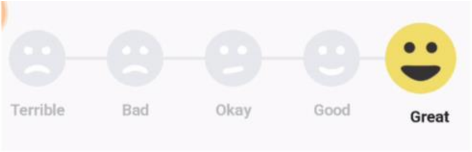
Algoma
UNIVERSITY

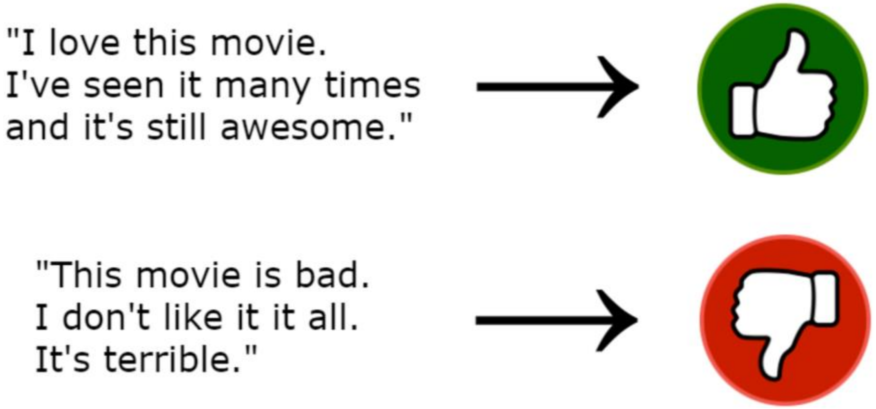
A MACHINE LEARNING(COSC540600224F) PROJECT

SENTIMENT ANALYSIS OF MOVIE REVIEWS



Guided By :
Prof. Ajmery Sultana

Submitted By:
Hardiksinh Solanki (249531990)
Dikshaben Patel (249432540)



"I love this movie.
I've seen it many times
and it's still awesome."

"This movie is bad.
I don't like it it all.
It's terrible."

OUTLINE

- INTRODUCTION TO SENTIMENT ANALYSSIS
- UNDERSTANDING THE PROBLEM STATEMENT
- DATASET DESCRIPTIONS
- PROCESS FLOW
- UNDERSTANDING PREPROCESSING
- MODELS AND EVALUATION
- RESULTS AND ANALYSIS
- A SHORT DEMO
- CONCLUSION

INTRODUCTION TO SENTIMENT ANALYSSIS

- **What is sentiment analysis?**
  - Sentiment analysis is a Natural Language Processing (NLP) technique used to determine the emotional tone or sentiment expressed in text.

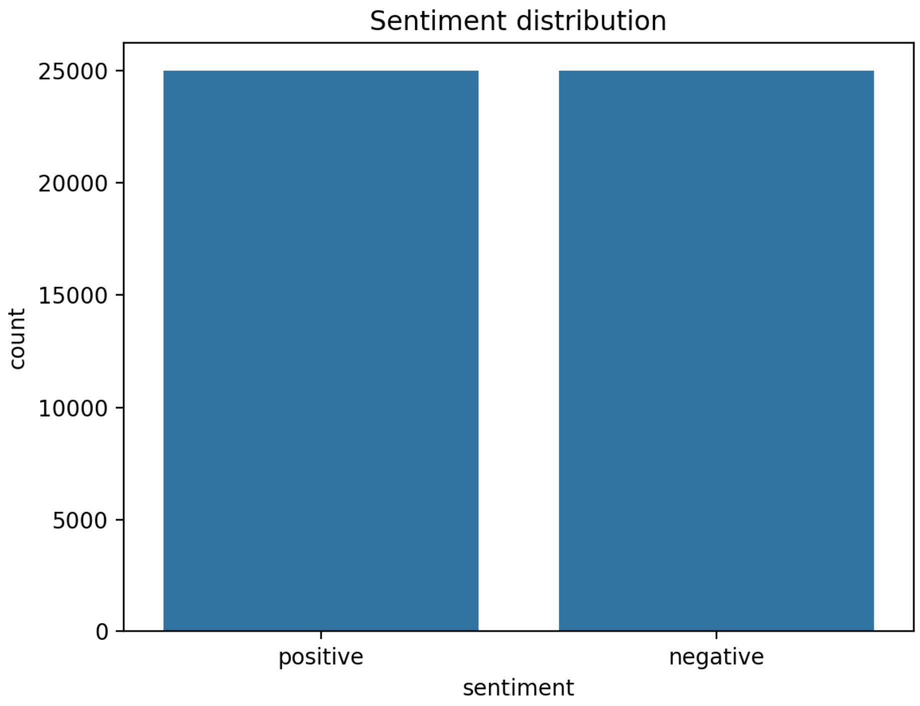
- **Why is it important in analyzing movie reviews?**
  - Understanding Public Opinion
  - Improved Decision-Making
  - Recommendation Systems
  - Quality Improvement of Movies
  - Trend Analysis
  - Box Office Prediction for market
  - Business Decisions for investments in movie productions

