In [1]:
```
import os
```

In [4]:
```
os.chdir(r"D:")
```

In [14]:
```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load the BBC News dataset into a pandas dataframe
df = pd.read_csv('bbc.csv')
df= df.fillna("")

# Clean the text by removing stop words, punctuation marks, and converting all word
# You can use NLTK library for this purpose

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['ArticleId'], df['Category'

# Convert the text data into numerical features using CountVectorizer
X_test = [str(x) for x in X_test]

vectorizer = CountVectorizer()
X_train = [str(x) for x in X_train]


X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Train a Naive Bayes model using the training set
nb = MultinomialNB()
nb.fit(X_train, y_train)

# Evaluate the performance of the model using the testing set
y_pred = nb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Predict the category of new news articles
new_article = "Apple releases new iPhone"
new_article_vectorized = vectorizer.transform([new_article])
predicted_category = nb.predict(new_article_vectorized)
print("Predicted category:", predicted_category)
```

```
Accuracy: 0.21140939597315436
Predicted category: ['sport']
```

In [15]:
```
import pandas as pd

# Load the dataset into a Pandas DataFrame
df = pd.read_csv('bbc.csv')

# Print the first five rows of the DataFrame
print(df.head())
```

```
   ArticleId                                               Text  Category
0       1833  worldcom ex-boss launches defence lawyers defe...  business
1        154  german business confidence slides german busin...  business
2       1101  bbc poll indicates economic gloom citizens in ...  business
3       1976  lifestyle  governs mobile choice  faster  bett...      tech
4        917  enron bosses in $168m payout eighteen former e...  business
```

In [17]:
```
import nltk
from nltk.corpus import stopwords
import string

# Download the necessary stopwords and punctuation
nltk.download('stopwords')
nltk.download('punkt')
df['ArticleId'] = df['ArticleId'].astype(str)

# Remove stopwords and punctuation, and convert to lowercase
stop_words = set(stopwords.words('english'))
df['ArticleId'] = df['ArticleId'].apply(lambda x: ' '.join([word.lower() for word
```

```
[nltk_data] Downloading package stopwords to C:\Users\G BHAVANI
[nltk_data]     SHANKAR\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\G BHAVANI
[nltk_data]     SHANKAR\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [ ]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
df= pd.read_csv('bbc.csv')

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['ArticleId'], df['Category'

# Convert the text data into numerical features using CountVectorizer
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)''
```

In [20]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# Train a Naive Bayes model using the training set
nb = MultinomialNB()
nb.fit(X_train, y_train)

# Evaluate the performance of the model on the testing set
y_pred = nb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.21140939597315436
```

In [ ]: