**Sentiment Analysis on IMDb Movie Reviews**

**Introduction**

This project aims to perform sentiment analysis on IMDb movie reviews. The dataset was obtained from Kaggle, which consists of 50,000 labeled movie reviews (25,000 for training and 25,000 for testing) that are marked as positive or negative. The objective is to train a machine learning model to predict the sentiment of a given movie review.

**Technologies Used**

* + Python
  + Jupyter Notebook
  + Pandas
  + NumPy
  + Natural Language Toolkit (NLTK)
  + Regular Expression (re)
  + Scikit-learn.
  + Matplotlib
  + Seaborn
  + WordCloud

**Data Preparation and Analysis:**

The first step in any data science project is to load, clean, and analyze the data. Here are the steps that were performed on the IMDb movie reviews dataset:

* + Loaded the dataset using pandas.
  + Inspected the dataset to get an understanding of the number of rows, columns, and feature names.
  + Checked for null values and duplicates and removed them.
  + Visualized the distribution of the positive and negative reviews using a bar chart to see if the dataset is balanced or imbalanced.
  + Calculated the percentage of each sentiment in the dataset.
  + Plotted a histogram to show the distribution of word counts in the movie reviews.
  + Created word clouds for positive and negative reviews separately to visualize the most frequent words in the reviews.

**Text Processing:**

Before training the machine learning model, the text data needs to be preprocessed to remove unwanted text, convert to lowercase, remove stopwords, and convert words to their base forms. Here are the steps that were performed:

* Created a function to remove special characters and HTML tags from the movie reviews using regular expressions.
* Applied the function to the text column of the dataset.
* Converted the text to lowercase and tokenized the text into words.
* Removed stopwords (common words such as "the," "a," "an," etc. that do not carry much meaning) using NLTK's stopword corpus.
* Converted words to their base forms using NLTK's PorterStemmer and WordNetLemmatizer.

**Training the Machine Learning Model**

The preprocessed text data is then converted into numerical vectors using the TfidfVectorizer. Four machine learning models were trained on the vectorized text data:

* Logistic Regression
* Naive Bayes
* Random Forest
* Support Vector Machine (SVM)

The accuracy of each model was evaluated using the accuracy score and classification report.

**Logistic Regression Model**

The logistic regression model is trained and tested using the TfidfVectorizer method, and the accuracy is calculated to be 0.89. The precision, recall, and F1-score are also calculated and shown in the classification report.

**Naive Bayes Model**

The naive Bayes model is trained and tested using the TfidfVectorizer method, and the accuracy is calculated to be 0.87. The precision, recall, and F1-score are also calculated and shown in the classification report.

**Random Forest Model**

The random forest model is trained and tested using the TfidfVectorizer method, and the accuracy is calculated to be 0.84. The precision, recall, and F1-score are also calculated and shown in the classification report.

**SVM Model**

The SVM model is trained and tested using the TfidfVectorizer method, and the accuracy is calculated to be 0.90. The precision, recall, and F1-score are also calculated and shown in the classification report.

**Conclusion**

Based on the analysis, the SVM model has the highest accuracy of 0.90, making it the most effective model for predicting the sentiment of a given movie review as positive or negative.