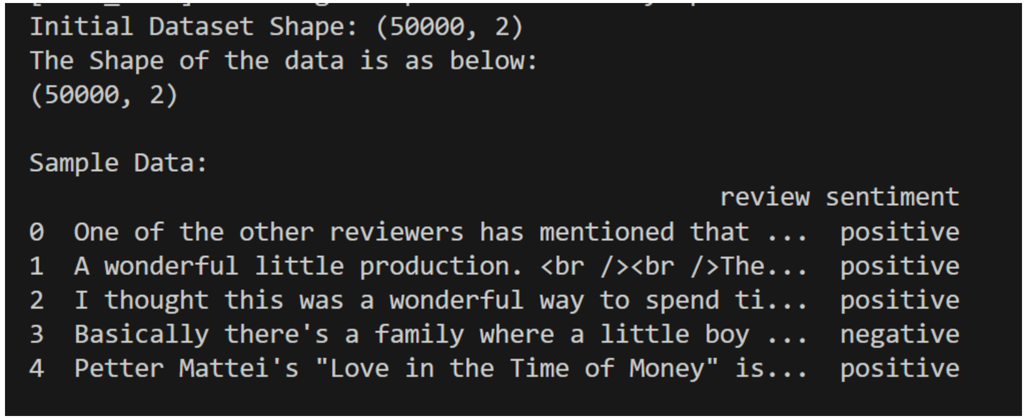
UNDERSTANDING THE PROBLEM STATEMENT

- **How can we determine whether a movie review is positive or negative based on its text?**
- **Challenges: Processing natural language.**
  - Converting textual data into numerical representations using techniques like Bag of Words (BoW) or TF-IDF.
  - Text data is unstructured and noisy, with slang, abbreviations, and misspellings (e.g., "gr8 movie!" vs. "great movie!").
  - Detecting sarcasm or irony is difficult (e.g., "This movie deserves an Oscar... for worst film!").
  - Phrases like "not bad" need special handling as they imply positivity despite the presence of "not."
- **Sentiment analysis for movie reviews is both a fascinating and demanding task, requiring robust preprocessing and careful model selection**.

DATASET DESCRIPTION

- **Dataset Used**: IMDB Dataset of 50K Movie Reviews.
- **Size**: 50,000 reviews.
- **Classes**: Positive (1) and Negative (0).
- No Missing Values
- 422 Duplicate Values Removed

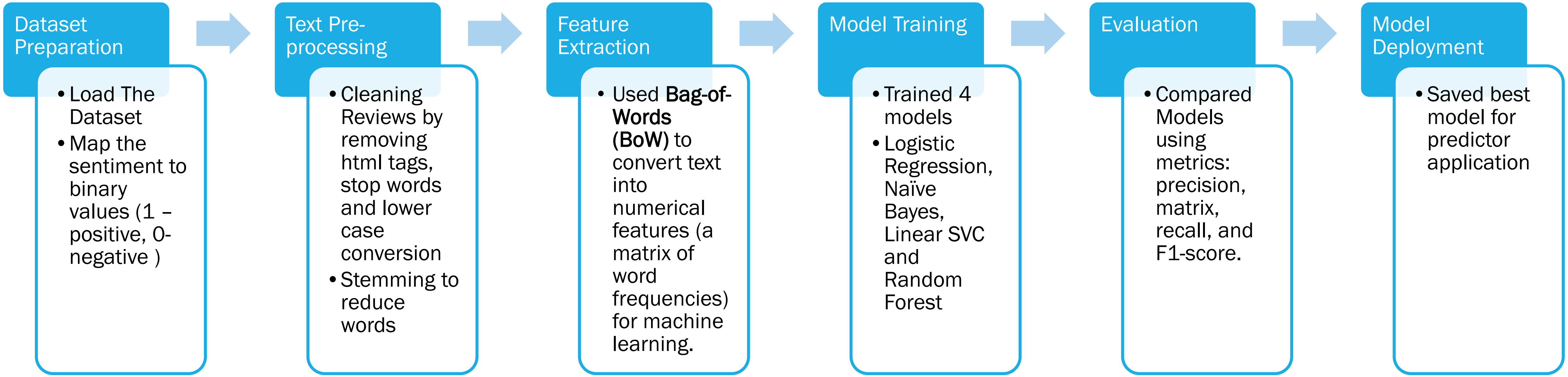




```
Initial Dataset Shape: (50000, 2)
The Shape of the data is as below:
(50000, 2)

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
```

PROCESS FLOW



Dataset Preparation

- Load The Dataset
- Map the sentiment to binary values (1 – positive, 0-negative)

Text Pre-processing

- Cleaning Reviews by removing html tags, stop words and lower case conversion
- Stemming to reduce words

Feature Extraction

- Used **Bag-of-Words (BoW)** to convert text into numerical features (a matrix of word frequencies) for machine learning.

Model Training

- Trained 4 models
- Logistic Regression, Naïve Bayes, Linear SVC and Random Forest

Evaluation

- Compared Models using metrics: precision, matrix, recall, and F1-score.

Model Deployment

- Saved best model for predictor application

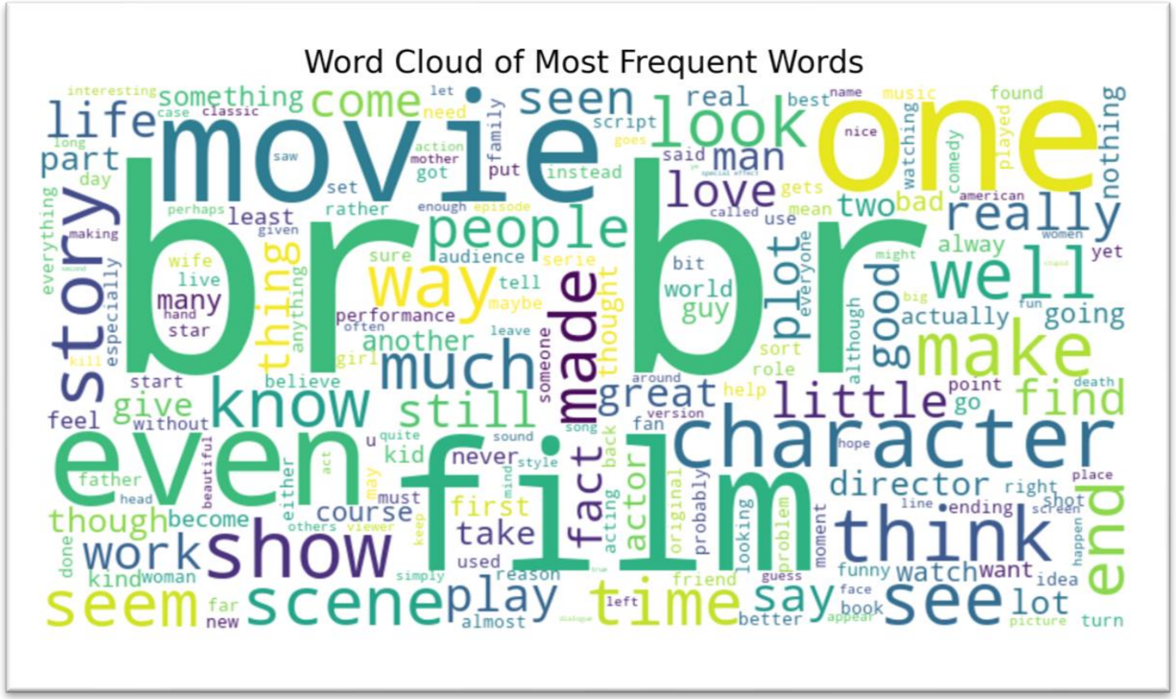
UNDERSTANDING DATA PREPROCESSING

- Converted text to Lowercase
- Removing HTML tags, punctuation, and special characters.
- Removing stop words, non-alphabetic characters and extra spaces.
- Stemming the Data.
  - **Before preprocessing**: "This movie is AMAZING!! Loved it!"
  - **After preprocessing**: "movie amazing loved"
- Feature Extraction using Bag of Words
  - Reviews : “loved movie” , “amazing acting” , “amazing movie acting”

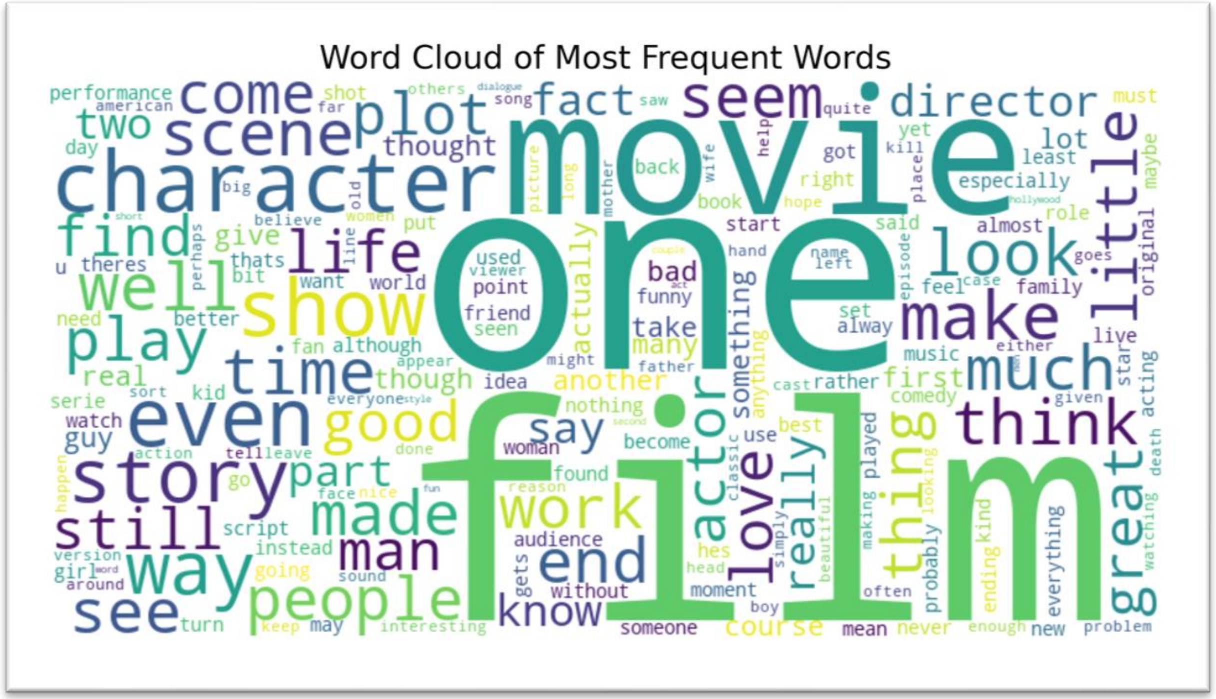
Word Index:	loved	movie	amazing	acting
Review 1:	1	1	0	0
Review 2:	0	0	1	1
Review 3:	0	1	1	1

- Word Cloud

WORD CLOUD BEFORE AND AFTER PRE-PROCESSING



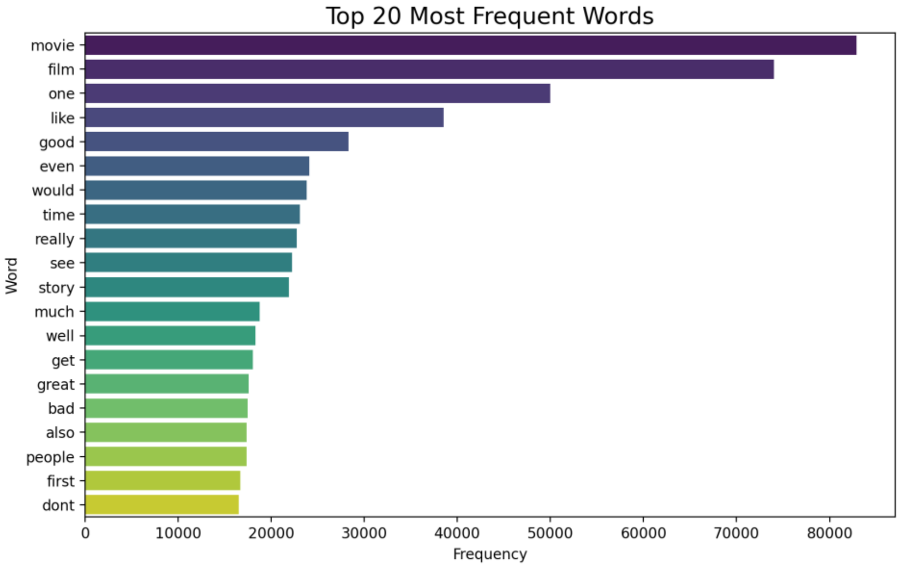
BEFORE PRE-PROCESSING



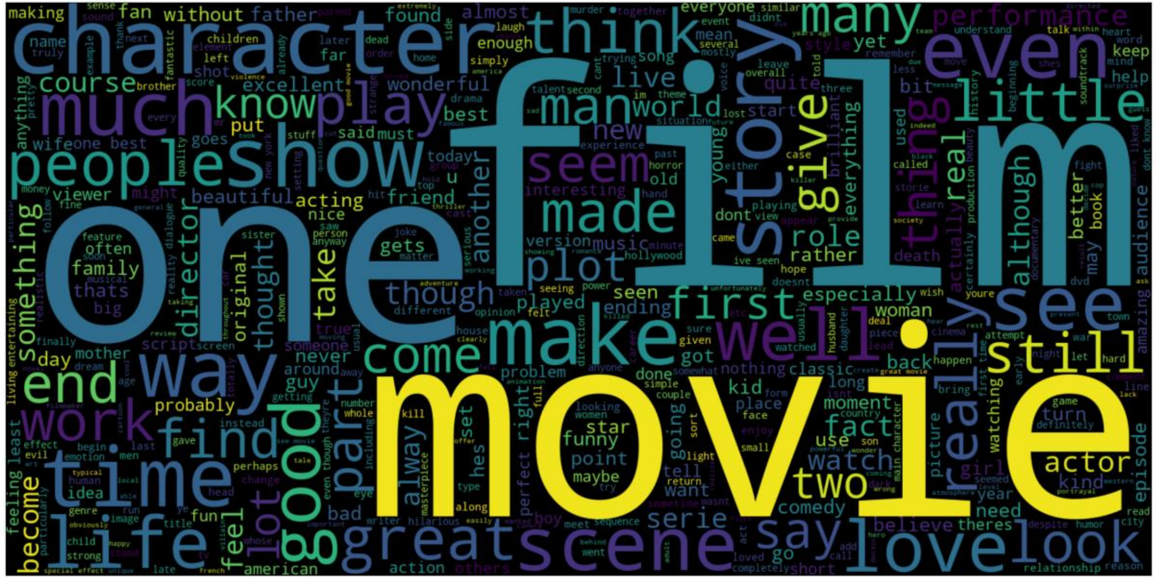
AFTER PRE-PROCESSING

IMPORTANCE OF DATA PREPROCESSING

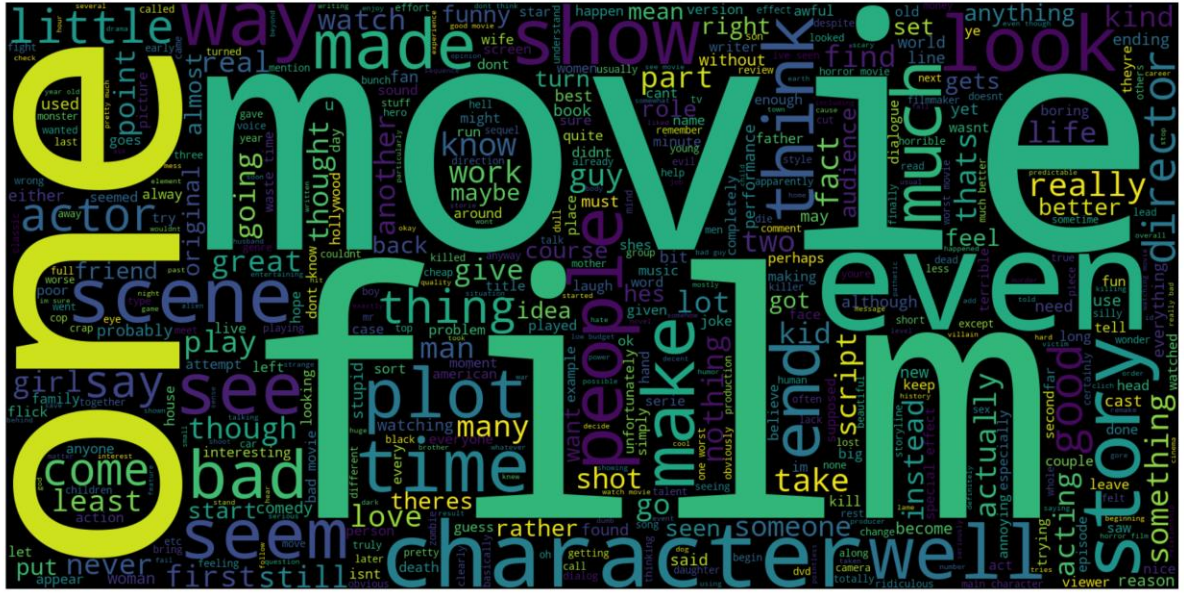
- **EDA Foundation:** Word Cloud and Frequent Word Analysis are essential for understanding data before modelling.
- **Model Improvement:** Helps refine preprocessing steps (e.g., stop word removal, html tags removal, special characters removal, stemming).
- **Interpretability:** Makes insights more accessible and actionable



Most frequent words in positive reviews



Most frequent words in negative reviews



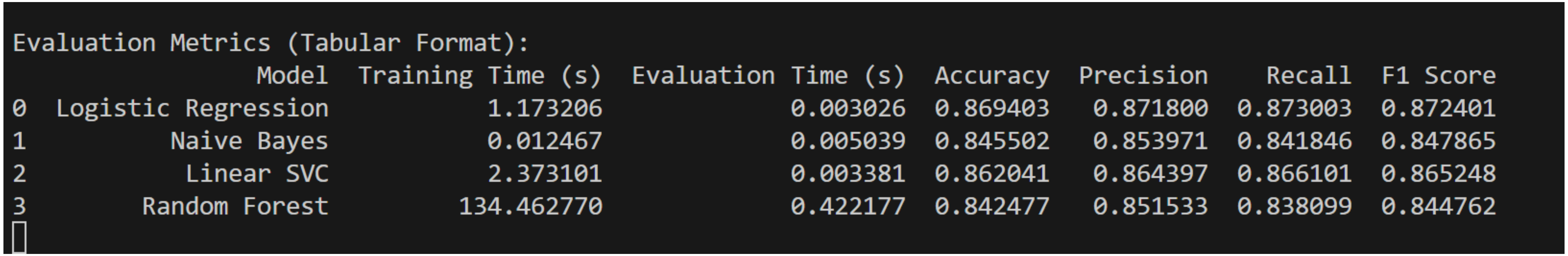
MAIN LIBRARIES USED:

- ❑ **Pandas (pd):** For data loading, cleaning, and manipulation.
- ❑ **NumPy (np):** To handle numerical computations efficiently.
- ❑ **NLTK (Natural Language Toolkit):** To preprocess text: remove stop words, apply stemming, and tokenize.
- ❑ **Scikit-Learn (sklearn):**
  - **Feature Extraction**:
    - **Count Vectorizer**: Converts text into numerical feature vectors using BoW.
  - **Model Training and Evaluation**:
    - Includes classifiers like Logistic Regression, Multinomial NB, Linear SVC, and Random Forest Classifier.
    - Metrics: Accuracy, Precision, Recall, F1-score (accuracy_score, etc.).
- ❑ **Matplotlib & Seaborn:** To create visualizations like word clouds and bar plots.
- ❑ **Word Cloud:** Generates word cloud visualizations to highlight the most frequent words in positive and negative reviews.
- ❑ **Joblib:** Saves and loads trained models and vectorizers efficiently.

MODELS AND EVALUATION

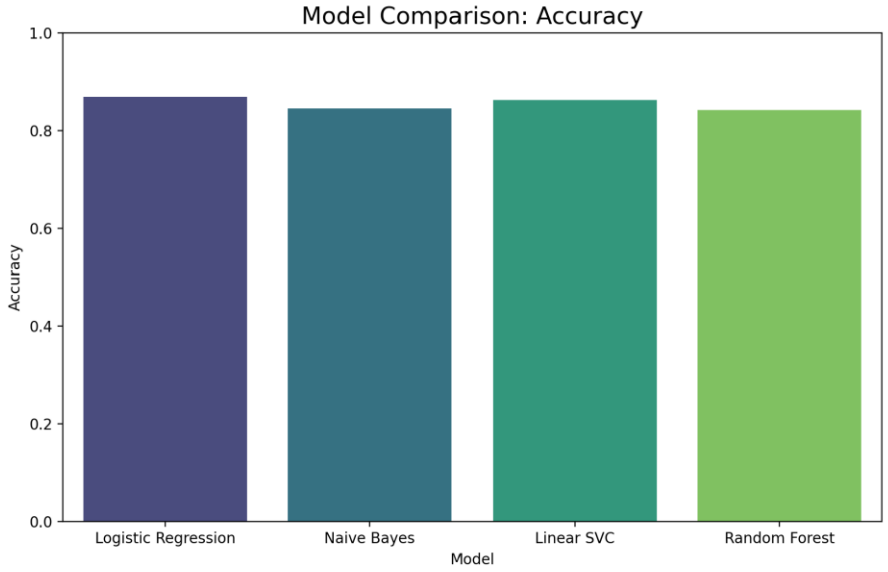
•The Models we have trained for evaluation are as follows:

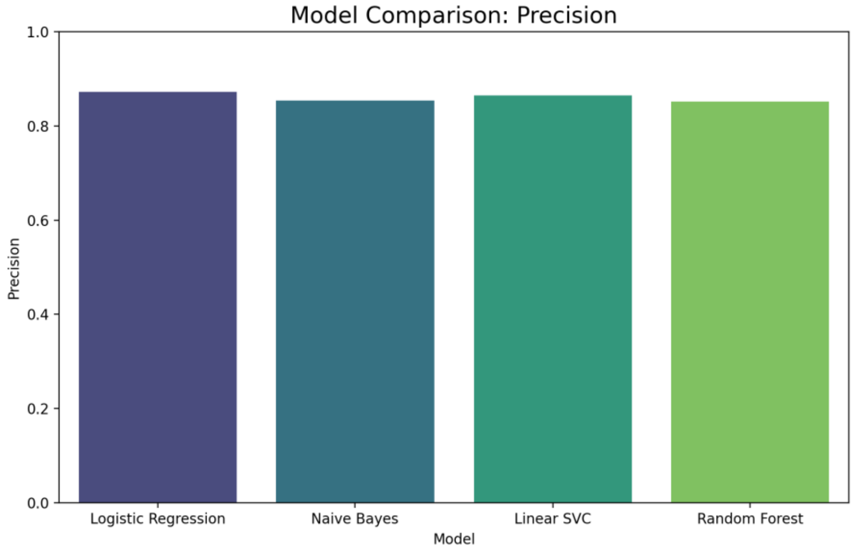
- Logistic Regression.
- Naive Bayes.
- Linear Support Vector Machine.
- Random Forest.

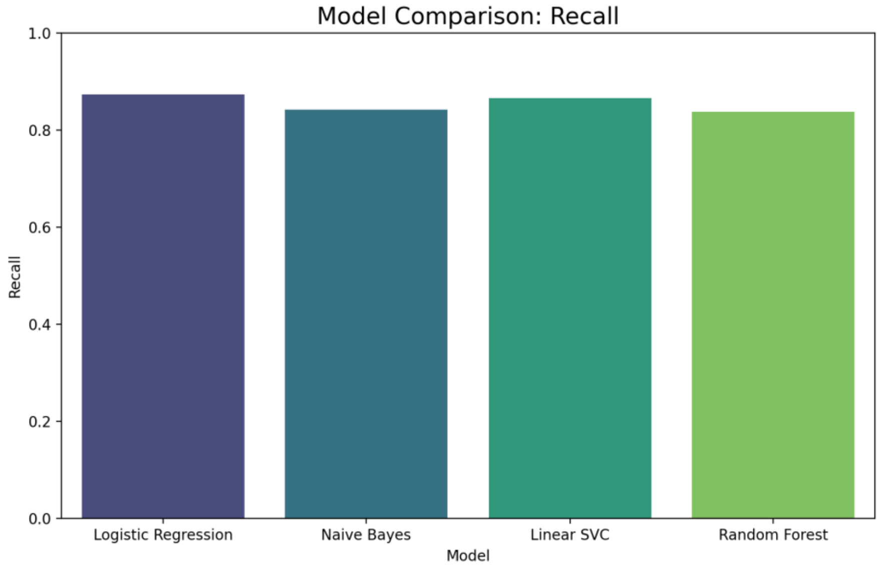


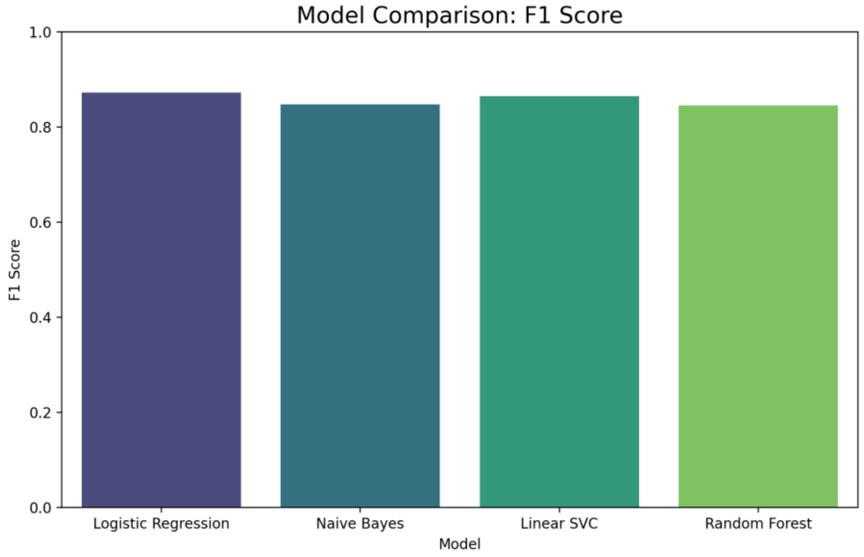
```
Evaluation Metrics (Tabular Format):
                Model  Training Time (s)  Evaluation Time (s)  Accuracy  Precision    Recall  F1 Score
0  Logistic Regression           1.173206             0.003026  0.869403   0.871800  0.873003  0.872401
1          Naive Bayes           0.012467             0.005039  0.845502   0.853971  0.841846  0.847865
2           Linear SVC           2.373101             0.003381  0.862041   0.864397  0.866101  0.865248
3        Random Forest         134.462770             0.422177  0.842477   0.851533  0.838099  0.844762
```

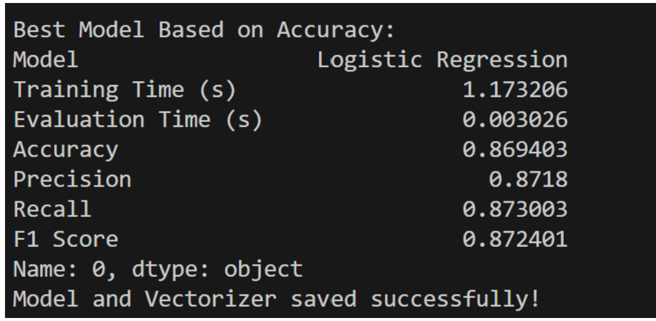
RESULTS AND ANALYSIS









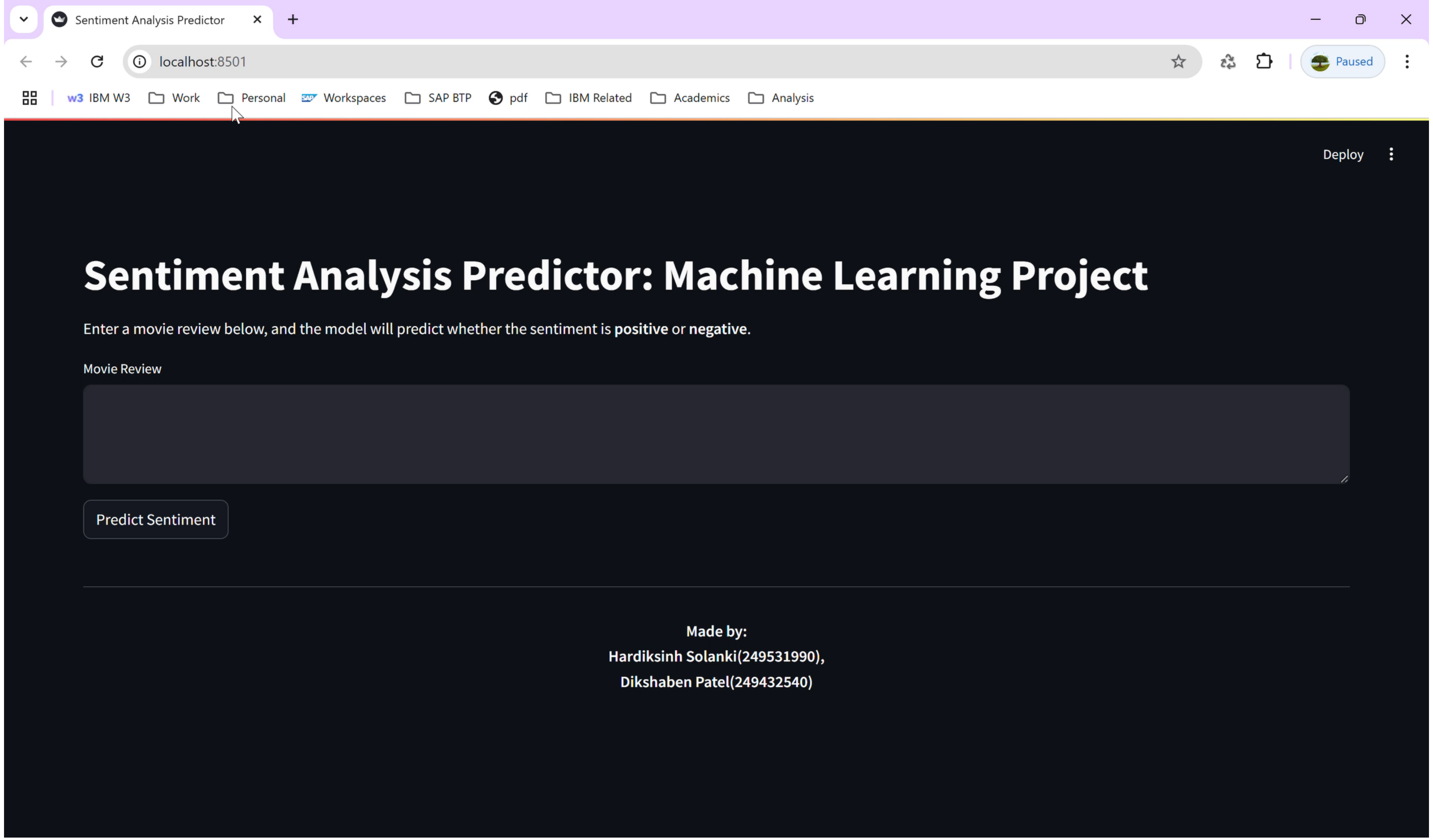


```
Best Model Based on Accuracy:
Model                Logistic Regression
Training Time (s)               1.173206
Evaluation Time (s)             0.003026
Accuracy                        0.869403
Precision                         0.8718
Recall                          0.873003
F1 Score                        0.872401
Name: 0, dtype: object
Model and Vectorizer saved successfully!
```

CONCLUSION BASED ON THE EVALUATION METRICS:

- **Best Model for Accuracy and F1 Score:**
  - **Logistic Regression** achieves the highest **F1 Score (0.8724)** and **Accuracy (0.8694)**, making it the best-performing model in terms of balanced performance.
- **Fastest Training Model**:
  - **Naive Bayes** has the fastest training time (0.012 seconds), making it ideal for scenarios where speed is a priority.
- **Most Time-Consuming Model**:
  - **Random Forest** takes the longest time to train (134.46 seconds), which is significantly slower than other models.
- **Close Competition**:
  - **Linear SVC** has comparable accuracy (0.8620) and F1 score (0.8652) to Logistic Regression but takes longer to train (2.37 seconds)
- **Trade-offs**:
  - While **Random Forest** is robust, it is computationally expensive and doesn't outperform Logistic Regression or Linear SVC in terms of accuracy or F1 score.

A SHORT DEMO



Sentiment Analysis Predictor: Machine Learning Project

Enter a movie review below, and the model will predict whether the sentiment is **positive** or **negative**.

Movie Review

Predict Sentiment

Made by:
Hardiksinh Solanki(249531990),
Dikshaben Patel(249432540)

THANK YOU