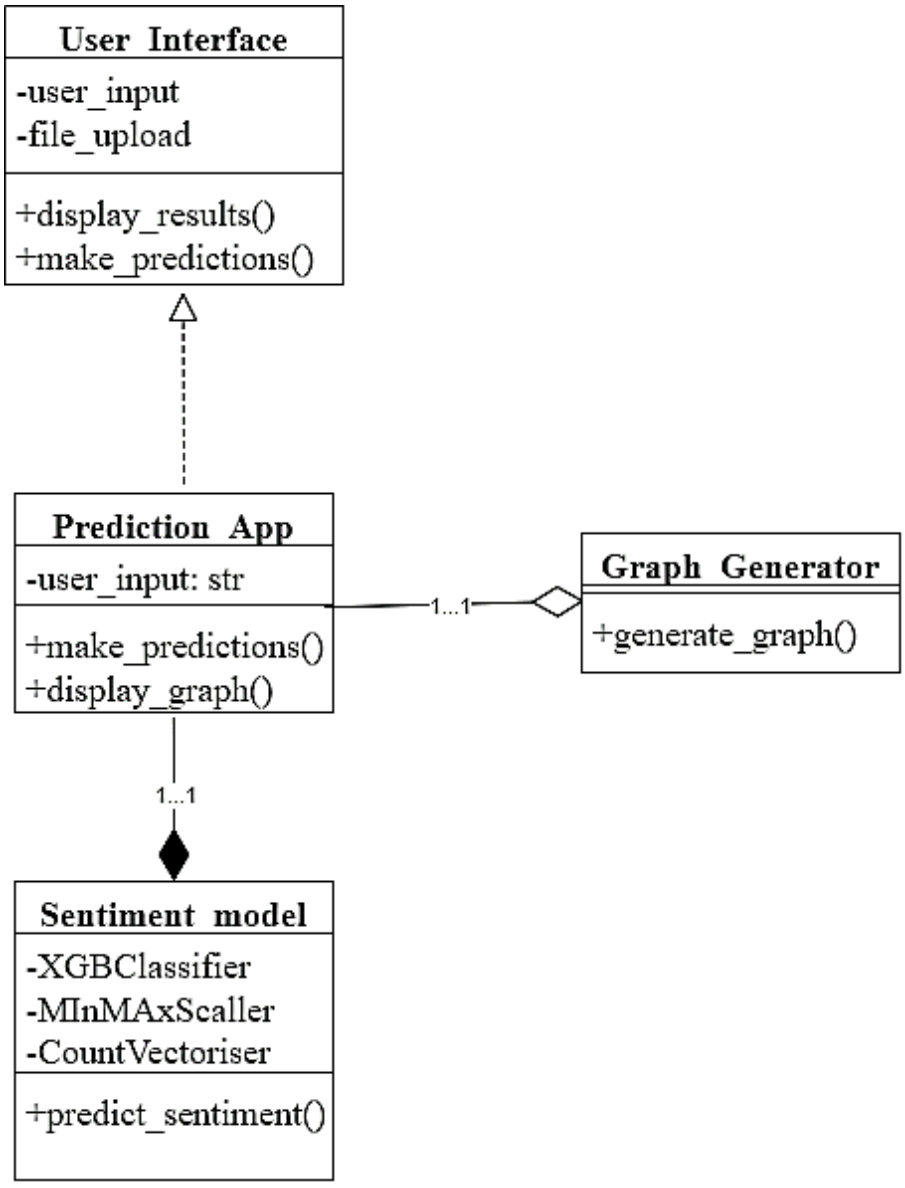

Fig:3.2. 1 CLASS DIAGRAM

1. **Input:** This class is responsible for handling user input, including selecting the data to be analyzed and providing text input for sentiment analysis. Select data(): This method allows the user to select the dataset to be analyzed for sentiment analysis. Provide_text(): This method allows the user to input text for sentiment analysis.

2. **Preprocessing:** This class is responsible for preprocessing the input data, including handling missing values and vectorizing the data for machine learning model application. Missing_values(): This method handles missing values in the input data, potentially using techniques such as imputation or deletion. Vectorization(): This method vectorizes the input data, converting it into a format suitable for machine learning model applications.

3. **Data Splitting:** This class is responsible for splitting the preprocessed data into training and testing sets, a crucial step in machine learning model development. Test(): This method splits the preprocessed data into a testing set, which is used to evaluate the performance of the machine learning model. Train(): This method splits the preprocessed data into a training set, which is used to train the machine learning model.

4. **Classification:** This class is responsible for applying machine learning models to the preprocessed and split data for sentiment analysis. XGBoost(): This method applies the XGBoost machine learning model to the preprocessed and split data for sentiment analysis. Random Forest(): This method applies the Random Forest machine learning model to the preprocessed and split data for sentiment analysis.

5. **Performance Analysis:** This class is responsible for evaluating the performance of the applied machine learning models and generating sentiment analysis predictions. Prediction(): This method generates sentiment analysis predictions based on the applied machine learning models, potentially using metrics such as accuracy, precision, recall, or F1 score to evaluate performance.

3.2.2 USE CASE DIAGRAM

A use case diagram is a visual representation of how a user interacts with a system. It shows the various use cases and user types for a system. Use case diagrams are a type of behavior diagram.

1. **User:** The primary actor in the system, interacting with the client interface to provide input for sentiment analysis. The user initiates the sentiment analysis process by providing input data and expects to receive the sentiment analysis results.

2. **System:** The system itself, comprises various components responsible for processing the user input, performing sentiment analysis, and presenting the results back to the user.

