Sentiment Analysis of Movie Reviews

Submitted by:

Hardiksinh Solanki (249531990)

Dikshaben Patel (249432540)

Instructor: Prof.

Ajmery Sultana

Algoma University

Department of Computer Science

Final Report for COSC5406002 – Machine Learning

Brampton, Ontario, Canada

December 6th ,2024

Introduction:

Sentiment analysis is a Natural Language Processing (NLP) technique that identifies and categorizes the emotional tone expressed in a piece of text. In the context of movie reviews, it is particularly valuable for understanding public opinions and making informed decisions in areas such as recommendation systems, box office performance, and market investments in film production. By analyzing reviews, businesses can gauge audience reactions, predict trends, and improve the quality of movies.
This project aims to leverage sentiment analysis to classify movie reviews as positive or negative. It presents unique challenges, including handling unstructured text data, dealing with slang, abbreviations, and sarcasm, and transforming text into numerical representations using techniques like Bag of Words (BoW) and TF-IDF. Effective preprocessing and careful model selection are crucial for success.

Literature Review:

Joscha et. al, in their paper [1] devised and compared various techniques like Bag of words models, n-grams for using semantic information to improve the performance of sentiment analysis. The earlier approaches did not consider the semantic associations between sentences or documents parts. Research by A. Hogenboom et al. [2] neither compared the methodological variants nor provided a method to merge disclosure units in the most favorable manner. They aimed to improve the sentiment analysis by using Rhetoric Structure Theory (RST) as it gives a hierarchical representation at the document level. They proposed an integration of the grid search and weighting to find out the average scores of sentiment from Rhetoric Structure Theory (RST) tree. They encoded the binary data into the random forest by using feature engineering as it greatly reduced the complexity of original RST tree. They concluded that machine learning raised the balanced accuracy and gives a high F1 score of 71.9%.

Amir Hossein Yazdavar et al. in this paper [3] provided novel understanding of sentiment analysis problem containing numerated data in drug reviews. They analyzed sentences which contained quantitative terms to classify them into opinionated or non-opinionated and also to identify the polarity expressed by using fuzzy set theory. The development of fuzzy knowledge base was done by interviewing several doctors from various medical centers. Although the number of researches has been done in this field (Bhatia, et al., [4]) these do not consider the numerical (quantitative) data contained in the reviews while recognizing the sentiment polarity. Also, the training data used has a high domain dependency and hence cannot be used in different domains. They concluded that their proposed method knowledge engineering based on fuzzy sets was much simpler, efficient and has high accuracy of over 72% F1 value. Dhiraj Murthy in his paper [5] he identified what roles do tweets play in political elections. He pointed out that even though there were various researches and studies done to find out the political engagement of Twitter, no work was done to find out if these tweets were Predictive or Reactive. In his paper, he concluded that the tweets are more reactive than predictive. He found out that electoral success in not at all related to the success on Twitter and that various social media platforms were used to increase the popularity of a candidate by generating a buzz around them.

Ahmad Kamal in his paper [6] designed an opinion mining framework that facilitates objectivity or subjectivity analysis, feature extraction and review summarization etc. He used supervised machine learning approach for subjectivity and objectivity classification of reviews. The various techniques used by him were Naive Bayes, Decision Tree, Multilayer Perceptron and Bagging. He also improved mining performance by preventing irrelevant extraction and noise as in Kamal's paper. [7]. Humera Shaziya et al. in this paper [8] classified movie reviews for sentiment analysis using WEKA Tool. They enhanced the earlier work done in sentiment categorization which analyzes opinions which express either positive or negative sentiment. In this paper, they also considered the fact that reviews that have opinions from more than one person and a single review may express both the positive and negative sentiment. They conducted their experiment on WEKA and concluded that Naïve Bayes performs much better than SVM for movie reviews as well as text. Naive Bayes has an accuracy of 85.1%.

Main Goal :

The primary objective of this project is to develop a robust machine learning model that accurately predicts the sentiment of movie reviews. Key goals include:

- **Preprocessing and Feature Extraction:** Clean and transform text data for modeling.
- **Model Training and Evaluation:** Train various models and evaluate their performance using metrics like accuracy, precision, recall, and F1-score.
- **Prediction and Deployment:** Use the trained model to predict sentiments of new reviews and deploy it for real-time applications.
- **Comprehensive Reporting:** Document the entire process, from data preprocessing to deployment, and present the findings in a detailed report and presentation.

Rationale:

The motivation behind pursuing this project on Movie Review Sentiment Analysis stems from the growing importance of sentiment analysis in the digital age. With the rise of online platforms where people share their opinions on movies, understanding audience sentiment has become crucial for film producers, marketers, and streaming platforms. This project allows for a deeper exploration of how machine learning and Natural Language Processing (NLP) can be used to analyze and predict public opinion.
Why it's important:

- Industry Impact: Sentiment analysis helps improve decision-making in marketing, production, and investment.
- Consumer Insight: It provides valuable insights into audience preferences and expectations.
- Technological Advancement: This project contributes to advancing sentiment analysis techniques by exploring various machine learning models and preprocessing methods.

Benefits and New Knowledge:

- Practical Application: The project will demonstrate how NLP can be applied to real-world problems.
- Model Insights: By evaluating different models, we gain insights into their performance and best-use scenarios.

- Skill Development: It enhances skills in data science, machine learning, and NLP, which are highly relevant in today’s job market.

Scope:

The scope of this project involves the following steps:

- Data Collection and Preprocessing: Using the IMDB dataset of 50,000 movie reviews for training and testing.
- Feature Extraction: Techniques like Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).
- Model Training and Evaluation: Training models including Logistic Regression, Naive Bayes, Linear SVC, and Random Forest, and evaluating their performance using metrics such as accuracy, precision, recall, and F1-score.
- Prediction and Deployment: Deploying the best-performing model for sentiment prediction of new reviews.
- Documentation and Reporting: Creating a comprehensive report and presentation of the findings.

Limitations:

- Text Complexity: Challenges include handling sarcasm, slang, and ambiguous phrases.
- Model Limitations: No single model may perfectly predict sentiment in all cases.
- Resource Constraints: Computational limitations may affect the training time of complex models like Random Forest.

Timetable:

Here is the table which summarizes project timeline and milestones.

