IMDb Movie Review Sentiment Analysis Project Report

---

1. Project Overview

This project aims to classify IMDb movie reviews as either positive or negative using NLP and machine learning. We use text preprocessing, feature extraction (TF-IDF), and classification models to predict sentiment with high accuracy. This helps stakeholders like filmmakers, platforms, and analysts understand audience opinion at scale.

Build a machine learning pipeline that:

- Preprocesses raw text
- Transforms it into numerical features
- Trains classification models to predict review sentiment
- Accepts user input to predict sentiment in real-time

2. Dataset Description

The dataset includes 50,000 IMDb reviews, each labeled with:

review: Raw movie review text

sentiment: Either "positive" or "negative"

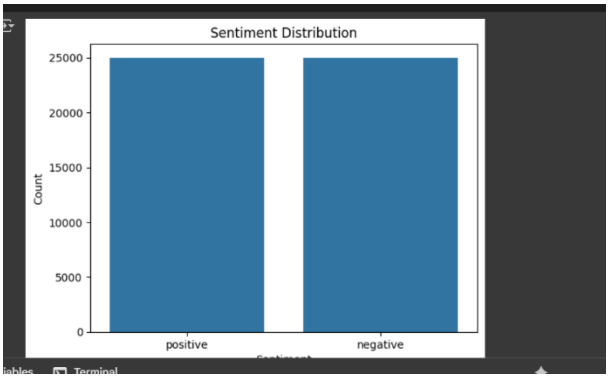
3. Exploratory Data Analysis (EDA)

Missing Values

- Checked with df.info() and .isnull().sum()
- Result: No missing values found.

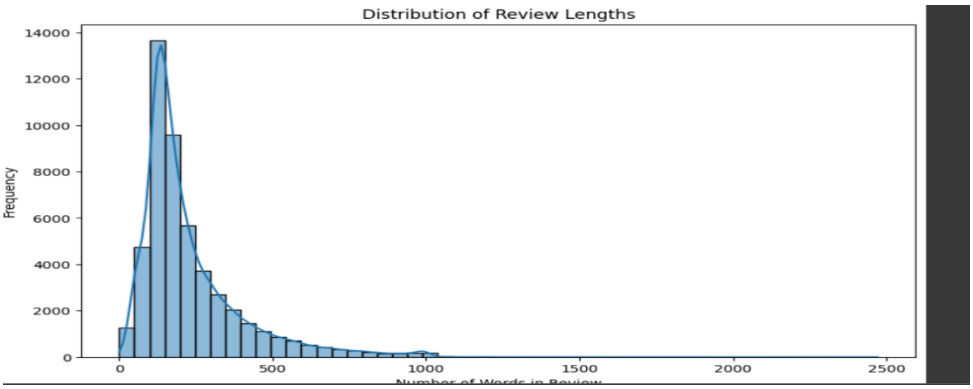
Sentiment Distribution

- Visualized with Seaborn countplot()
- Result: Roughly equal distribution of positive and negative reviews — no class imbalance.



Review Length

- Added a review_length column using df['review'].apply(len)
- Plotted with a histogram (sns.histplot)
- Observation: Most reviews are 50–300 words long.



5. Text Preprocessing

Used NLTK tools for:

Lowercasing

HTML tag and punctuation removal

Stopword removal (nltk.corpus.stopwords)

Tokenization (used split() for speed)

Stemming and Lemmatization via PorterStemmer and WordNetLemmatizer

6. Feature Engineering

TF-IDF Vectorization (Transform the textual data into numerical features that can be used by machine learning models)

Used TfidfVectorizer(max_features=5000)

Transformed cleaned text into a sparse matrix of shape (50000, 5000)

7. Model Development

1. Logistic Regression

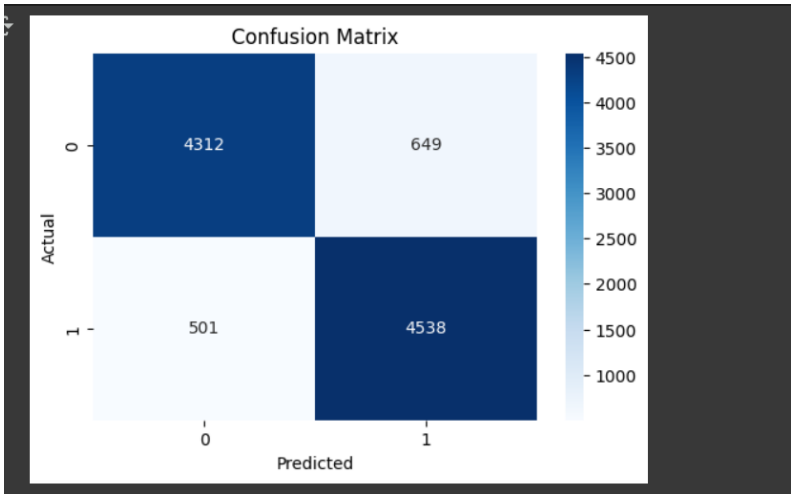
A linear model that estimates probabilities using the logistic function.
It's simple, fast, and effective for high-dimensional text data like TF-IDF features.

```
Accuracy: 0.885
Precision: 0.8748795064584538
Recall: 0.9005755110140901
F1 Score: 0.8875415607275572

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88      4961
           1       0.87      0.90      0.89      5039

    accuracy                           0.89     10000
   macro avg       0.89      0.88      0.88     10000
weighted avg       0.89      0.89      0.88     10000
```



2. Naive Bayes

A probabilistic classifier based on Bayes' theorem with a strong (naive) independence assumption.
It performs well on text classification due to its efficiency and simplicity, especially with sparse data

3. Support Vector Machine (SVM)

SVM finds the hyperplane that best separates classes in high-dimensional space.
It's highly accurate and effective in text classification tasks but can be computationally intensive.

4. Random Forest

An ensemble of decision trees that votes on the final classification.
It handles overfitting better than a single tree and can capture nonlinear patterns in the data.

evaluation metric

1. Accuracy

Accuracy measures the proportion of correctly predicted reviews (both positive and negative) out of all predictions.
It's useful when the dataset is balanced but can be misleading with imbalanced classes.

2. Precision

Precision tells us how many of the reviews predicted as positive were actually positive.
It's crucial when false positives are costly, ensuring reliable positive predictions

3. Recall

Recall measures how many of the actual positive reviews were correctly identified.
It's important when missing a positive review (false negative) is more critical than making a wrong one.

4. F1 Score

F1 Score is the harmonic mean of precision and recall, balancing both metrics.
It's especially useful when there's a trade-off between false positives and false negatives.

```
--- NAIVE_BAYES ---
Accuracy : 0.8483
Precision: 0.8442142298670836
Recall   : 0.8571145068465965
F1 Score : 0.8506154603643525

--- SVM ---
Accuracy : 0.8805
Precision: 0.8727695888285493
Recall   : 0.8930343322087716
F1 Score : 0.8827856792545365

--- RANDOM FOREST ---
Accuracy : 0.8458
Precision: 0.8548812664907651
Recall   : 0.8358801349474102
F1 Score : 0.8452739313666466
```

8 . Real-time Sentiment Prediction

```python
def predict_sentiment(review):
    cleaned = preprocess_text(review)
    vector = tfidf.transform([cleaned])
    prediction = model.predict(vector)[0]
    return "positive" if prediction == 1 else "negative"

review = input("Enter a movie review: ")
result = predict_sentiment(review)
print("Predicted Sentiment:", result)
```

```
Enter a movie review: it was good movie\
Predicted Sentiment: positive
```

Final Evaluation Summary

Among all the models tested, Logistic Regression achieved the best overall performance with an accuracy of 88.5%, a precision of 87.5%, and a recall of 90%, resulting in a high F1 score of 88.7%. These metrics indicate that the model is both precise and robust in identifying sentiments accurately across a balanced dataset.
Other models like SVM, Naive Bayes, and Random Forest also performed well, with SVM coming close to Logistic Regression. However, Logistic Regression provided the best balance between precision and recall, making it the most suitable model for sentiment classification in this project.

Video explaination

https://drive.google.com/file/d/1YqKniEj5JvG8lUXsav7OwjVkbVpqhPdq/view?usp=drive_link