3. **Dataset:** The system acquires the dataset containing Amazon Alexa reviews, which serves as the basis for sentiment analysis. Preconditions: The dataset must be available for the sentiment analysis process to begin. Postconditions: The dataset is ready for further processing and analysis.

4. **Data Preprocessing:** The system preprocesses the input data, which involves tasks such as removing special characters, converting text to lowercase, and tokenizing the input into individual words. Preconditions: The dataset must be available, and the preprocessing components must be operational. Postconditions: The preprocessed data is ready for further analysis and model application.

5. **Split the Data:** The system splits the preprocessed data into training and testing sets, a crucial step in machine learning model development. Preconditions: The preprocessed data must be available and ready for splitting. Postconditions: The training and testing sets are prepared for model training and evaluation.

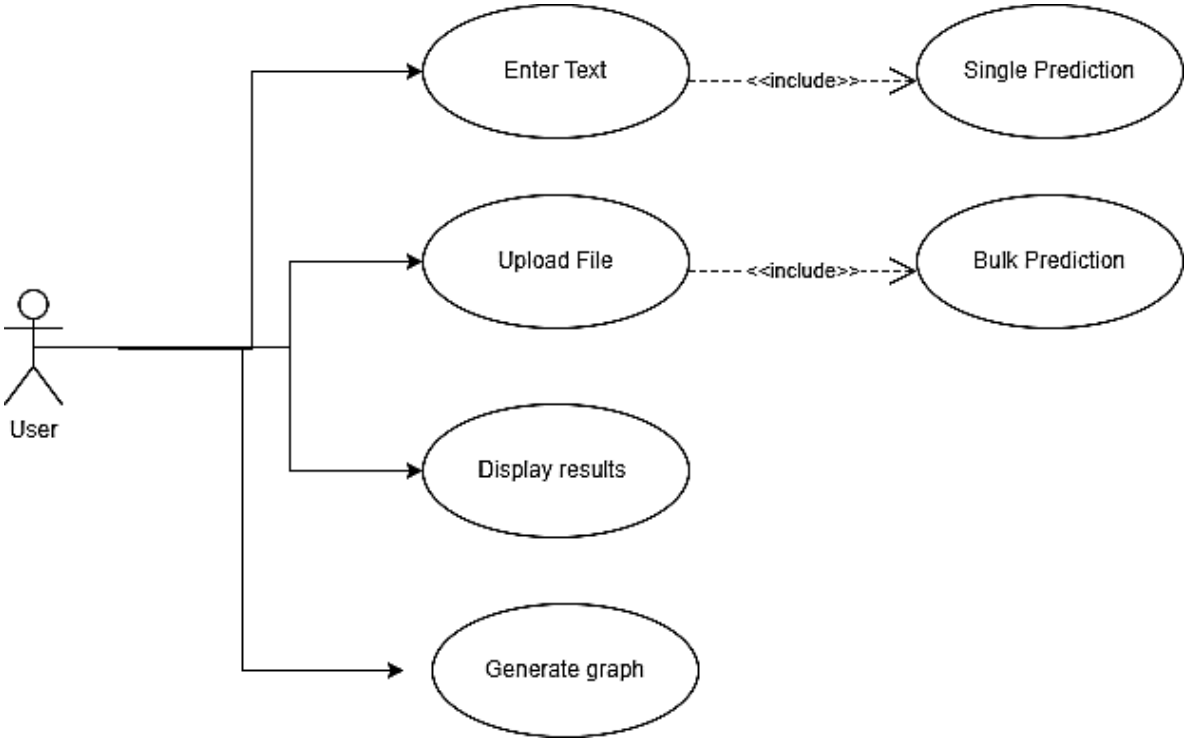

FIG 3.2.2 USE CASE DIAGRAM

6. **Apply Model:** The system applies the machine learning model like XGBoost or Random Forest classifier to the preprocessed and split data for sentiment analysis. Preconditions: The preprocessed and split data must be available, and the machine learning model must be trained and ready for application. Postconditions: The sentiment analysis results are generated based on the applied model.

7. **Evaluation**: The system evaluates the performance of the applied model, typically using metrics such as accuracy, precision, recall, or F1 score. Preconditions: The sentiment analysis results and ground truth labels must be available for evaluation. Postconditions: The model's performance metrics are determined, providing insights into its effectiveness.

8. **Display Results:** The system displays the sentiment analysis results to the user, potentially including visualizations such as word clouds or graphs to represent the sentiment distribution. Preconditions: The sentiment analysis results must be available and ready for presentation. Postconditions: The sentiment analysis results are presented to the user for review and interpretation.

3.2.3 SEQUENCE DIAGRAM

A sequence diagram 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.