Week	**Goal**	**Task Description**
Week 1	Literature Review & Topic Finalization	Research sentiment analysis techniques and finalize project scope
Week 2	Data Collection and Pre-processing	Gather and preprocess the IMDB dataset of movie reviews (noise removal, stop word elimination, stemming).
Week 3	Feature Extraction and Model Training	Apply BoW and TF-IDF for feature extraction, and train models (Logistic Regression, Naive Bayes, SVM, Random Forest).
Week 4	Model Evaluation and Tuning	Evaluate model performance using accuracy, precision, recall, and F1-score. Tune hyperparameters for optimization.
Week 5	Prediction, Deployment, and Reporting	Implement predictions on new reviews, deploy the best model, and document findings for the final report and presentation.

Methods:

Materials:

This project was developed using the following hardware and software resources:

Hardware:

- Laptop with the following specifications:
  - Processor: Intel Core i7
  - RAM: 16GB
  - Storage: 512GB SSD

Software:

- Visual Studio Code
- Python 3.8 programming language with libraries like Pandas (pd), NumPy (np),NLTK (Natural Language Toolkit),Scikit-Learn (sklearn),Matplotlib ,Seaborn,Word Cloud and Joblib.
- Kaggle for sourcing the movie review dataset.

Project Approach:

The project adopted an iterative and adaptive methodology inspired by Agile principles. This approach allowed for systematic progress through key development phases, ensuring flexibility and continuous refinement. Incremental Development: Each phase of the project, such as data preprocessing, feature extraction, and model evaluation, was handled in iterations. This enabled focused attention on each component, ensuring comprehensive improvements at every step.

Procedure:

The methodology for the Sentiment Analysis of Movie Reviews project involves several key steps, from data acquisition to model evaluation and deployment. The workflow is designed to ensure efficient preprocessing, robust feature extraction, and the application of machine learning models for accurate sentiment classification. Below is a detailed breakdown of the methods based on the information in the provided presentation

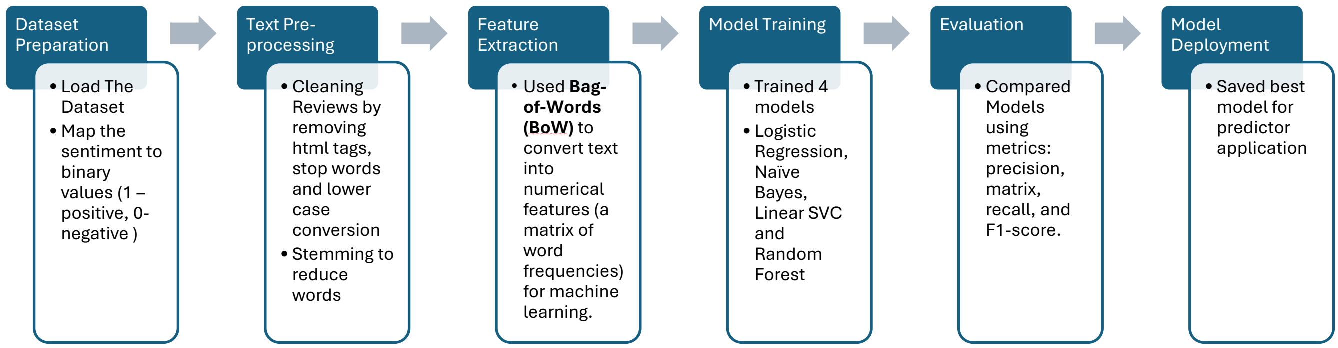

Figure 1: Flow diagram of the project

1. Data Collection:

Dataset: The project uses the IMDB dataset consisting of 50,000 movie reviews categorized into positive and negative sentiments. This dataset serves as the foundation for training and testing the machine learning models.

```
Initial Dataset Shape: (50000, 2)
The Shape of the data is as below:
(50000, 2)

Sample Data:
                                              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
```

Figure 2: first five rows of dataset

In this graph we can see the graphical representation of sentiment distribution,here in this dataset there is 25000 positive and 25000 negative reviews. This tells that the dataset is balanced and not bias on one feature.

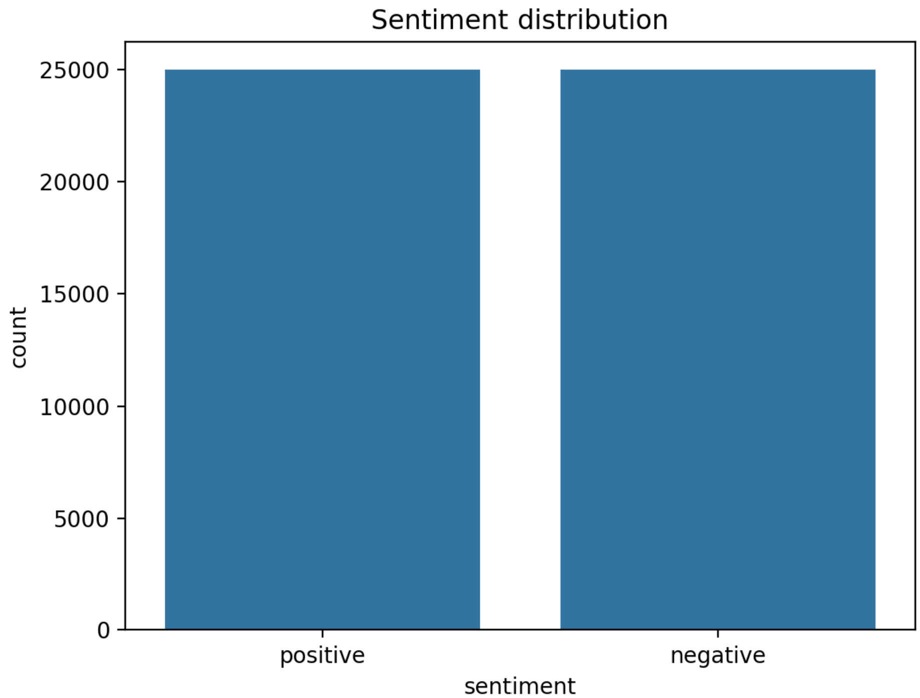

Figure 3 : Graph of sentiment distribution

2. Data Preprocessing:

Preprocessing is a crucial step in sentiment analysis as it cleans and standardizes the data, ensuring the machine learning model focuses on relevant features. Key steps include:

- **Text Normalization:**
  - Converting all text to lowercase.
  - Removing HTML tags, punctuation, and special characters.
  - Eliminating stop words and extra spaces.
- **Tokenization and Stemming:**
  - Splitting text into individual words (tokens).
  - Reducing words to their root forms using stemming techniques (e.g., "loved" becomes "love").
- **Example:**
  - Before preprocessing: *“This movie is AMAZING!! Loved it!”*
  - After preprocessing: *“movie amazing loved”*

Feature Extraction Using Bag of Words (BoW):
To convert text data into numerical vectors that machine learning models can process. The BoW model represents each review as a vector of word counts.**Process of extraction is to** Create a vocabulary from all unique words in the dataset.Each review is represented as a binary or frequency vector indicating the presence or count of words.
Example :

- **Reviews:**
  - “loved movie”

- "amazing acting"
- "amazing movie acting"

- **Word Index:** [loved, movie, amazing, acting]
- **Vectors:**
  - Review 1: [1, 1, 0, 0]
  - Review 2: [0, 0, 1, 1]
  - Review 3: [0, 1, 1, 1]

From the example,we can se that review 1 and review 2 used two words from the word index and review 3 has used 3 words.

Word Cloud:

Using word cloud we can visualize the most frequent occurring words in the dataset. Larger words in the word cloud indicate higher frequencies. This helps identify dominant terms in positive and negative reviews, providing insights into the dataset.

As we can see before preprocessing the data the word cloud of most frequent words represents that text data often contains noise, such as punctuation, stop words, special characters, html tags and irrelevant tokens. The word cloud generated from unprocessed reviews may display words like "the," "is," "movie," and punctuation marks, which do not contribute significantly to sentiment analysis. The word cloud is cluttered, with common, non-informative words overshadowing the meaningful ones.

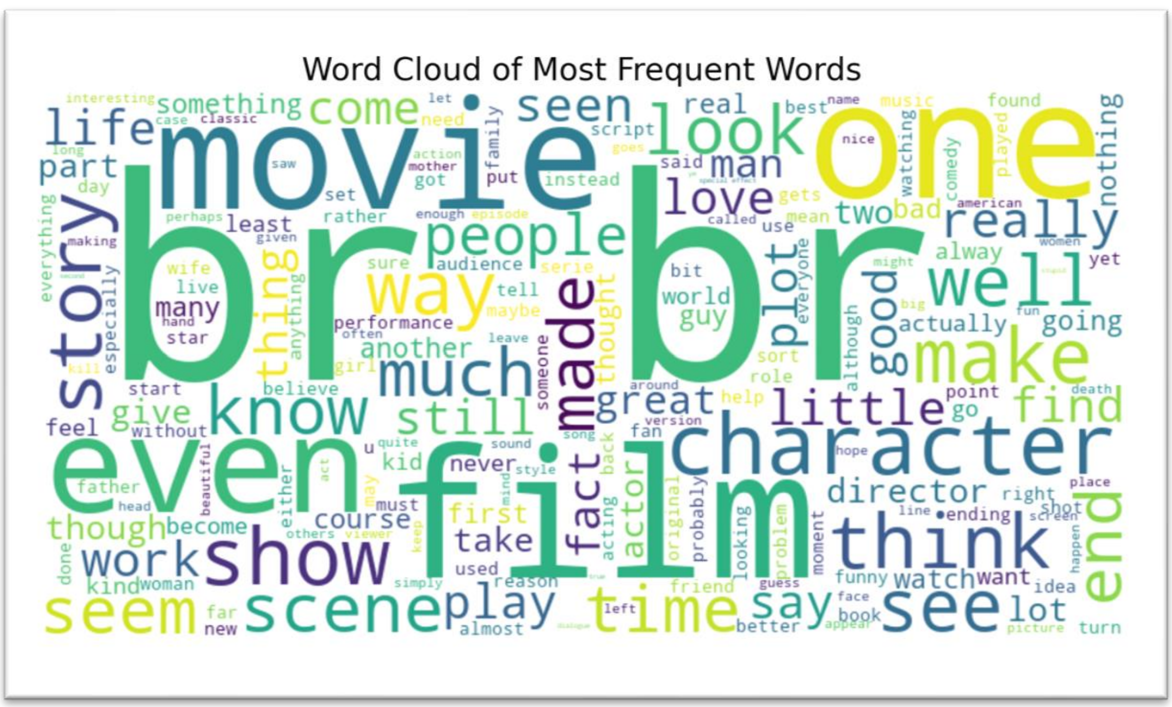

Figure 4:Word Cloud of most frequent words before preprocessing

Here are the word cloud visualization after preprocessing the data, we can we the difference that now the word cloud focuses on the most relevant words, excluding noise. By removing stop words and applying techniques like stemming, the processed word cloud highlights key terms that carry sentiment, such as *"amazing," "one," "film," "movie,"* which are critical for determining the review's sentiment.

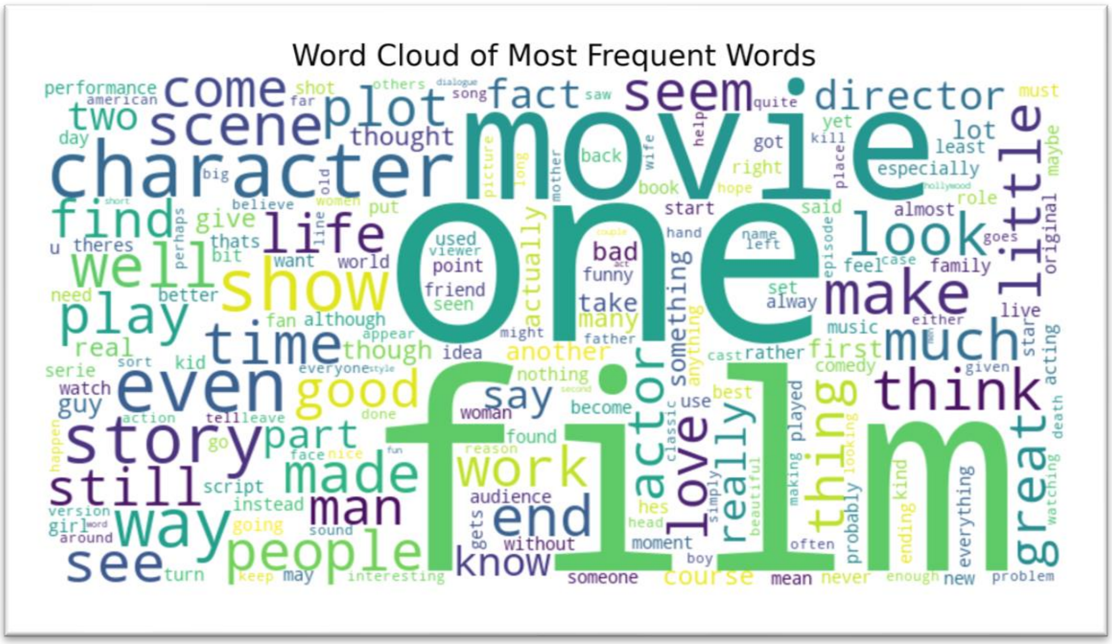

Figure 5:Word Cloud of most frequent words After preprocessing

The word cloud for positive reviews highlights key terms such as *"amazing," "love," "best,"* and *"excellent."* These words strongly convey positive emotions and are essential indicators of a favorable review. The prominence of these words suggests that users frequently use them to express satisfaction with movies.

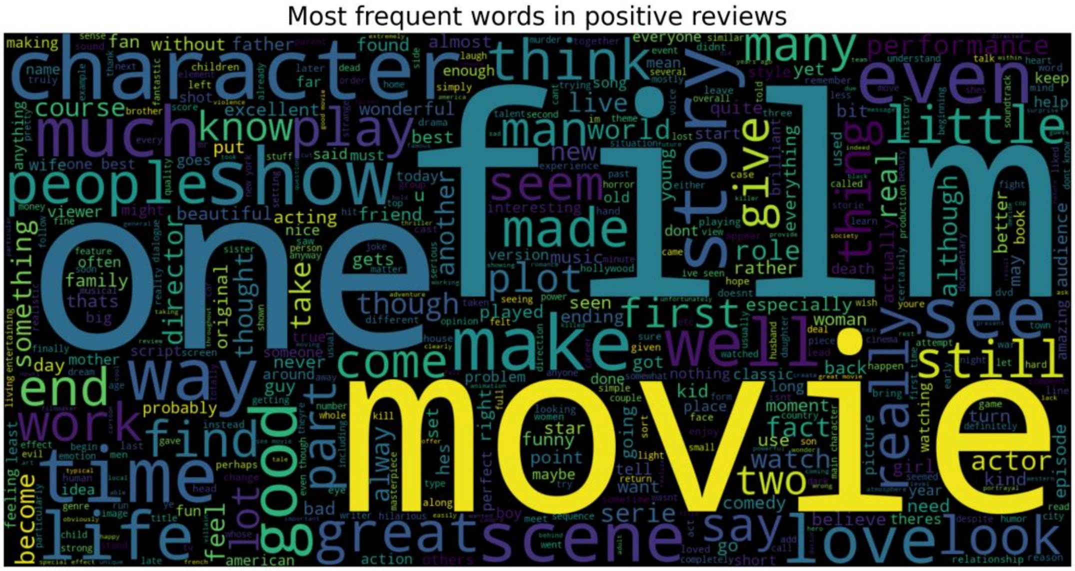

Figure 6:Word Cloud of most frequent word in positive reviews

For negative reviews, the word cloud emphasizes terms like *"boring," "worst," "disappointing,"* and *"terrible."* These words reflect dissatisfaction and are common in reviews expressing negative sentiments.

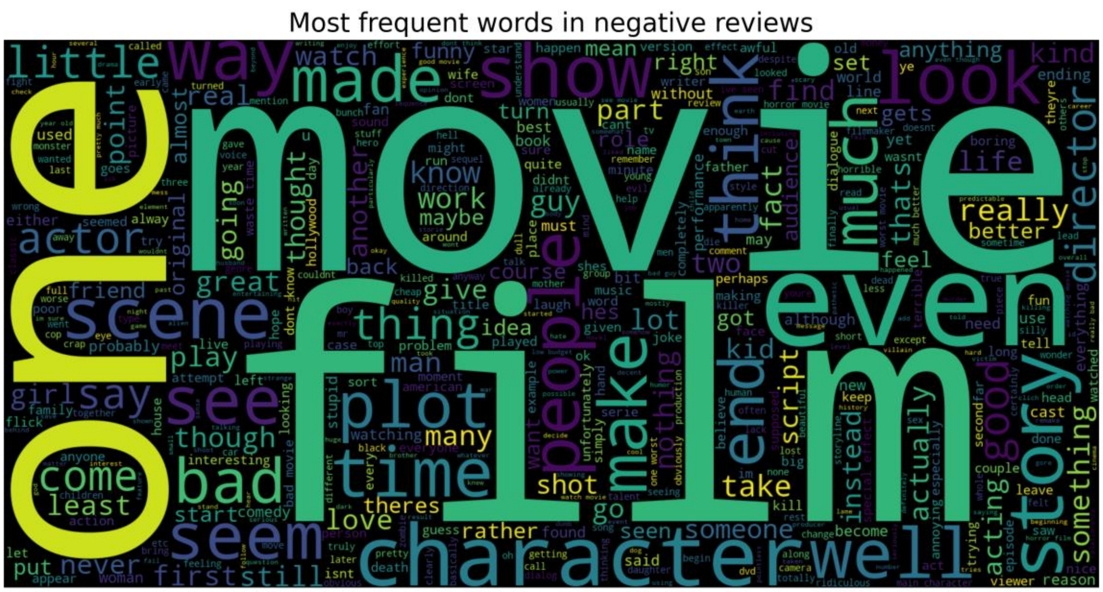

Figure 7:Word Cloud of most frequent word in positive reviews

The frequency graph shows the top words across the entire dataset, indicating their occurrence counts. Words like *"movie," "story,"* and *"acting"* appear frequently, regardless of sentiment. While these high-frequency words are important for context, they may not directly indicate sentiment. Therefore, feature selection techniques like TF-IDF are crucial to down weight such common terms and emphasize more sentiment-specific words.

