IMDB Movie Review Sentiment Analysis

Project Overview

This project aims to perform **Sentiment Analysis** on a dataset of IMDB movie reviews. The goal is to classify the reviews as either **positive** or **negative** based on the text of the review. We use various **machine learning models** like Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) to analyze and classify these reviews.

Steps Involved

1. Data Collection

The dataset used in this project contains **movie reviews** and their respective **sentiments** (positive or negative). Each review is a string of text, and the sentiment is labeled as either **positive** or **negative**.

2. Data Preprocessing

In order to properly analyze the review text, we need to preprocess it. This step involves cleaning the text and preparing it for modeling:

- **Lowercasing**: Convert all text to lowercase so that words are treated consistently (e.g., "Good" and "good" should be the same).
- **Removing unwanted characters**: We remove HTML tags, URLs, and special characters to ensure the text is clean.
- **Tokenization**: Split the review into individual words (tokens).

- **Removing stopwords**: Stopwords (common words like "the", "a", "is") are removed since they do not contribute to the sentiment.
- **Stemming**: Words are reduced to their root form (e.g., "running" becomes "run").

This results in a clean, ready-to-use text that can be fed into machine learning models.

3. Exploratory Data Analysis (EDA)

In this step, we visualize and explore the dataset to get insights:

- **Word Count Distribution**: We check the distribution of the number of words in the reviews, both for positive and negative reviews. This helps us understand the size and structure of the reviews.
- **Most Frequent Words**: We also visualize the most frequently occurring words in positive and negative reviews using **word clouds**.

These visualizations help us understand the data better and give us clues about what words might be important for sentiment classification.

4. Feature Engineering

We need to convert the review text into a format that machine learning algorithms can understand:

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: This is a technique that converts the text data into numerical vectors. It reflects how important a word is in a given review while considering how frequently it appears in all reviews. Words that are more unique to a review will have higher importance.

5. Model Building

In this step, we train multiple machine learning models on the preprocessed text data:

- **Logistic Regression**: This is a simple but powerful model for binary classification problems like this one (positive vs negative).
- **Naive Bayes**: A probabilistic classifier based on Bayes' theorem, ideal for text data.
- **Support Vector Machine (SVM)**: A powerful classifier that works well for high-dimensional spaces like text data.

6. Model Evaluation

After training the models, we evaluate their performance:

- **Accuracy**: This measures how often the model correctly classifies reviews.
- **Confusion Matrix**: This shows the true positives, true negatives, false positives, and false negatives.
- **Classification Report**: This provides precision, recall, and F1-score metrics, helping us understand the performance of the models in more detail.