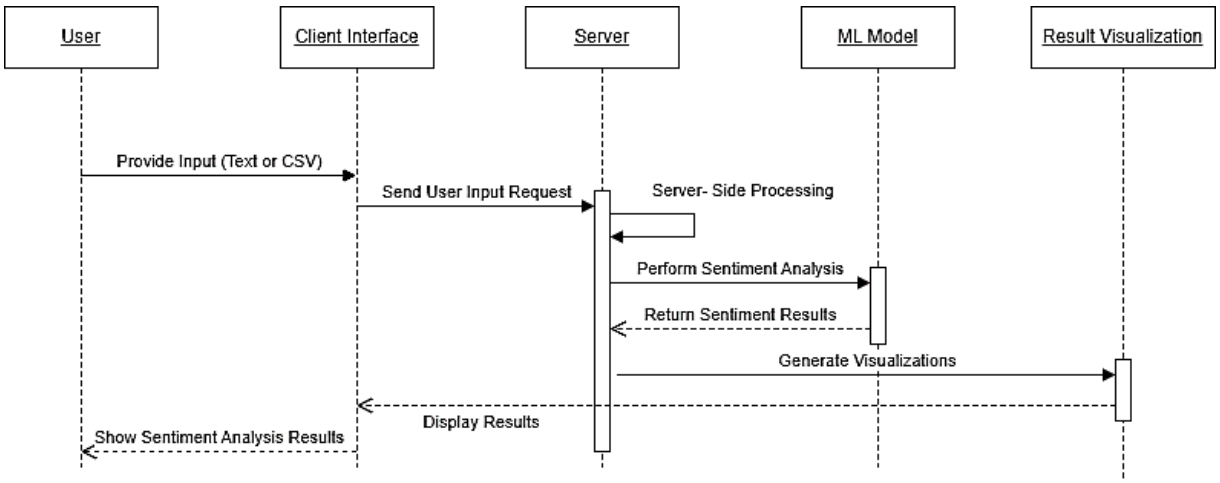

FIG 3.2.3 SEQUENCE DIAGRAM

1. **User Input Interaction:** The sequence diagram begins with the user interacting with the client interface, providing input in the form of text or uploading a CSV file for sentiment analysis.

2. **Client-Server Interaction:** Upon receiving the user input, the client interface sends a request to the server, either as a text input or a file upload request.

3. **Server-Side Processing:** The server receives the user input request and initiates the sentiment analysis process. This involves pre-processing the input data, transforming it using the CountVectorizer and MinMaxScaler, and passing it to the machine learning model for sentiment prediction.

4. **Machine Learning Model Interaction:** The server interacts with the machine learning model to perform sentiment analysis on the pre-processed input data. This interaction includes passing the transformed data to the model and receiving the predicted sentiment results.

5. **Result Visualization and Output:** After sentiment analysis, the server may generate visualizations and prepare the sentiment analysis results for presentation to the user.

6. **Client Response:** The server sends the sentiment analysis results back to the client interface, which then displays the results to the user, either as individual sentiment predictions or as a downloadable CSV file.

3.2.4 ACTIVITY DIAGRAM

An activity diagram shows the flow from one activity to another in a system or process. It's used to describe the different dynamic aspects of a system and is referred to as a 'behavior diagram' because it describes what should happen in the modeled system.

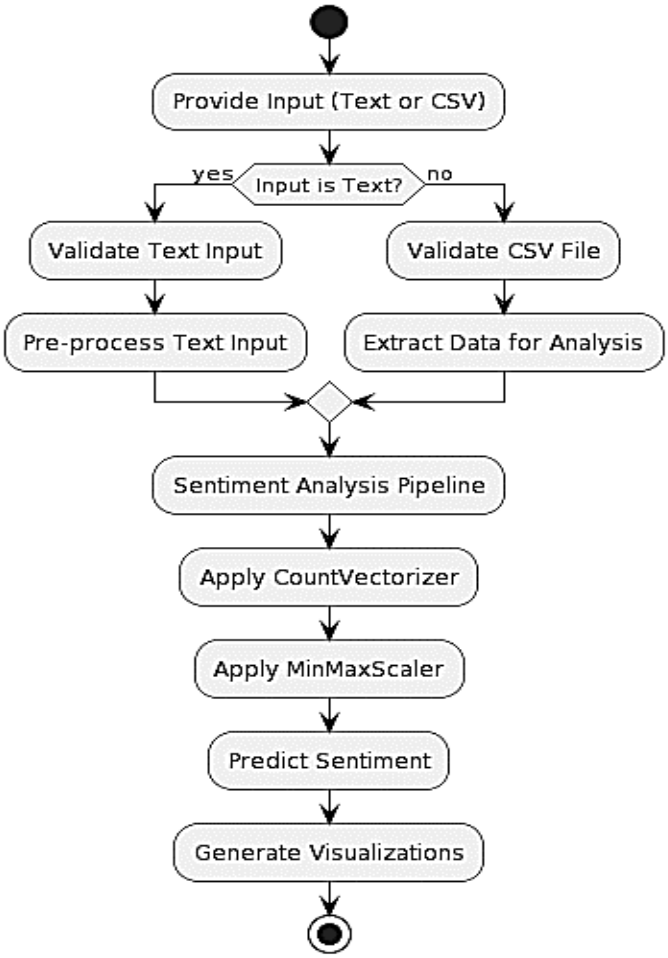

FIG:3.2. 4 ACTIVITY DIAGRAM

1. **User Input Processing:** The activity diagram can start with a user providing input to the system, either as a text input or by uploading a CSV file containing multiple sentences for sentiment analysis.

2. **Input Validation and Pre-processing:** Upon receiving the user input, the system validates the input to ensure it meets the required format and data integrity. For text input, the system may perform pre-processing steps such as removing special characters, converting text to lowercase, and tokenizing the input into individual words. - If a CSV file is uploaded, the system reads and validates the file, extracting the relevant data for sentiment analysis.

3. **Sentiment Analysis:** The pre-processed input data is then passed through the sentiment analysis pipeline, which involves applying the CountVectorizer and MinMaxScaler to transform the input data into a format suitable for the machine learning model. - The transformed data is then fed into the trained machine learning

model like, XGBoost or Random Forest classifier to predict the sentiment of each input sentence.

4. **Result Visualization and Output:** After sentiment analysis, the system may generate visualizations such as word clouds or graphs to represent the sentiment distribution of the input data. - The system then provides the sentiment analysis results to the user, either by displaying the predicted sentiment for each input sentence or by generating a downloadable CSV file containing the predictions.

3.2.5 DEPLOYMENT DIAGRAM

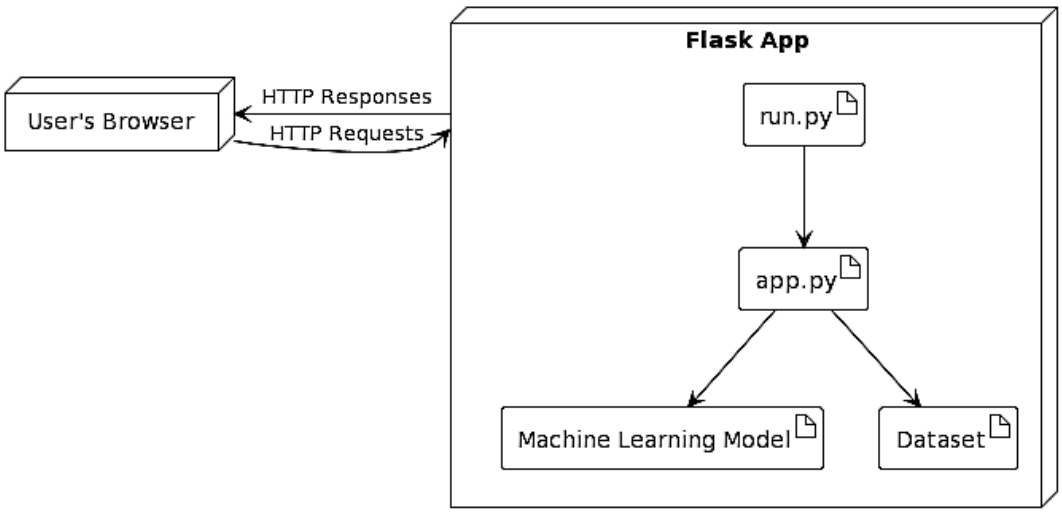

FIG:3.2 5 DEPLOYMENT DIAGRAM

NODES:

Browser: Represents the user's web browser.

Server: Represents the local machine where the Flask app is deployed.

Flask App: Represents the Flask application itself, containing its components.

ARTIFACTS:

app.py: The main Flask application script.

run.py: The script used to start the Flask development server.

Machine Learning Model: The trained machine learning model used by the app.

Dataset: The dataset used to train the machine learning model.

4. SYSTEM IMPLEMENTATION

4.1 DATASET:

The dataset for this project is sourced from Amazon's official Alexa skill store. The dataset encompasses the 5th generation of Echo Dot reviews, ranging from 2023, It contains a total of 3150 customer reviews. The dataset is in .tsv format and can be processed using the pandas library.

Useful Attributes: Verified_Reviews, Feedback, Length.

Useless Attributes: Date, Variation.

```
[6]: data = pd.read_csv("amazon_alexa.tsv", delimiter = '\t',)

     print(f"Dataset shape : {data.shape}")

     Dataset shape : (3150, 5)

[7]: data.head()
```

Date	Verified_Reviews	Feedback	Length
31-Jul-23	Love my Echo!	1	13
31-Jul-23	Loved It!	1	7
29-Jul-23	Fun item to play with and sometimes a hard time answering...	1	58
27-Jul-23	Alexa is great and plays my sleep songs with a single command.!	1	63
27-Jul-23	It's like Siri in fact Siri answers more accurately than Alexa. However good one.	1	81
25-Jul-23	Not much features...	0	20
25-Jul-23	Really disappointed Alexa has to be plugin to wall socket all the time	0	70
25-Jul-23	It worked for a month or so then it stopped	0	43

Fig 4.1.1 Data set selection

4.2 STATISTICAL MEASURES:

```
[32]: data['length'].describe()

[32]: count    3150.000000
      mean      132.673651
      std       182.526953
      min         1.000000
      25%        30.000000
      50%        74.000000
      75%       166.000000
      max      2853.000000
      Name: length, dtype: float64
```

Fig 4.2.1 Statistical Measures

Count: Total number of non-null values in the column. In this case, there are 3150 non-null values.

Mean: The average value of the column. The average length is approximately 132.67.

Std: The standard deviation, which measures the dispersion or spread of the values. It's approximately 182.53, indicating a

wide variation in the lengths.

Min: The minimum value in the column. The shortest length is 1.

25%: The value below which 25% of the data falls. Also known as the first quartile or Q1. In this case, 25% of the data have a length of 30 or less.

50%: The median of the data, also known as the second quartile or Q2. Half of the data have a length of 74 or less.

75%: The value below which 75% of the data falls. Also known as the third quartile or Q3. In this case, 75% of the data have a length of 166 or less.

Max: The maximum value in the column. The longest length is 2853.

