Business Case Study: News Article Classification Using Natural Language Processing and Machine Learning Models

Problem Description

News article classification involves categorizing articles based on their content into predefined labels, such as politics, sports, technology, or finance. This task is significant for news agencies, search engines, and social media platforms that need to organize and recommend content efficiently.

The objective of the case study is to categorize a set of news articles into predefined categories such as politics, technology, sports, business, and entertainment, based on their content. The dataset contains articles with their respective categories, and the goal is to apply natural language processing (NLP) techniques to analyze the text data and build machine learning models for accurate classification. At least three different models will be created and compared to determine the most effective approach for categorizing the articles.

Methodology

1. **Exploratory Data Analysis**

   Import the dataset and check its shape to determine the number of rows (news articles) and columns (category) present. Then analyze the distribution of news articles across different categories by counting the number of articles in each category and visualizing this information using a bar chart.

2. **Data preprocessing**

   In the data preprocessing step, we begin by processing the textual data, which involves cleaning and preparing the news articles for analysis. This includes removing non-letter characters, tokenizing the text into individual words, eliminating common stopwords, and performing lemmatization to reduce words to their base forms.

   Next, we encode and transform the data. This involves encoding the target variable (the category of the news articles) into a numerical format and applying techniques like Bag of Words or TF-IDF to convert the textual data into a structured format suitable for machine learning models.

3. **Model Training**

   In the model training process, the processed dataset is split into training and testing sets, following a simple approach where a portion of the data is used to train the model, and the rest is reserved for evaluating its performance.

   The following machine learning algorithms are applied to classify the news article into different categories:

   Naive Bayes: A probabilistic model that works well with text data and assumes that features are independent of each other.

   Decision Tree: A tree-based algorithm that splits data into branches based on feature values, helping to classify the news articles.

   Nearest Neighbors: A distance-based algorithm that classifies articles based on the similarity of nearby points.

   Random Forest: An ensemble learning model that uses multiple decision trees to improve accuracy and reduce overfitting.

4. **Performance Evaluation**

   The performance of the trained models is evaluated using metrics like the confusion matrix, which provides a summary of correct and incorrect predictions across all classes, and the classification report, which details key performance indicators such as precision, recall, F1-score, and accuracy for each class, helping to understand how well the model performs in classifying the news articles.

Business Questions to be answered from Analysis

1. How many news articles are present in the dataset that we have?
2. Most of the news articles are from _____ category.
3. Only ___ no. of articles belong to the ‘Technology’ category.
4. What are Stop Words and why should they be removed from the text data?
5. Explain the difference between Stemming and Lemmatization.
6. Which of the techniques Bag of Words or TF-IDF is considered to be more efficient than the other?
7. What’s the shape of train & test data sets after performing a 75:25 split.
8. Which of the following is found to be the best performing model.

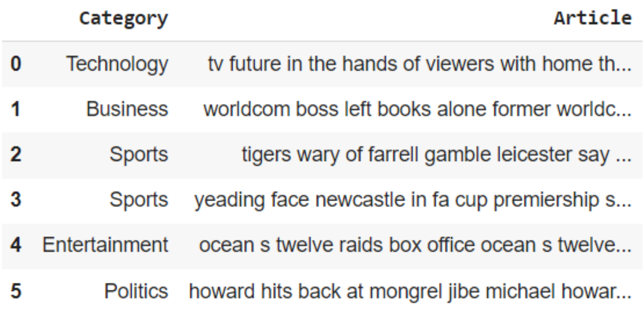
a. Random Forest b. Nearest Neighbors c. Naive Bayes

9. According to this particular use case, both precision and recall are equally important. (T/F)

Analysis

1. Exploratory Data Analysis

- **Importing the libraries & reading the data file:** The dataset consists of two columns, the first is the category that lists the category of the article given in the second column. The sample data frame is shown in Figure 5.1.



	Category	Article
0	Technology	tv future in the hands of viewers with home th...
1	Business	worldcom boss left books alone former worldc...
2	Sports	tigers wary of farrell gamble leicester say ...
3	Sports	yeading face newcastle in fa cup premiership s...
4	Entertainment	ocean s twelve raids box office ocean s twelve...
5	Politics	howard hits back at mongrel jibe michael howar...

Figure 5.1: Sample dataset

2. Exploring the dataset

- The dataset comprises 2225 rows and two columns as Category and Article. The categories in the dataset are Technology, Business, Sports, Entertainment, and Politics.
- The distribution of News articles per category is illustrated in Figure 5.2 and the percentage of each category of news article is exhibited in Figure 5.3.

