Text Classification

Problem Statement:

Developing a machine learning model for multi-class text classification of a dataset using various techniques such as Naive Bayes and Support Vector Machines (SVM). The dataset consists of textual content labelled with corresponding categories.

Approach:

Library Addition:

Addition of the required libraries such as numpy, pandas, nltk, etc.

Data Pre-processing:

Handling Empty Entries: Rows with missing or empty entries in the 'text' column were removed to ensure the quality of the dataset.

Text Lowercasing: To simplify analysis, all entries in the 'text' column were converted to lowercase.

Tokenization: Utilizing the NLTK library, sentences were tokenized into individual words

POS Tagging and Lemmatization: POS tagging was applied to each word, and lemmatization was performed using the WordNetLemmatizer from NLTK.

Stop-word Removal: Common English stopwords were removed from the tokenized words to focus on more meaningful content.

Data Splitting:

Split the dataset into training and testing sets using train_test_split() from scikit-learn.

Feature Extraction:

TF-IDF (Term Frequency-Inverse Document Frequency):

Used the TfidfVectorizer from scikit-learn to convert text data into numerical features.

Max features were limited to 5000 to control dimensionality.

CountVectorizer:

Used the CountVectorizer from scikit-learn to convert text data into numerical features.

Similar to above, max features in this were limited to 5000.

Model Training and Evaluation:

Used 2 models for the training with 2 different encoded data for both.

Naive Bayes:

Implemented Multinomial Naive Bayes models using encoded data by both TF-IDF and CountVectorizer and evaluated and reported accuracy scores on the testing dataset.

Support Vector Machines (SVM):

Implemented SVM models with linear kernels using encoded data by both TF-IDF and CountVectorizer, and evaluated and reported accuracy scores on the testing dataset.

Results:

The results were as follows:

Naive Bayes:

Accuracy Score (TF-IDF): 97.31%

Accuracy Score (CountVectorizer): 97.75%

Support Vector Machines (SVM):

Accuracy Score (TF-IDF): 98.35%

Accuracy Score (CountVectorizer): 97.01%

The text classification task achieved high accuracy using both Naive Bayes and SVM models. SVM demonstrated a superior accuracy using TF-IDF representations.