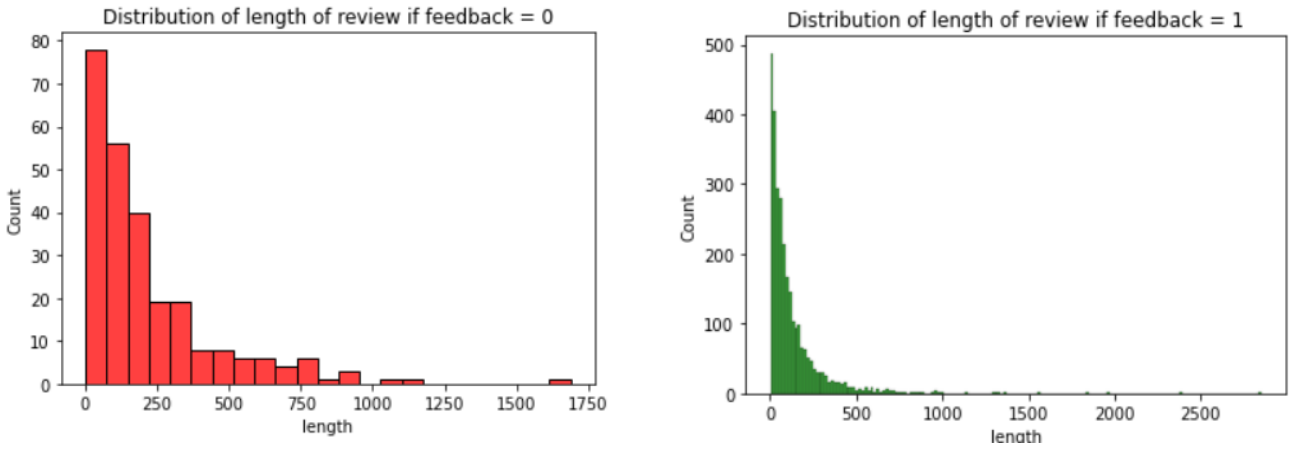

Fig 4.2.2 Positive and Negative Reviews Length Graph

4.3 DATA PREPROCESSING AND NLP TECHNIQUES:

1. **Stopword Removal:** This involves removing common words that do not carry much meaning, such as "the", "and", "a", etc. In this project, the *`nltk.corpus.stopwords`* module is used to remove stopwords from the text data.

2. **Stemming:** This involves reducing words to their root form, such as "running" to "run". In this project, the *`nltk.stem.PorterStemmer`* module is used to stem the words in the text data.

3. **Text Cleaning:** This involves removing any unwanted characters, symbols, or punctuation from the text data. In this project, the *`re.sub()`* method is used to replace any non-alphabetic characters with a space.

4. **Feature Scaling:** This involves scaling the numerical features in the data to a common scale to avoid bias towards certain features. In this project,`*sklearn.preprocessing.MinMaxScaler*` module is used to scale the numerical features in the data.

4.4 CODE

a) Data Preprocessing and NLP techniques code:

```
corpus = []
for i in range(0,3150):
  review = re.sub('[^a-zA-Z]', ' ', data['Review'][i])
  review = review.lower()
  review = review.split()
  stemmer = PorterStemmer()
  all_stopwords = stopwords.words('english')
  all_stopwords.remove('not')
  review = [stemmer.stem(word) for word in review if not word in set(all_stopwords)]
  review = ' '.join(review)
  corpus.append(review)
```

```
data['cleaned_text']=corpus
```

b) Data Splitting and Code:

Splitting the Data: The dataset is split into training and testing sets using the train_test_split function, where 70% of the data is used for training and 30% of the data is used for testing.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,

print(f"X train: {X_train.shape}")
print(f"y train: {y_train.shape}")
print(f"X test: {X_test.shape}")
print(f"y test: {y_test.shape}")
```

```
X train: (2205, 2500)
y train: (2205,)
X test: (945, 2500)
y test: (945,)
```

c) Implementation:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
from xgboost import XGBClassifier
data = pd.read_csv('content/amazon_alexa_Reviews.tsv', delimiter='\t')
corpus = []
for i in range(0,3150):
review = re.sub('[^a-zA-Z]', ' ', data['Review'][i])
review = review.lower()
review = review.split()
stemmer = PorterStemmer()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
review = [stemmer.stem(word) for word in review if not word in set(all_stopwords)]
review = ' '.join(review)
corpus.append(review)
data['cleaned_text']=corpus
data.head()
#Fitting xgboost model
cv = CountVectorizer(max_features = 2500)
#Storing independent and dependent variables in X and y
X = cv.fit_transform(corpus).toarray()
y = data['Liked'].values
```

```
#Saving the Count Vectorizer
import pickle
pickle.dump(cv, open('content/countVectorizer.pkl', 'wb'))
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,random_state = 15)
scaler = MinMaxScaler()
X_train_scl = scaler.fit_transform(X_train)
X_test_scl = scaler.transform(X_test)
#Saving the scaler model
pickle.dump(scaler, open('content/scaler.pkl', 'wb'))
from xgboost import XGBClassifier
model_xgb = XGBClassifier()
model_xgb.fit(X_train_scl, y_train)
y_preds = model_xgb.predict(X_test)
#Saving the XGBoost classifier
pickle.dump(model_xgb, open('content/model_xgb.pkl', 'wb'))
```

4.5 CLASSIFICATION REPORTS:

a) XGBOOST Classification Report

	precision	recall	f1-score	support
0	0.81	0.38	0.52	78
1	0.95	0.99	0.97	867
accuracy			0.94	945
macro avg	0.88	0.69	0.75	945
weighted avg	0.94	0.94	0.93	945

b) RANDOM FOREST Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.37	0.52	78
1	0.95	0.99	0.97	867
accuracy			0.94	945
macro avg	0.90	0.68	0.74	945
weighted avg	0.94	0.94	0.93	945

c) DECISION TREE Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.31	0.51	0.38	78
1	0.95	0.90	0.92	867
accuracy			0.86	945
macro avg	0.63	0.70	0.65	945
weighted avg	0.90	0.86	0.88	945

5. OUTPUT SCREENS

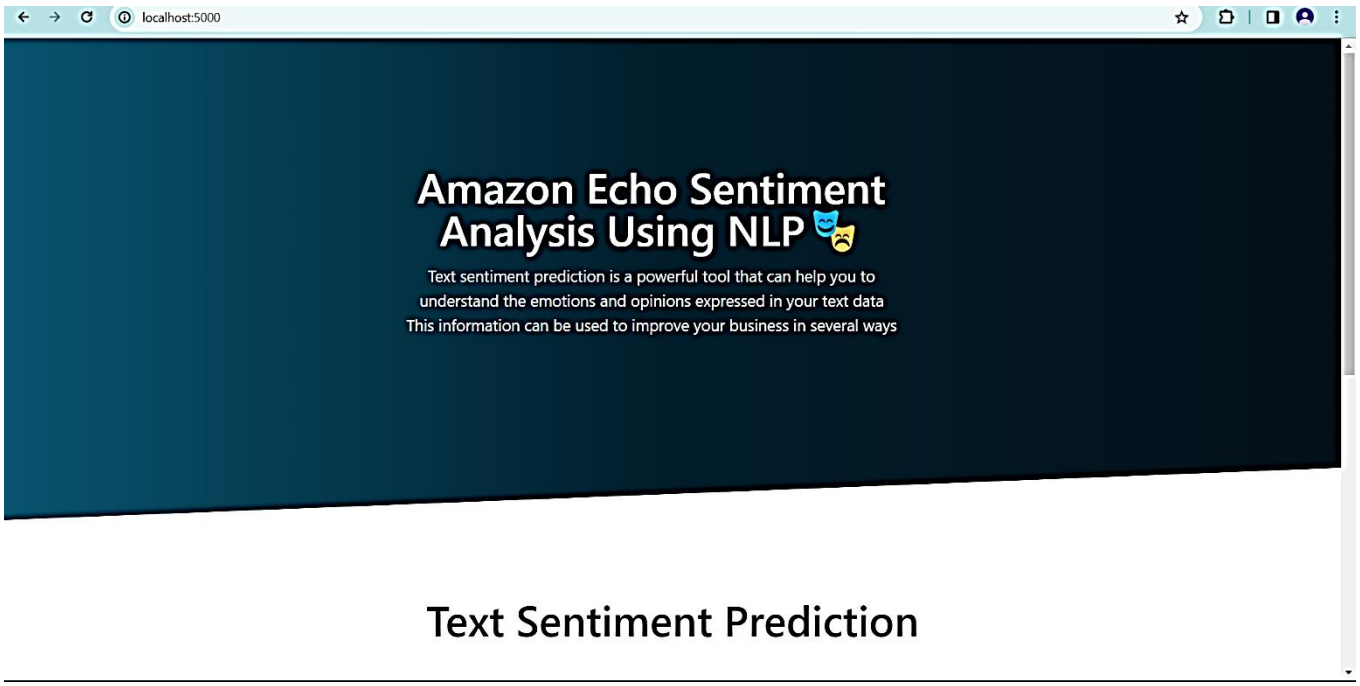

Fig 5.1 Flask Interface

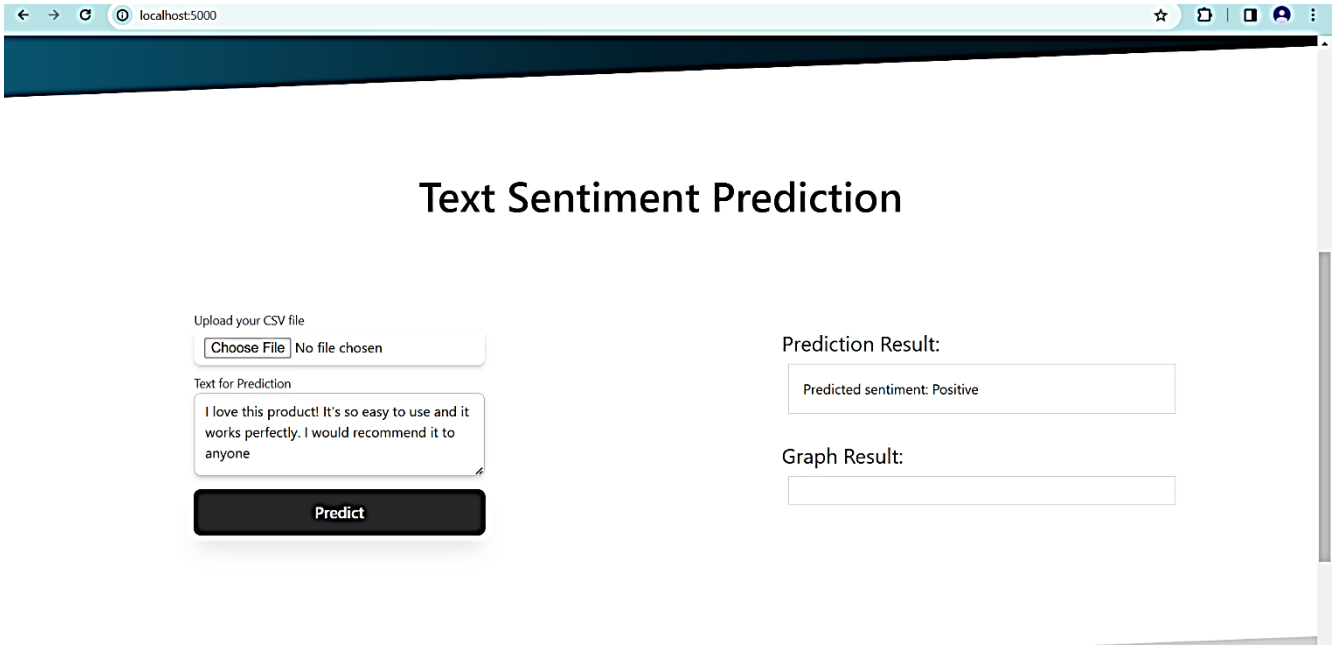

Fig 5.2 Positive Sentiment Example

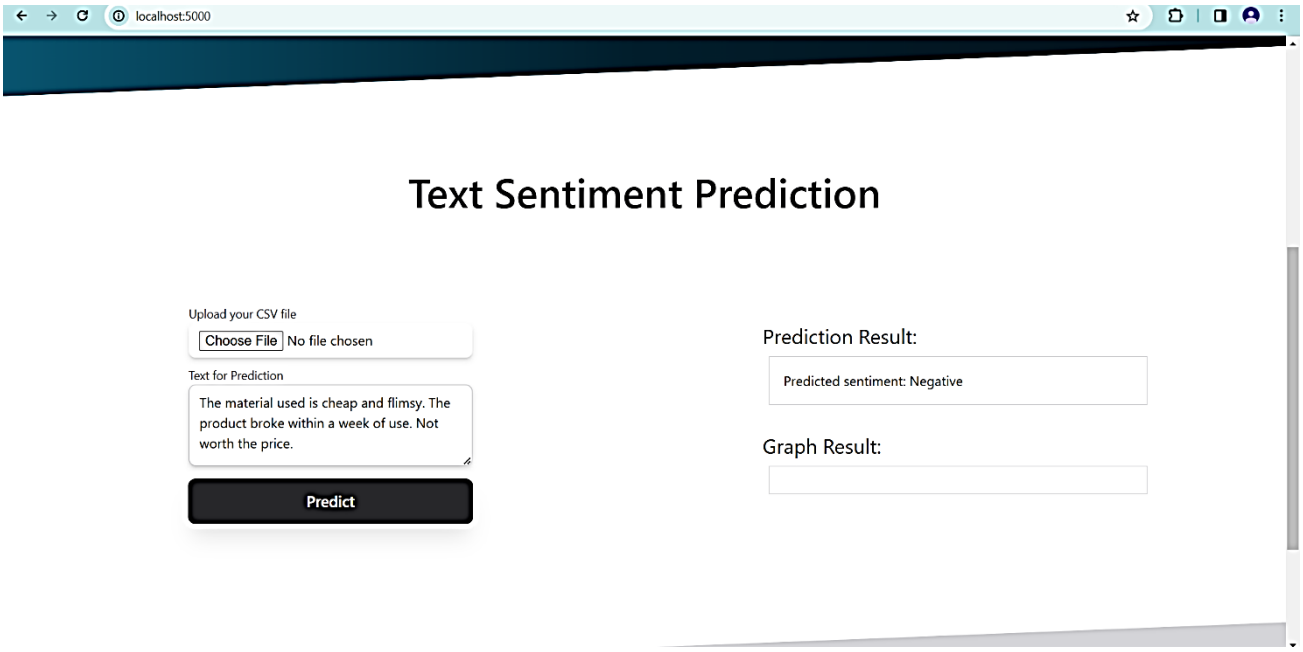

Fig 5.3 Negative Review Example

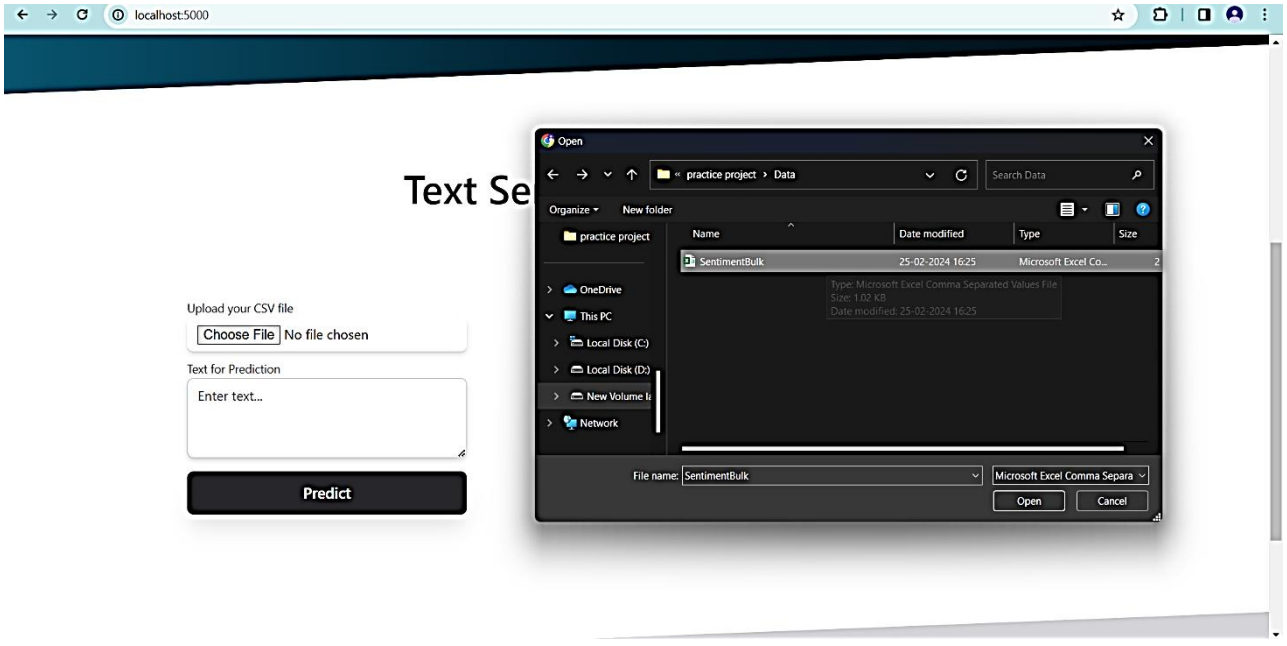

Fig 5.4 Bulk Sentiment Prediction

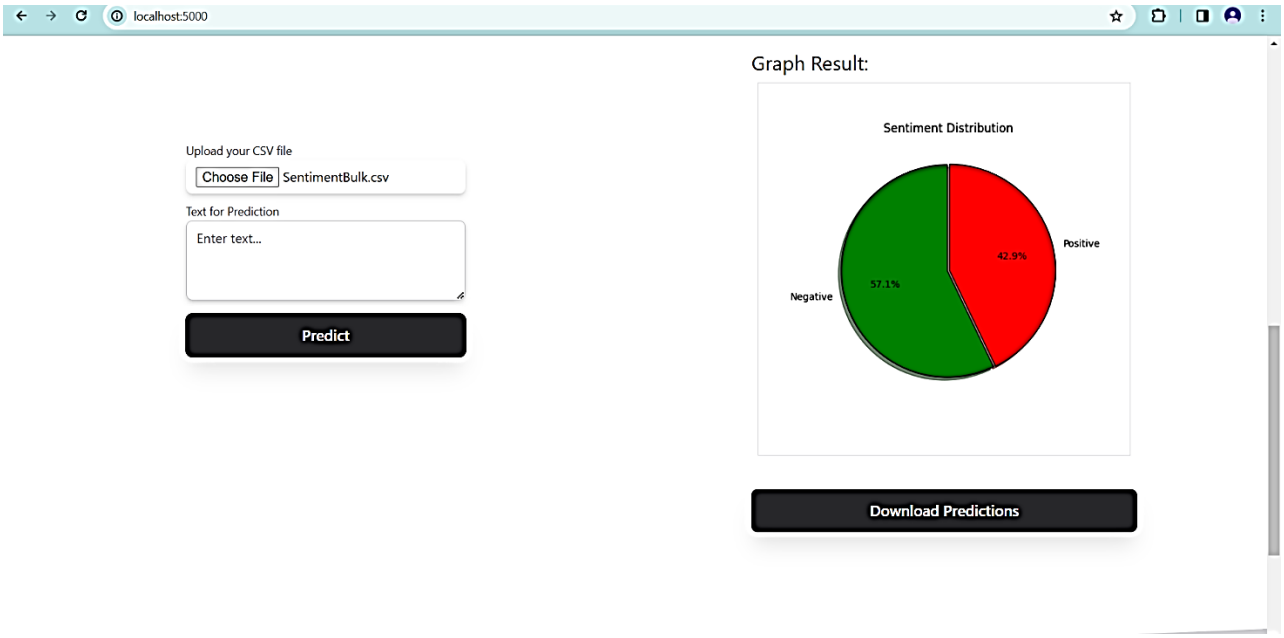

Fig 5.5 Graph Generation

Fig 5.6 Word Cloud Generation

6. CONCLUSION

The model aimed to enhance the sentiment analysis of Amazon Echo reviews by utilizing machine learning and natural language processing (NLP) techniques. By adopting this approach, a more adaptable, data-driven, and context-aware methodology was incorporated, which effectively overcame the limitations of a rule-based approach. Additionally, the interface offers several functionalities such as dataset selection, text input, and sentiment analysis result display.

The project comprises several critical components such as Data Selection, Data Preprocessing, NLP techniques, Data Splitting, Feature Scaling, Classification, and Result Generation. NLP techniques are used to preprocess the text data by executing tasks such as tokenization, lowercasing, removing stop words, stemming, and handling special characters. The preprocessed data is then split into training