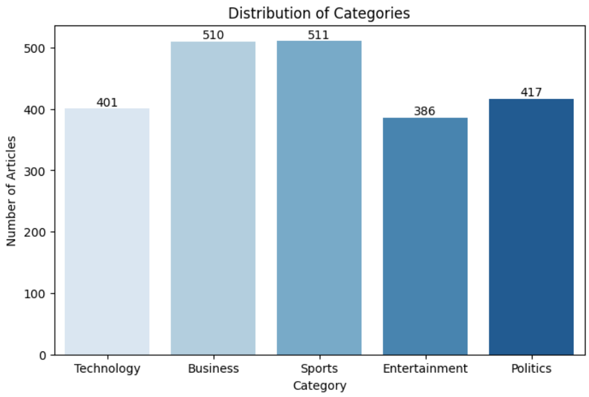


Figure 5.2: Distribution of News articles per category

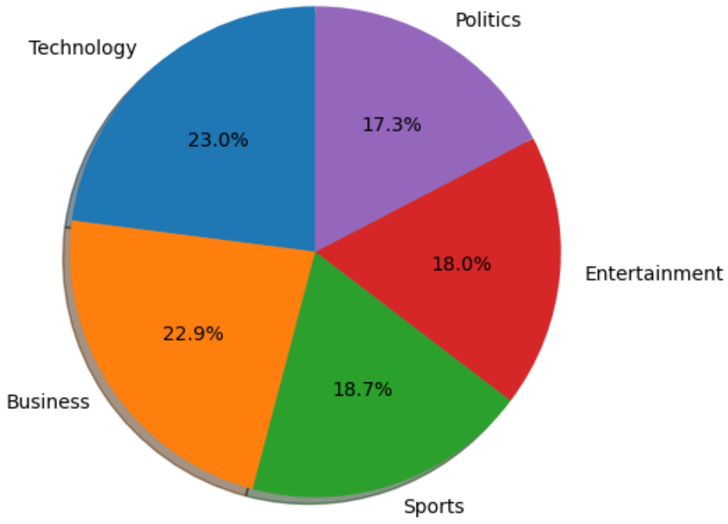
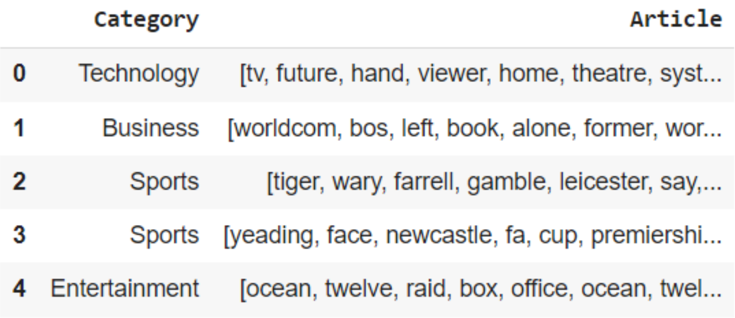


Figure 5.3: Pie chart showing the percent of each category of news article

3. Processing the Textual Data i.e. the news articles

- **Removing the non-letters:** Using regular expressions, remove any non-alphabetic characters from a given article column in the dataset, replacing them with spaces.
- **Tokenizing the text:** Tokenize the text into individual words using word_tokenize from nltk.
- **Removing stop words:**
  - Downloads the list of English stop words from the NLTK library
  - Filters these stop words from the 'Article' column in the dataset
  - The stop words are the common words like "the", "is", and "and" that don't contribute much to the meaning of the text.
- **Lemmatization:**
  - Downloads the WordNet lemmatizer from the NLTK library and applies it to the 'Article' column in the dataset, transforming each word into its base or root form (lemma) using a lambda function.
  - This process helps reduce words to their simplest forms, improving the consistency of the text data for further analysis.
  - Finally, it displays the first few rows of the updated dataset as shown in Figure 5.4.

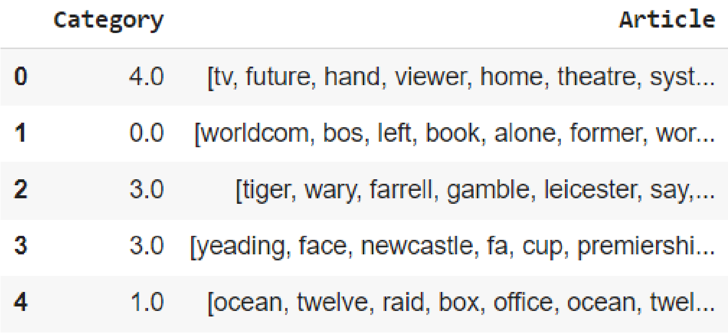


	Category	Article
0	Technology	[tv, future, hand, viewer, home, theatre, syst...
1	Business	[worldcom, bos, left, book, alone, former, wor...
2	Sports	[tiger, wary, farrell, gamble, leicester, say,...
3	Sports	[yeading, face, newcastle, fa, cup, premiershi...
4	Entertainment	[ocean, twelve, raid, box, office, ocean, twel...

Figure 5.4: Processed the Textual Data

4. Encoding and transforming the data

- **Encoding the target variable:** Ordinal encoding uses a single column of integers to represent the classes. Encode 'Category', the target variable column using the ordinal encoding technique and displayed in Figure 5.5.



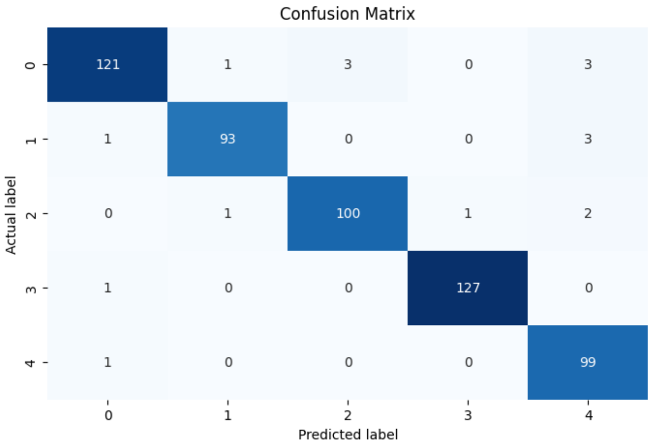
	Category	Article
0	4.0	[tv, future, hand, viewer, home, theatre, syst...
1	0.0	[worldcom, bos, left, book, alone, former, wor...
2	3.0	[tiger, wary, farrell, gamble, leicester, say,...
3	3.0	[yeading, face, newcastle, fa, cup, premiershi...
4	1.0	[ocean, twelve, raid, box, office, ocean, twel...

Figure 5.5: Encoded dataset

- **Bag of Words or TF-IDF:** Bag of Words (BoW) is a text representation technique used in natural language processing (NLP) that converts textual data into a numerical format for machine learning models. In BoW, each document is represented as a vector of word frequencies, disregarding grammar, word order, and context. Using the BoW technique, the textual data of the article column of the dataset are converted into the numerical format by extracting a maximum of 2000 features and support n-grams ranging from 1 to 3.
- **Train-Test Split:** The dataset is split in an 80: 20 ratio as a train set of 1668 rows and a test set of 557 rows.

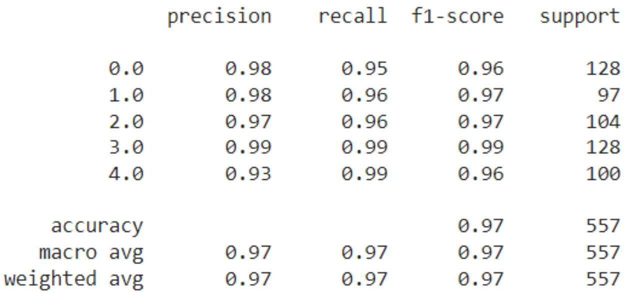
5. Model Training & Evaluation

- **Naive Bayes:** The dataset was trained and tested using the Naive Bayes model and got a training accuracy of 98.4 % and a test accuracy of 96.9%. The ROC AUC score, precision, recall, and F1 Score for this model are 0.999, 0.970, 0.969, and 0.970 respectively. The confusion matrix and classification report are shown in Figures 5.6 and 5.7.
- **Decision Tree:** The dataset was trained and tested using the Decision tree model and got a training accuracy of 100 % and a test accuracy of 83.8%. The ROC AUC score, precision, recall, and F1 Score for this model are 0.899, 0.841, 0.838, and 0.838 respectively. The confusion matrix and classification report are shown in Figures 5.8 and 5.9.



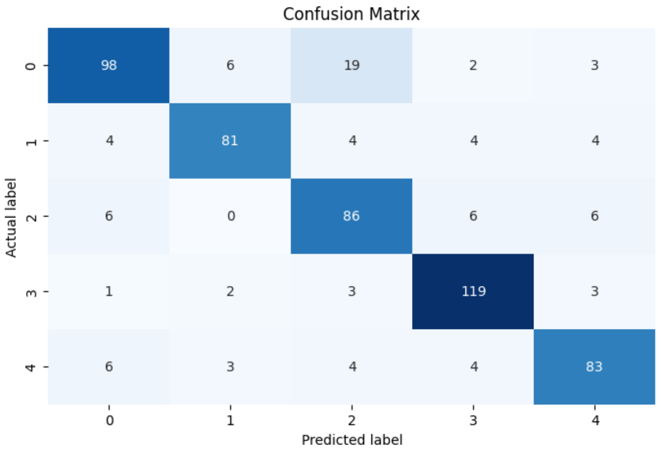
	0	1	2	3	4
0	121	1	3	0	3
1	1	93	0	0	3
2	0	1	100	1	2
3	1	0	0	127	0
4	1	0	0	0	99

Figure 5.6: Confusion matrix of the Naive Bayes model



```
              precision    recall  f1-score   support

         0.0       0.98      0.95      0.96       128
         1.0       0.98      0.96      0.97        97
         2.0       0.97      0.96      0.97       104
         3.0       0.99      0.99      0.99       128
         4.0       0.93      0.99      0.96       100

    accuracy                           0.97       557
   macro avg       0.97      0.97      0.97       557
weighted avg       0.97      0.97      0.97       557
```

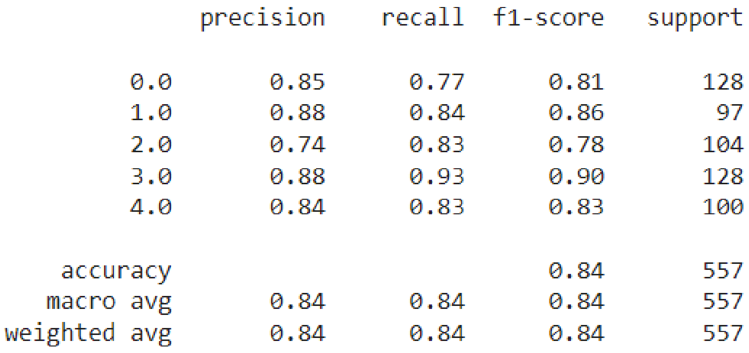
Figure 5.7: Classification report of the Naive Bayes model



	0	1	2	3	4
0	98	6	19	2	3
1	4	81	4	4	4
2	6	0	86	6	6
3	1	2	3	119	3
4	6	3	4	4	83

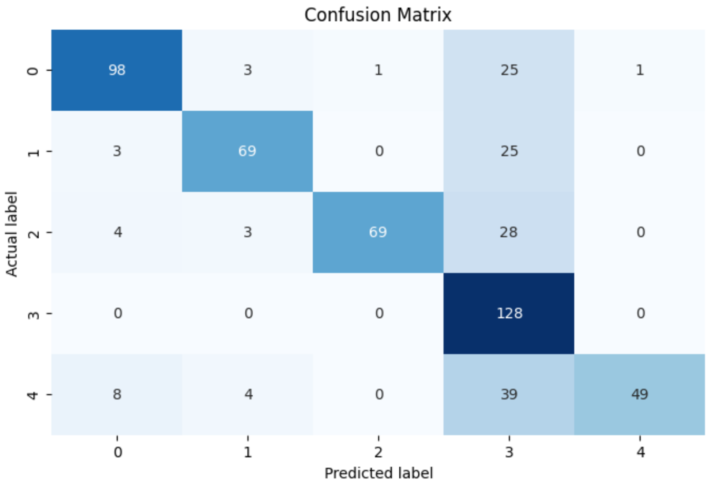
Figure 5.8: Confusion matrix of the Decision Tree model

- **Nearest Neighbors:** The dataset was trained and tested using the Nearest Neighbors model and got a training accuracy of 80.2 % and a test accuracy of 74.1%. The ROC AUC score, precision, recall, and F1 Score for this model are 0.940, 0.831, 0.741, and 0.747 respectively. The confusion matrix and classification report are shown in Figures 5.10 and 5.11.



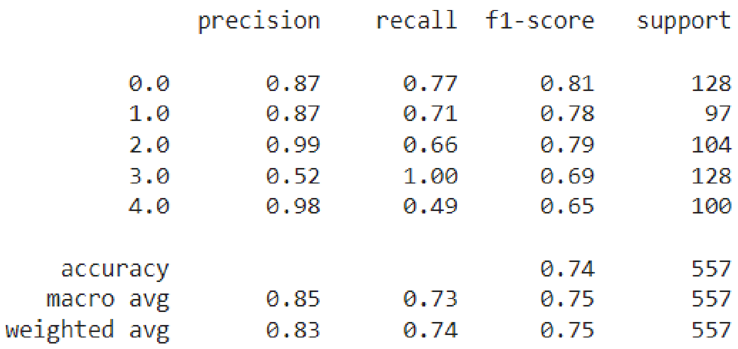
```
              precision    recall  f1-score   support

         0.0       0.85      0.77      0.81       128
         1.0       0.88      0.84      0.86        97
         2.0       0.74      0.83      0.78       104
         3.0       0.88      0.93      0.90       128
         4.0       0.84      0.83      0.83       100

    accuracy                           0.84       557
   macro avg       0.84      0.84      0.84       557
weighted avg       0.84      0.84      0.84       557
```

Figure 5.9: Classification report of the Decision Tree model



	0	1	2	3	4
0	98	3	1	25	1
1	3	69	0	25	0
2	4	3	69	28	0
3	0	0	0	128	0
4	8	4	0	39	49

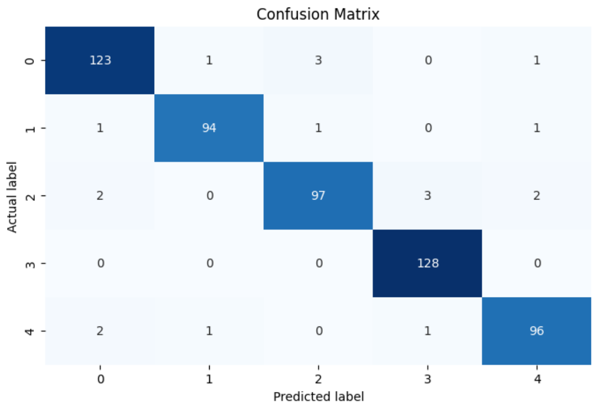
Figure 5.10: Confusion matrix of the Nearest Neighbors model



```
              precision    recall  f1-score   support

         0.0       0.87      0.77      0.81       128
         1.0       0.87      0.71      0.78        97
         2.0       0.99      0.66      0.79       104
         3.0       0.52      1.00      0.69       128
         4.0       0.98      0.49      0.65       100

    accuracy                           0.74       557
   macro avg       0.85      0.73      0.75       557
weighted avg       0.83      0.74      0.75       557
```

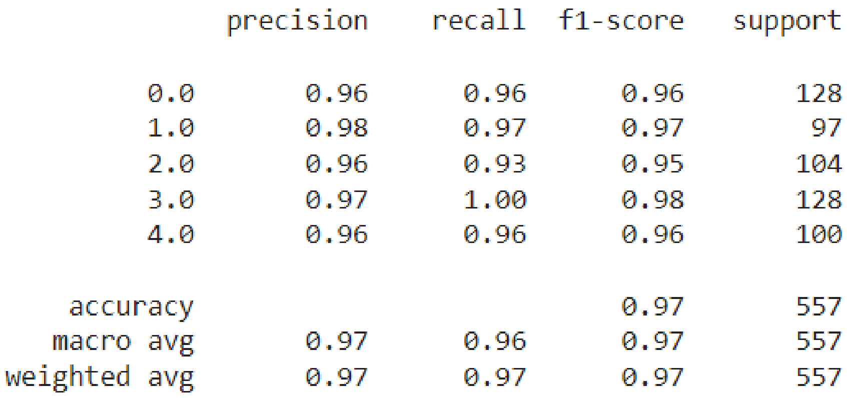
Figure 5.11: Classification report of the Nearest Neighbors model

- **Random Forest:** The dataset was trained and tested using the Random Forest model and got a training accuracy of 100 % and a test accuracy of 96.6 %. The ROC AUC score, precision, recall, and F1 Score for this model are 0.999, 0.966, 0.966, 0.966, and 0.966 respectively. The confusion matrix and classification report are shown in Figures 5.12 and 5.13.



