🎬 Movie Review Sentiment Analysis – Report

Objective

This project aims to build and evaluate machine learning models for **sentiment analysis** on movie reviews, classifying them as **positive** (1) or **negative** (0). We used the IMDb dataset with labeled movie reviews to explore different classification techniques and identify the most effective model.

Approach

1. Dataset Overview

- IMDb Dataset with 50,000 labeled movie reviews.
- Balanced between 25,000 positive and 25,000 negative reviews.
- The dataset was split into training and testing sets (80:20).

2. Text Preprocessing

To prepare the text data for model training, we applied the following preprocessing steps without using **NLTK**:

- Lowercasing the text
- Removing HTML tags, numbers, punctuation
- Tokenization using regex and basic Python string methods
- Stopword removal using `sklearn`'s built-in English stopword list
- Vectorization using **TF-IDF** (Term Frequency-Inverse Document Frequency)

3. Models Used

Three machine learning models were implemented:

- **Logistic Regression**
- **Multinomial Naive Bayes**
- **Support Vector Machine (SVM)**

Each model was trained on the processed data and evaluated using accuracy, precision, recall, and F1-score.

Challenges Faced

- **NLTK Dependency Errors**: Due to environment constraints, NLTK was avoided. Alternatives like regex and `sklearn` stopword tools were used.
- **Text Cleaning Trade-offs**: Removing too much from the text could lose meaning; the challenge was to retain meaningful tokens while ensuring clean input for models.
- **Vector Representation**: Choosing between CountVectorizer and TF-IDF required testing for optimal performance. TF-IDF yielded better results overall.

Model Evaluation Results

```
Model: Logistic Regression
Accuracy: 0.8840
F1-Score: 0.8868
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88      4961
           1       0.87      0.90      0.89      5039

    accuracy                           0.88     10000
   macro avg       0.88      0.88      0.88     10000
weighted avg       0.88      0.88      0.88     10000


Model: Naive Bayes
Accuracy: 0.8477
F1-Score: 0.8492
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.84      0.85      4961
           1       0.85      0.85      0.85      5039

    accuracy                           0.85     10000
   macro avg       0.85      0.85      0.85     10000
weighted avg       0.85      0.85      0.85     10000


Model: SVM
Accuracy: 0.8726
F1-Score: 0.8751
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.86      0.87      4961
           1       0.86      0.89      0.88      5039

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
```

Conclusion

- **Best Performer**: **Logistic Regression** achieved the highest F1-score of **0.8868**, indicating balanced performance across precision and recall.
- **Naive Bayes** was the weakest in comparison, possibly due to its simplifying assumptions about word independence.
- **SVM** closely followed Logistic Regression, proving to be a robust alternative.

Future Improvements

- Try **deep learning models** like LSTM or BERT for richer language understanding.
- Deploy the model in a **Flask or Streamlit** app for interactive sentiment testing.