and testing sets. Machine learning models including XGBoost and Random forest classifiers are applied to the preprocessed and split data to perform sentiment analysis. The efficiency of these models is evaluated using metrics such as accuracy, precision, recall, and F1 score to determine their effectiveness in predicting sentiment.

7. FUTURE SCOPE

In the future, it is possible to incorporate more advanced Natural Language Processing (NLP) libraries such as spaCy into sentiment analysis operations. By incorporating pre-trained language models, the feature extraction and sentiment analysis processes can be improved. This would enable the implementation of real-time sentiment analysis capabilities to analyze sentiments from live Amazon reviews as they are posted, providing timely insights to users. Moreover, the use of aspect-level sentiment analysis would allow for a more in-depth understanding of specific aspects or features mentioned in the reviews, like product quality and customer service. This level of detail could provide essential insights into what customers like or dislike, enabling businesses to make more informed decisions about their products and services These libraries offer pre-trained language models that could more effectively capture the nuances of human language, resulting in more accurate sentiment analysis.

Furthermore, the integration of more advanced NLP libraries into sentiment analysis operations could revolutionize the way businesses analyze customer feedback. By providing more accurate and detailed insights, businesses could improve their products and services, resulting in happier customers and increased revenue.

8. BIBLIOGRAPHY:

[1] A. Bandi and A. Fellah, “Socio-analyzer: A sentiment analysis using social media data,” in Proc. 28th Int. Conf. Softw. Eng. Data Eng., in EPiC Series in Computing, vol. 64, F. Harris, S. Dascalu, S. Sharma, and R. Wu, Eds. Amsterdam, The Netherlands: EasyChair, 2019, pp. 61–67.

[2] U. Naseem, S. K. Khan, M. Farasat, and F. Ali Abusive language detection:
A comprehensive review. Boca Raton, FL, USA: CRC Press, 2013.

[3] Krishnendu Chakraborty, Machine Learning with Real World Projects. Infosys wingspan.

[4] C. C. Aggarwal and C. K. Reddy, Data Clustering: Algorithms and Applications. Boca Raton, FL, USA: CRC Press, 2013.

[5] N. Ahmad and J. Siddique, “Personality assessment using Twitter tweets,” Procedia Comput. Sci., vol. 112, pp. 1964–1973, Sep. 2017.

[6] T. Ahmad, A. Ramsay, and H. Ahmed, “Detecting emotions in English and Arabic tweets,” Information, vol. 10, no. 3, p. 98, Mar. 2019.