# Sentiment Analysis of Movie Reviews

## 1. Introduction

Sentiment analysis is a natural language processing task used to determine the emotional tone of textual data. This project analyzes movie reviews from the IMDB dataset to classify them as either positive or negative.

The primary objectives of this project are:

* To clean and preprocess textual data.
* To implement a machine learning model for sentiment classification.
* To evaluate the model's performance.
* To provide functionality for predicting the sentiment of new reviews.

## 2. Dataset

The dataset used in this project is the **IMDB Movie Review Dataset**. It contains two columns:

* **review**: The text of the review.
* **sentiment**: The sentiment of the review, labeled as either *positive* or *negative*.

**Dataset Summary:**

* Number of records: 50,000
* Sentiment Distribution:
  + Positive: ~25,000 reviews
  + Negative: ~25,000 reviews

## 3. Methodology

**3.1. Data Preprocessing**

Text data in its raw form is unstructured. Several preprocessing steps are applied to prepare the data for analysis:

1. **Remove HTML Tags:** Remove any HTML elements present in the text using regular expressions.
2. **Remove Special Characters:** Replace non-alphanumeric characters with spaces.
3. **Convert to Lowercase:** Standardize all text to lowercase for consistency.
4. **Remove Stopwords:** Remove common words like "the," "and," etc., which do not contribute to sentiment classification.
5. **Stemming:** Reduce words to their root forms using the Snowball Stemmer.

**3.2. Text Vectorization**

The preprocessed text is converted into a numerical format using the **CountVectorizer**, which creates a bag-of-words representation. The top 1000 features are selected based on frequency.

**3.3. Model Training**

The **Multinomial Naive Bayes (MNB)** algorithm was chosen for classification due to its effectiveness in text data analysis. The model was trained on 80% of the data and tested on the remaining 20%.

**3.4. Model Saving**

The trained model was saved as a serialized .pkl file for deployment and reuse.

4. Results

The model achieved an accuracy of ~85% on the test dataset.

## 4. Implementation

**4.1. Libraries Used**

* **Python Libraries:**
  + numpy, pandas: Data handling.
  + nltk: Text processing.
  + sklearn: Machine learning and evaluation.
  + pickle: Model serialization.
  + re: Regular expressions for cleaning text.

**4.2. Pipeline Overview**

1. **Data Loading:**

python

Copy code

data = pd.read\_csv('IMDB-Dataset.csv')

1. **Preprocessing:**
   * Sequential application of cleaning, removing special characters, stopword removal, and stemming.
2. **Feature Extraction:**
   * Vectorized text data into numerical features using CountVectorizer.
3. **Model Training:**

python

Copy code

mnb = MultinomialNB(alpha=1.0, fit\_prior=True)

mnb.fit(trainx, trainy)

1. **Evaluation:**

python

Copy code

accuracy = accuracy\_score(testy, ypm)

print("Model Accuracy:", accuracy)

## Results

**Model Accuracy:** ~85% on the test dataset.

The model effectively classifies reviews into positive and negative categories.

## ****7. Single Review Prediction****

A single review can be processed through the same cleaning and vectorization pipeline to predict its sentiment. Example code:

input\_vector = cv.transform([processed\_review]).toarray()

predicted\_sentiment = mnb.predict(input\_vector)

print("Predicted Sentiment:", predicted\_sentiment)

This functionality allows real-time sentiment analysis.

## ****8. Conclusion****

* The project demonstrates the implementation of a machine learning pipeline for sentiment analysis.
* The accuracy of ~85% highlights the effectiveness of the **Multinomial Naive Bayes** algorithm in text classification tasks.