NAME:PRAPTI HALDER

Roll:M25CSA022

Github: https://github.com/Prapti1234/Sports_Politics_Classifier

Overview

Automatic text classification is an essential tool for handling large volumes of text data that emerge from online discussions, forums, and online communication platforms. With the increasing amount of text data, the process of manually categorizing documents into thematic groups is inefficient and infeasible. A comparison of various feature extraction techniques and classification models is done to identify the best strategy.

Data Source and Preparation

The dataset employed in this research is obtained from the 20 Newsgroups dataset, which is a popular benchmark dataset consisting of about 20,000 messages divided into 20 different groups. Unlike the news dataset, this dataset is comprised of actual user-submitted forum messages, making it more difficult and representative of real-world text classification tasks.The two categories related to sports were assigned the category label Sport, while the remaining three categories related to politics were assigned the category label Politics. This assignment of categories turned the original multi-class classification problem into a binary classification task.

To avoid any bias caused by the structured email metadata, the headers, footers, and quotes were removed from the data during the loading process.

```
Well over 100,000 in Lebanon alone.
1,000,000 - 2,000,000 in the Iran/Iraq conflict, even if Iranians
aren't Arabs, strictly speaking.  (They seem to hate the Zionists at
least as much as anyone else in the neighborhood.  Is there some
correlation perhaps between hating Israel and killing off your own
people?)
```

```
Average Document Length: 223.8124729320052
Minimum Document Length: 0
Maximum Document Length: 11251
```

Exploratory Data Analysis

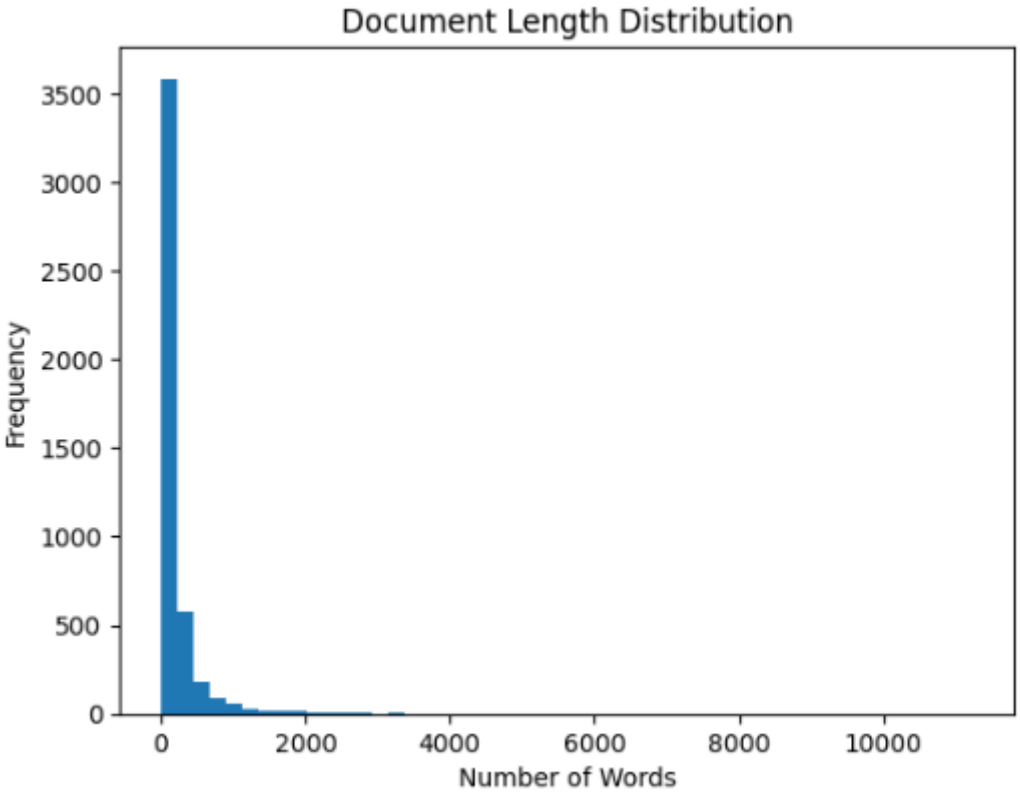
Before the model training process, the dataset was examined for its structural properties.