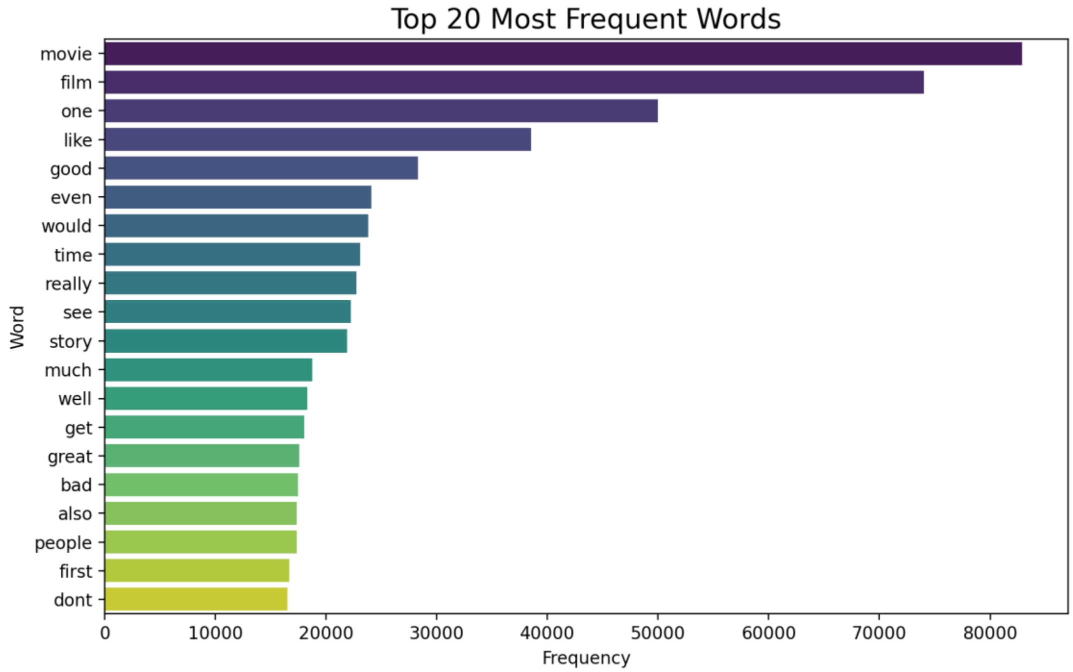

Figure 8:graph for top 20 most frequent words used in dataset

After preprocessing the dataset,we can see there is no missing values and also the 422 duplicate words are also removed .

```
PS D:\MCS-Algoma\Machine Learning\Project> & "C:/Program Files/Python312/python.exe" "d:/MCS-Algoma/M
[nltk_data] Downloading package stopwords to C:\Users\Hardiksinh
[nltk_data]     Solanki\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Initial Dataset Shape: (50000, 2)
The Shape of the data is as below:
(50000, 2)

Sample Data:
                                              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

Checking for Missing Values:
review       0
sentiment    0
dtype: int64

Sentiment Mapping Complete.

Preprocessing Text Data...
Text Preprocessing Complete.

Removed 422 duplicate reviews.

Generating Word Cloud...
```

Figure 9 : Dataset After preprocessing the dataset

3. Model Training and Evaluation

The training and evaluation of machine learning models are critical phases in the sentiment analysis project. The aim is to develop models that can accurately classify movie reviews into positive or negative categories. Several machine learning models are trained and evaluated to find the most suitable one for sentiment classification. The models include:

- **Logistic Regression:** A simple yet effective linear model that predicts the probability of a positive or negative sentiment. It was chosen for its interpretability and high accuracy on text data.
- **Naive Bayes:** A probabilistic classifier based on Bayes' theorem, effective for text classification tasks. It assumes independence between features and performs well even with small datasets.
- **Linear Support Vector Machine:** A powerful classifier that constructs a hyperplane to separate data points. SVM is effective in high-dimensional spaces, making it suitable for text data.
- **Random Forest:** An ensemble method that builds multiple decision trees and merges their predictions to improve accuracy and control overfitting. It’s robust but computationally intensive.

Here, the dataset is split into training and testing sets, typically in an 80:20 ratio. The training set is used to build the model, while the test set evaluates its performance.

To assess the model's performance, various evaluation metrics are used:

- **Accuracy:** The percentage of correctly classified reviews.
- **Precision:** The proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** The proportion of true positives identified from all actual positive cases.
- **F1-Score:** The harmonic mean of precision and recall, useful for imbalanced datasets.
- **Confusion Matrix:** A table that shows the counts of true positives, true negatives, false positives, and false negatives, providing deeper insights into classification performance.

4. Model performance based on the evaluation metrics are shown in the Image below:

```
Results for Logistic Regression:
Training Time: 1.22 seconds
Evaluation Time: 0.01 seconds
Accuracy: 0.87
Precision: 0.87
Recall: 0.87
F1 Score: 0.87

Results for Naive Bayes:
Training Time: 0.04 seconds
Evaluation Time: 0.01 seconds
Accuracy: 0.85
Precision: 0.85
Recall: 0.84
F1 Score: 0.85

Results for Linear SVC:
Training Time: 4.71 seconds
Evaluation Time: 0.00 seconds
Accuracy: 0.86
Precision: 0.86
Recall: 0.87
F1 Score: 0.87
```

Figure 10:Results for each model performance

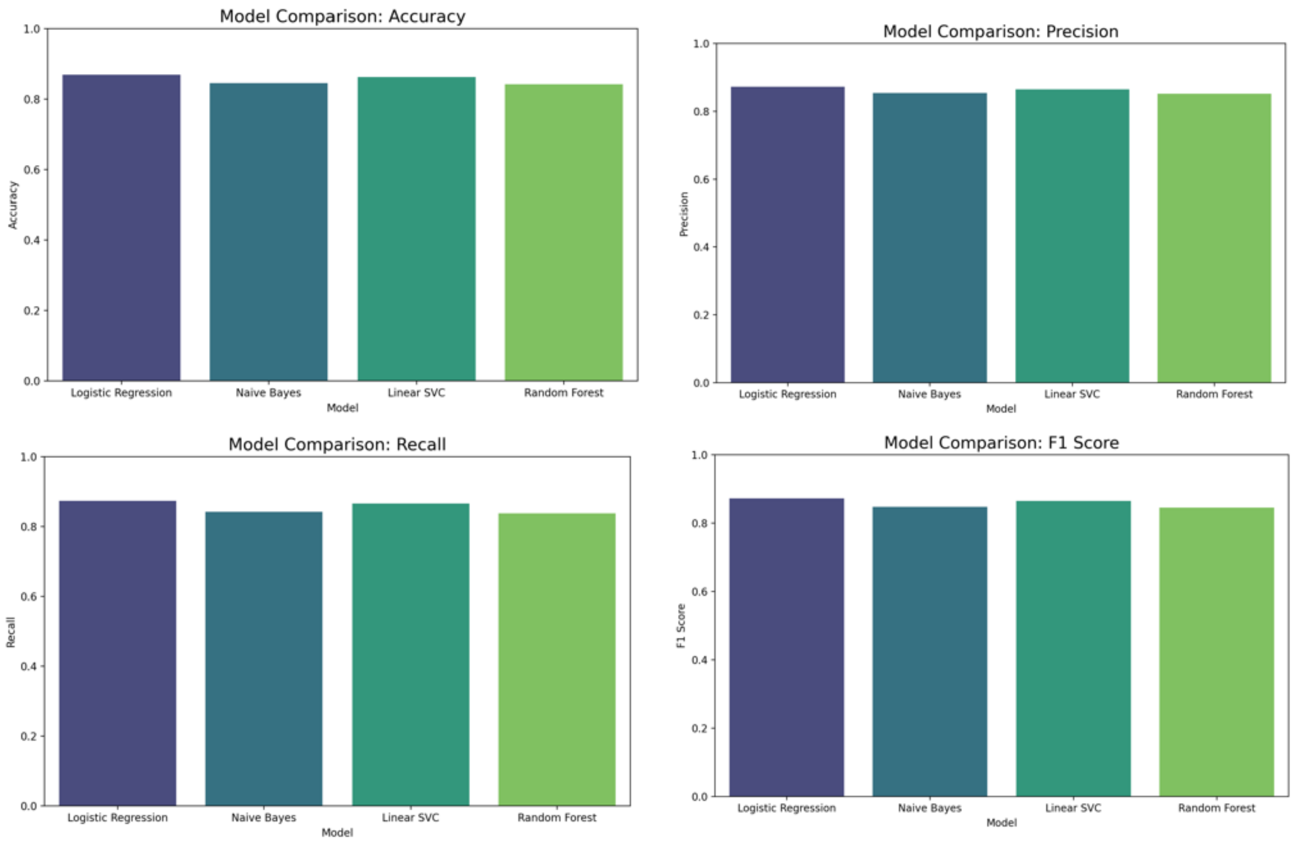

Figure 11: Model comparison based on evaluation metrics

```
Evaluation Metrics (Tabular Format):
                 Model  Training Time (s)  Evaluation Time (s)  Accuracy  Precision    Recall  F1 Score
0  Logistic Regression           1.223746             0.006029  0.869403   0.871800  0.873003  0.872401
1          Naive Bayes           0.039154             0.011120  0.845502   0.853971  0.841846  0.847865
2           Linear SVC           4.710786             0.004464  0.862041   0.864397  0.866101  0.865248
3        Random Forest         194.398807             0.409696  0.842477   0.851533  0.838099  0.844762

Best Model Based on Accuracy:
Model                  Logistic Regression
Training Time (s)                 1.223746
Evaluation Time (s)               0.006029
Accuracy                          0.869403
Precision                           0.8718
Recall                            0.873003
F1 Score                          0.872401
Name: 0, dtype: object
Model and Vectorizer saved successfully!
```

Figure 12:Evaluation metrics and best model

5. Best Model for Accuracy and F1 Score: Logistic Regression

Logistic Regression achieves the highest F1 score (0.8724) and accuracy (0.8694), making it the most balanced model in terms of precision and recall. This model is ideal for applications requiring a good balance of speed and performance.

Fastest Training Model: Naive Bayes
Naive Bayes is the fastest to train, with a time of just 0.012 seconds, making it exceptionally efficient for real-time or resource-constrained applications. While it's fast, its accuracy and F1 score do not match Logistic Regression or Linear SVC.

Most Time-Consuming Model: Random Forest
Random Forest, despite its robustness, has the slowest training time at 134.46 seconds. For this dataset, it is computationally expensive, requiring significant time, resources and does not outperform simpler models like Logistic Regression or Linear SVC in terms of accuracy or F1 score.

Close Competition: Linear SVC
Linear SVC delivers an accuracy of 0.8620 and an F1 score of 0.8652, which are close to those of Logistic Regression. However, its training time is significantly longer at 2.37 seconds. This means it perform slightly slower compared to Logistic Regression and less computationally efficient for large-scale datasets.

6. Result

The evaluation highlights Logistic Regression as the best overall model for this project due to its combination of high accuracy, balanced F1 score, and efficiency in training. Naive Bayes is suitable for scenarios requiring ultra-fast processing, while Random Forest and Linear SVC are better suited for more complex tasks where computational resources are not a constraint.

7. Web Application for Sentiment Analysis Predication For Movie Review:

As shown in image we have to enter the review for the movie and it will predict that is positive or negative review.