	0	1	2	3	4
0	123	1	3	0	1
1	1	94	1	0	1
2	2	0	97	3	2
3	0	0	0	128	0
4	2	1	0	1	96

Figure 5.12: Confusion matrix of the Random Forest model



```
              precision    recall  f1-score   support

         0.0       0.96      0.96      0.96       128
         1.0       0.98      0.97      0.97        97
         2.0       0.96      0.93      0.95       104
         3.0       0.97      1.00      0.98       128
         4.0       0.96      0.96      0.96       100

    accuracy                           0.97       557
   macro avg       0.97      0.96      0.97       557
weighted avg       0.97      0.97      0.97       557
```

Figure 5.13: Classification report of the Random Forest model

Comparing all four methods, the random forest method has better parameters (train and test accuracy, ROC AUC score, precision, recall, F1 score). The training accuracy, test accuracy, ROC AUC score, precision, recall, and F1 Score for all four models are summarized in Table 5.1.

Table 5.1: Training accuracy, test accuracy, ROC AUC score, precision, recall, and F1 Score for all four models

Model	Training Accuracy	Test Accuracy	ROC AUC Score	Precision	Recall	F1 Score
Naive Bayes	0.984	0.959	0.999	0.97	0.959	0.97
Decision Tree	1	0.838	0.899	0.841	0.838	0.838
Nearest Neighbors	0.882	0.741	0.904	0.831	0.741	0.747
Random Forest	1	0.966	0.999	0.966	0.966	0.966

Insights and Recommendations

Actionable Insights

Recommendations

Business Questions to be answered from Analysis

1. How many news articles are present in the dataset that we have?
   2225

2. Most of the news articles are from the __category.
   Sports, Business, and Politics.
   The count of each category is given below:
   - Sports 511
   - Business 510
   - Politics 417
   - Technology 401
   - Entertainment 386

3. Only ____ no. of articles belong to the ‘Technology’ category

4. What are Stop Words and why should they be removed from the text data?
   Stop words are common words in a language that carry little meaning on their own in a text analysis context. Examples in English include "the", "is", "a", "an", "in", "on", "to", etc. These words are important for grammatical structure but don't hold much weight for understanding the core content of a text.The benefit of removing Stop Words from the text data is focuses on the content that truly matters, identifies patterns and relationships between the important words, and computes tasks become faster.

5. Explain the difference between Stemming and Lemmatization.
   Stemming and Lemmatization are both techniques used in Natural Language Processing (NLP) to normalize words into their base forms. However, they differ in their approach and outcome.
   - Stemming:
     Simpler and faster: Stemming employs a set of rules to chop off prefixes or suffixes from words, aiming to get to a root form.
     Less accurate: This process can be aggressive and sometimes lead to incorrect or unrecognizable words. For instance, stemming "running" might result in "run" which is valid, but stemming "agrees" could lead to "agre" which isn't a real word.
     Doesn't consider context: Stemming focuses solely on the word itself, ignoring its grammatical role (part of speech) in the sentence.
   - Lemmatization:
     More complex and slower: Lemmatization involves a deeper understanding of the language. It uses dictionaries and morphological analysis to map a word to its dictionary form, called a lemma.
     More accurate: By considering context and grammatical information, lemmatization ensures the output is a valid word in the language. For example, lemmatizing "running" would return "run" (the base verb form), and "agrees" would become "agree" (the base adjective form).
     Preserves meaning: Since lemmatization aims for the dictionary base form, it ensures the core meaning of the word is retained.
     Choosing between Stemming and Lemmatization:
     - If speed and simplicity are crucial, stemming can be a good initial approach.
     - If accuracy and preserving meaning are essential, lemmatization is the preferred choice, especially for tasks like sentiment analysis or topic modeling.

6. Which of the techniques Bag of Words or TF-IDF is considered to be more efficient than the other
   In terms of efficiency, Bag of Words (BoW) is generally considered to be faster and more lightweight compared to TF-IDF.
7. What’s the shape of train & test data sets after performing a 75:25 split?

The shape of the training dataset is (1668,2) and the test dataset is (557,2).

8. Which of the following is found to be the best-performing model.
   a. Random Forest b. Nearest Neighbors c. Naive Bayes
   a. Random Forest

9. According to this particular use case, both precision and recall are equally important. (T/F)
   T(True)