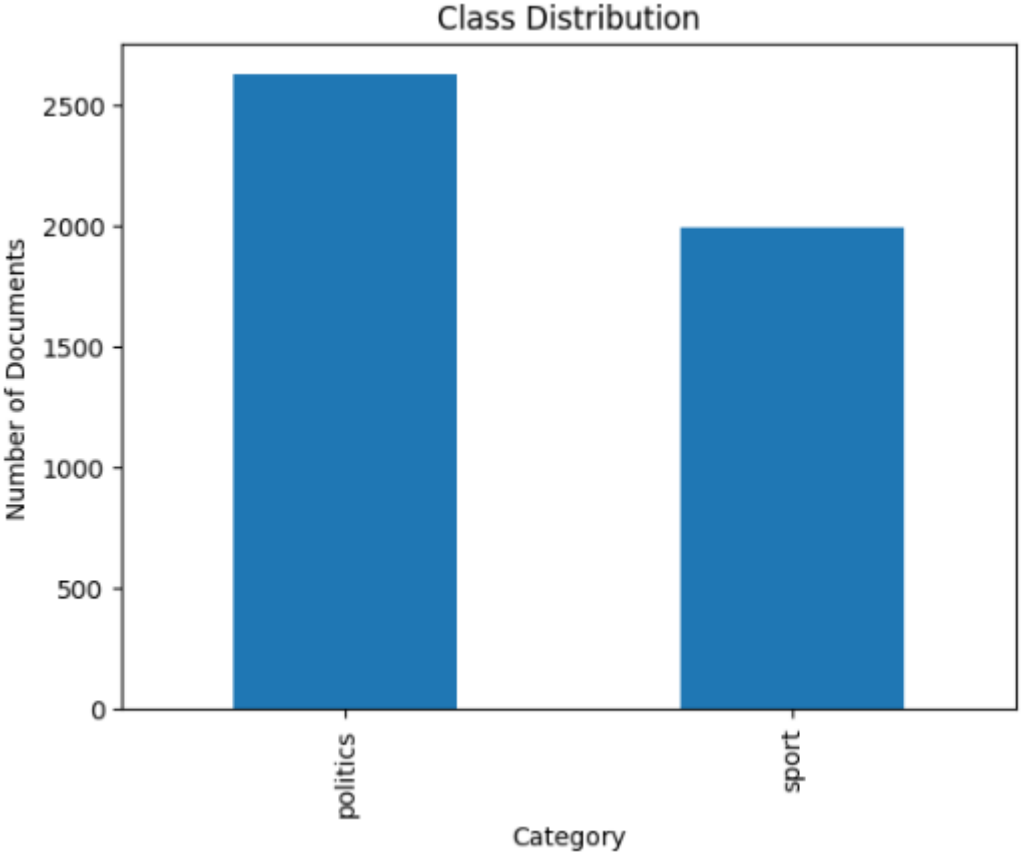
Text Length Features

The distribution of text lengths shows that most of the data points are between 30 and 45 words, with an average of 38 words. The fixed length shows that the dataset is composed of short summaries of news articles and not the full articles. The feature is very useful in that it is likely to reduce the complexity of computation and improve feature extraction.

Vocabulary Expansion with Different Representations

The different vectorization techniques resulted in different numbers of features. The unigram-based feature representations, such as Bag of Words and TF-IDF, resulted in the creation of approximately 41,000 features. The addition of bigram features substantially increased the number of features.


Document Length Distribution
Frequency
Number of Words
0
500
1000
1500
2000
2500
3000
3500
0
2000
4000
6000
8000
10000


Class Distribution
Number of Documents
Category
0
500
1000
1500
2000
2500
politics
sport

Text Representation Methods

The conversion of raw text data into numerical vectors is an important step in the application of machine learning to NLP.

Bag of Words

The Bag of Words is a simple text processing method where a document is converted into a vector of word frequencies. The method does not consider the relationships between words or the word order but is a simple method to represent the frequency of terms.

TF-IDF

The Term Frequency-Inverse Document Frequency method is an extension of the BoW method, where the weight of each term is calculated based on the relative importance of the term in the corpus. The terms that are more frequent in all documents are given less weight, and more important terms are given more weight.

N-gram Augmentation

To consider the relationships between words in context, bigram features were added to the system. This allows the system to identify important phrases rather than words. For example, phrases such as “prime minister” or “world cup” provide domain-specific information that cannot be extracted from single-word models.

Classification Algorithms

Three machine learning algorithms were coded.

Multinomial Naive Bayes

This Naive Bayes classifier is a kind of probabilistic classifier that uses Bayes’ theorem and the independence of features. Although the assumption of independence is quite naive, this classifier surprisingly performs well on text classification problems because of the sparse and high-dimensional vector representation of word vectors.

```
Naive Bayes Accuracy: 0.933982683982684

Classification Report:

              precision    recall  f1-score   support

    politics       0.90      1.00      0.94       525
       sport       0.99      0.85      0.92       399

    accuracy                           0.93       924
   macro avg       0.95      0.92      0.93       924
weighted avg       0.94      0.93      0.93       924
```

Linear Support Vector Machine

The SVM algorithm constructs a decision boundary that maximizes the margin between classes. In high-dimensional text feature spaces, linear SVMs are known to be extremely effective because they can efficiently exploit the sparse representation of feature vectors.

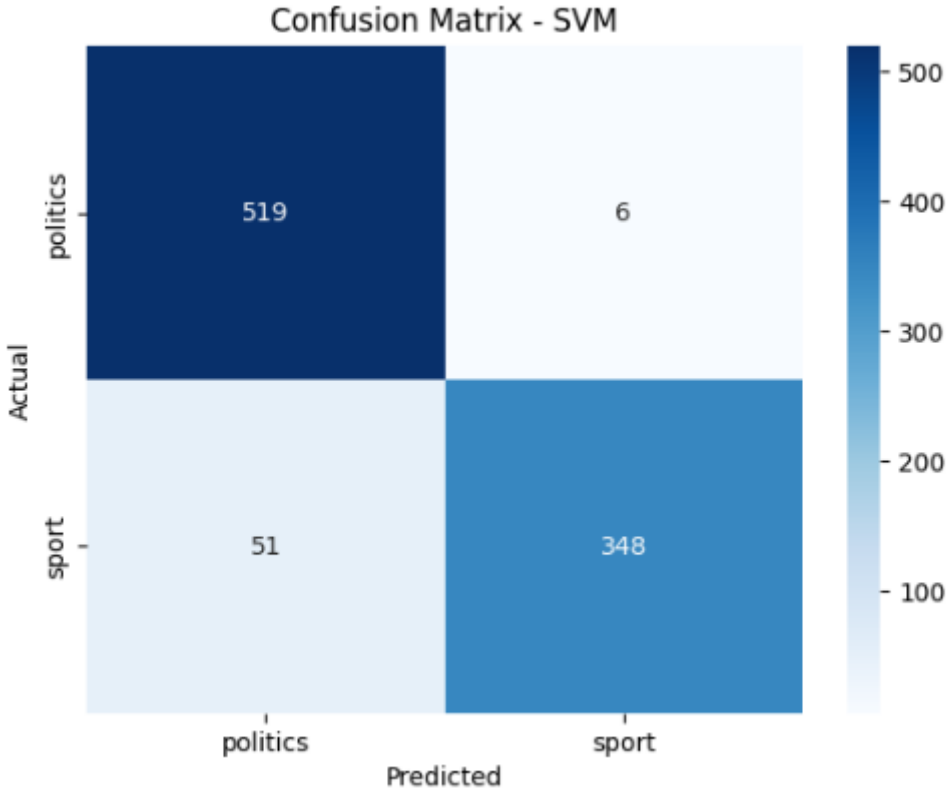
```
SVM Accuracy: 0.9383116883116883

Classification Report:

              precision    recall  f1-score   support

    politics       0.91      0.99      0.95       525
       sport       0.98      0.87      0.92       399

    accuracy                           0.94       924
   macro avg       0.95      0.93      0.94       924
weighted avg       0.94      0.94      0.94       924
```



Logistic Regression

```
Logistic Regression Accuracy: 0.9264069264069265

Classification Report:

              precision    recall  f1-score   support

    politics       0.89      1.00      0.94       525
       sport       0.99      0.83      0.91       399

    accuracy                           0.93       924
   macro avg       0.94      0.92      0.92       924
weighted avg       0.93      0.93      0.93       924
```

Performance Evaluation

The comparative analysis of the three machine learning algorithms, Multinomial Naive Bayes, Logistic Regression, and Linear Support Vector Machine, showed competitive results on all parameters.
The Linear SVM showed the highest overall accuracy of 93.83%, along with the highest precision of 94.18% and F1-score of 93.77%. This clearly shows that SVM generalizes slightly better on the 20 Newsgroups dataset than the other two algorithms.
Multinomial Naive Bayes showed an accuracy of 93.39%, which is very close to SVM. Its performance is not surprising, given its appropriateness for high-dimensional sparse text data. The assumption of independence may be very naive, but it still works well for text classification tasks.
Unlike the news headlines, which are structured and contain less overlapping vocabulary, the 20 Newsgroups dataset contains informal, conversational text with a lot of overlapping vocabulary between sports and politics.

	Model	Accuracy	Precision	Recall	F1-Score
0	Naive Bayes	0.933983	0.939876	0.933983	0.933143
1	Logistic Regression	0.926407	0.933755	0.926407	0.925312
2	SVM	0.938312	0.941844	0.938312	0.937736

Final Prediction System

In addition to model assessment, a comprehensive prediction pipeline has been developed to classify new, unseen documents. This development further verifies that the project is more than just a comparison of algorithms and is a fully functional end-to-end classification system. The prediction interface shows the feasibility of the trained model in real-world text classification problems.

Conclusion

This particular project has showcased the development of a binary text classification system, which aims to classify Sport and Politics documents based on certain categories of the 20 Newsgroups dataset. The dataset itself is quite complex and needed to be preprocessed by removing metadata and grouping categories for effective learning from text.Various methods of feature extraction, such as Bag of Words, TF-IDF, and n-grams, have been attempted, and three machine learning models have been compared. Of the models attempted, Linear Support Vector Machine performed the best and proved to be quite effective at dealing with high-dimensional sparse text data.

Even though the dataset is composed of informal discussion forum posts with a lot of shared vocabulary, the models performed quite well, proving that traditional machine learning models are still effective for document classification tasks.