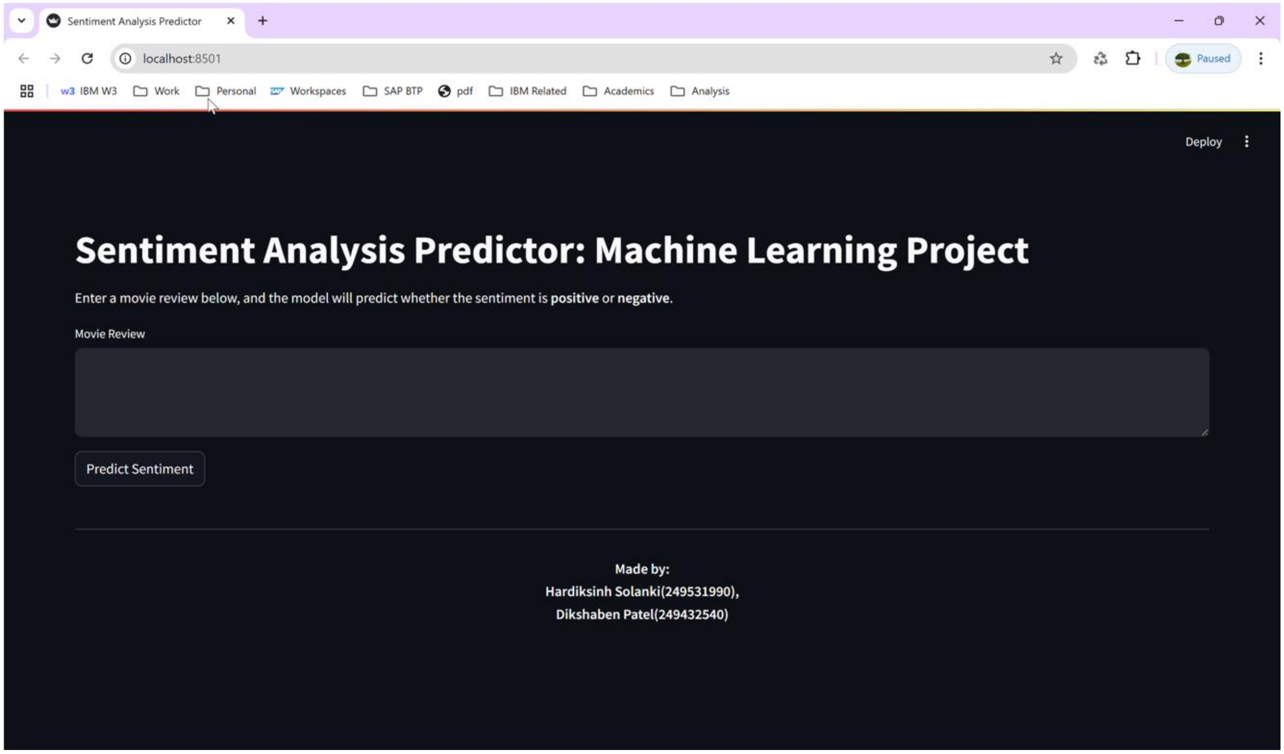

Figure 13 : Web app for sentiment analysis

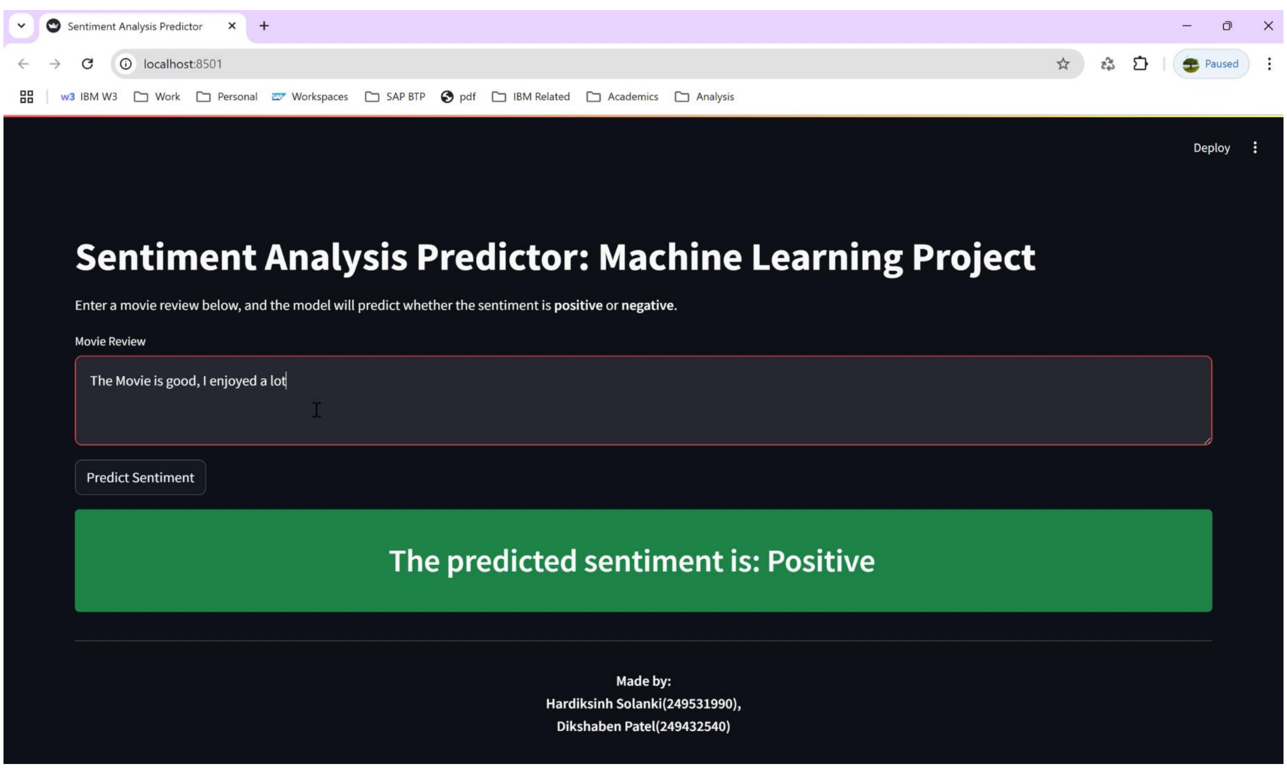

Figure 14: Positive Prediction based on movie review

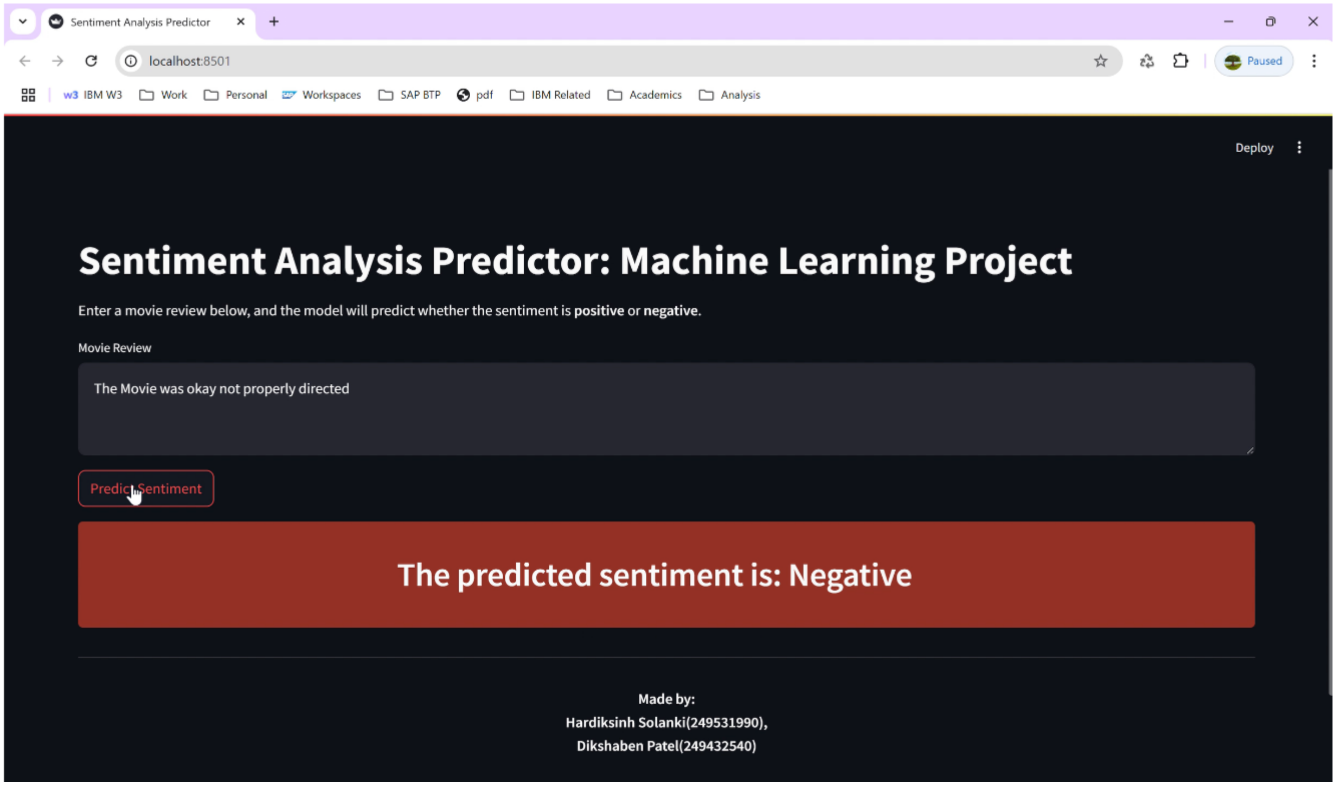

Figure 15: Negative Prediction based on movie review

Results and Conclusions

This project successfully implements sentiment analysis on movie reviews using machine learning techniques, achieving its objective of predicting the sentiment (positive or negative) of Movie reviews. By following a systematic approach, including data preprocessing, feature extraction, model training, and evaluation, the project highlights the potential of machine learning models to handle real-world textual data effectively.

The sentiment analysis system developed here can be extended to other domains, such as product reviews, social media analysis, and customer feedback systems, demonstrating its versatility and practical value. Expanding the project to include advanced techniques like deep learning (e.g., LSTMs or transformers) could further enhance performance. Additionally, integrating the model into a live application or API would provide a real-world deployment scenario, showcasing its practical application in various industries.

References

[1] J. Markle-Hub, S. Feuerriegel, and H. Prendinger, "Improving Sentiment Analysis with Document-Level Semantic Relationships from Rhetoric Discourse Structures," in *Proc. 50th Hawaii Int. Conf. System Sciences (HICSS)*, 2017.

[2] A. Hogenboom, F. Frasincar, F. de Jong, and U. Kaymak, "Using Rhetorical Structure in Sentiment Analysis," *Commun. ACM*, vol. 58, no. 7, pp. 69–77, 2015.

[3] A. H. Yazdavar, M. Ebrahimi, and N. Salim, "Fuzzy Based Implicit Sentiment Analysis on Quantitative Sentences," *J. Soft Comput. Decis. Support Syst.*, vol. 3, no. 4, pp. 7–18, 2016.

[4] R. S. Bhatia, A. Graystone, R. A. Davies, S. McClinton, J. Morin, and R. F. Davies, "Extracting Information for Generating a Diabetes Report Card from Free Text in Physicians' Notes," in *Proc. NAACL HLT 2010 2nd Louhi Workshop on Text and Data Mining of Health Documents*, 2010.

[5] D. Murthy, "Twitter and elections: Are tweets predictive, reactive, or a form of buzz?" *Inf. Commun. Soc.*, vol. 18, no. 7, pp. 816–831, 2015, DOI: 10.1080/1369118X.2015.1006659.

[6] A. Kamal, "Review Mining for Feature-Based Opinion Summarization and Visualization," *Int. J. Comput. Appl.*, vol. 179, no. 7, pp. 49–53, Dec. 2017.

[7] A. Kamal, "Subjectivity Classification using Machine Learning Techniques for Mining Feature-Opinion Pairs from Web Opinion Sources," *Int. J. Comput. Sci. Issues*, vol. 10, no. 5, pp. 191–200, 2013.

[8] H. Shaziya, G. Kavitha, and R. Zaheer, "Text Categorization of Movie Reviews for Sentiment Analysis," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 4, no. 11, pp. 45–51, 2015.

[9] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter Sentiment Analysis of Movie Reviews Using Machine Learning Techniques," School of Computer Science and Engineering, VIT University, Vellore, 2015.

[10] "Sentiment Analysis of Movie Reviews - Python Machine Learning," YouTube, 2023. [Online]. Available: https://youtu.be/yK9Ya6KzVAg. [Accessed: Dec. 6, 2024].

Appendices:

Here is the screenshot of code:

```python
# Step 5: Preprocessing the Text Data
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    # Lowercase the text
    text = text.lower()
    # Remove HTML tags
    text = re.sub('<br />', '', text)
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r"https\S+|www\S+|http\S+", '', text, flags = re.MULTILINE)
    # Remove non-alphabetic characters
    text = re.sub(r'[^a-z\s]', '', text)
    #Remove extra spaces
    text = re.sub(r'\s+', ' ', text).strip()
    # Remove stop words
    tokens = text.split()
    filtered_tokens = [word for word in tokens if word not in stop_words]
    return " ".join(filtered_tokens)

print("\nPreprocessing Text Data...")
df['cleaned_review'] = df['review'].apply(preprocess_text)
print("Text Preprocessing Complete.")
```

Figure 16: code for preprocessing dataset

```python
# Step 7 Generating Word Cloud
print("\nGenerating Word Cloud...")
wordcloud = WordCloud(width=800, height=400, background_color="white").generate(" ".join(df['cleaned_review']))
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Word Cloud of Most Frequent Words", fontsize=16)
plt.show()

# Positive Review Word Cloud
pos_reviews =  df[df.sentiment == 1]
text = ' '.join([word for word in pos_reviews['cleaned_review']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most frequent words in positive reviews', fontsize = 19)
plt.show()

#Negative Review Word Cloud
neg_reviews =  df[df.sentiment == 0]
text = ' '.join([word for word in neg_reviews['cleaned_review']])
plt.figure(figsize=(20,15), facecolor='None')
wordcloud = WordCloud(max_words=500, width=1600, height=800).generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most frequent words in negative reviews', fontsize = 19)
plt.show()
```

Figure 17: code for word cloud

```python
# Step 13: Train and Evaluate Models
results = []

# Logistic Regression
start = time.time()
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
training_time = time.time() - start
results.append(evaluate_model("Logistic Regression", log_reg, X_test, y_test, training_time))

# Naive Bayes
start = time.time()
nb_clf = MultinomialNB()
nb_clf.fit(X_train, y_train)
training_time = time.time() - start
results.append(evaluate_model("Naive Bayes", nb_clf, X_test, y_test, training_time))

# Linear Support Vector Machine
start = time.time()
linear_svc = LinearSVC(random_state=42, max_iter=1000)
linear_svc.fit(X_train, y_train)
training_time = time.time() - start
results.append(evaluate_model("Linear SVC", linear_svc, X_test, y_test, training_time))

# Random Forest
start = time.time()
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train, y_train)
training_time = time.time() - start
results.append(evaluate_model("Random Forest", rf_clf, X_test, y_test, training_time))
```

Figure 16 : code for evaluation of models

```python

# Step 17: Display Best Model
best_model = results_df.loc[results_df['Accuracy'].idxmax()]
print("\nBest Model Based on Accuracy:")
print(best_model)

# Step 18 Save the Model
# Save the trained Logistic Regression model
joblib.dump(log_reg, 'log_reg.pkl')

# Save the CountVectorizer
joblib.dump(vectorizer, 'count_vectorizer.pkl')

print("Model and Vectorizer saved successfully!")
```

Figure 17 : code for Best Model