Load Dataset - Loading dataset from downloaded dataset from Kaggle

In [90]:

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import nltk.corpus
from nltk.corpus import wordnet
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from spacy.cli import download
from spacy import load
from string import digits
import re
import itertools
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import NMF
from sklearn.metrics import accuracy_score
import sklearn.metrics as metrics
from sklearn.linear_model import LogisticRegression
```

In [91]:

```
os.chdir('/Users/evelynhaskins/Downloads/learn-ai-bbc')
```

Take a look into the dataset, get a feel for what it looks like

In [92]:

```
train_data = pd.read_csv('BBC News Train.csv')
test_data = pd.read_csv('BBC News Test.csv')
sample_solution = pd.read_csv('BBC News Sample Solution.csv')

print("Training Data Overview:")
print(train_data.info())
print(train_data.head())

print("\nTest Data Overview:")
print(test_data.info())
print(test_data.head())

print("\nSample Solution Overview:")
print(sample_solution.info())
print(sample_solution.head())
```

```
Training Data Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1490 entries, 0 to 1489
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ArticleId  1490 non-null   int64 
 1   Text       1490 non-null   object
 2   Category   1490 non-null   object
dtypes: int64(1), object(2)
memory usage: 35.0+ KB
None
   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

Test Data Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 735 entries, 0 to 734
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ArticleId  735 non-null    int64 
 1   Text       735 non-null    object
dtypes: int64(1), object(1)
memory usage: 11.6+ KB
None
   ArticleId                                               Text
0       1018  qpr keeper day heads for preston queens park r...
1       1319  software watching while you work software that...
2       1138  d arcy injury adds to ireland woe gordon d arc...
3        459  india s reliance family feud heats up the ongo...
4       1020  boro suffer morrison injury blow middlesbrough...

Sample Solution Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 735 entries, 0 to 734
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ArticleId  735 non-null    int64 
 1   Category   735 non-null    object
dtypes: int64(1), object(1)
memory usage: 11.6+ KB
None
   ArticleId       Category
0       1018          sport
1       1319           tech
2       1138       business
3        459  entertainment
4       1020       politics
```

Removing noise from articles "Text" column

Adding category ID mapped to a specific number

In [93]:

```
category_mapping = {category: idx for idx, category in enumerate(train_data[
train_data['CategoryId'] = train_data['Category'].map(category_mapping)

category = pd.DataFrame(list(category_mapping.items()), columns=['Category',

print("Unique categories with IDs:")
print(category)

print("Updated train_data with CategoryId:")
print(train_data[['ArticleId', 'Category', 'CategoryId']].head())
```

```
Unique categories with IDs:
        Category  CategoryId
0       business           0
1           tech           1
2       politics           2
3          sport           3
4  entertainment           4
Updated train_data with CategoryId:
   ArticleId  Category  CategoryId
0       1833  business           0
1        154  business           0
2       1101  business           0
3       1976      tech           1
4        917  business           0
```

In [94]:

```
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet

nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('averaged_perceptron_tagger')

# Convert NLTK POS tags to WordNet POS tags
def get_wordnet_pos(word):
    tag = nltk.pos_tag([word])[0][1][0].upper()
    tag_dict = {
        'J': wordnet.ADJ,
        'N': wordnet.NOUN,
        'V': wordnet.VERB,
        'R': wordnet.ADV
    }
    return tag_dict.get(tag, wordnet.NOUN)

def clean_text(dataframe, text_col):
    # Handle missing values
    dataframe[text_col] = dataframe[text_col].fillna('')
    # Convert to lowercase
    dataframe['lower_text'] = dataframe[text_col].str.lower()

    # Remove punctuation
    dataframe['no_punct'] = dataframe['lower_text'].apply(
        lambda row: re.sub(r'[^\w\s]+', '', row))

    # Remove numbers
    dataframe['no_punct_num'] = dataframe['no_punct'].apply(
        lambda row: re.sub(r'[0-9]+', '', row))

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    dataframe['no_stopwords'] = dataframe['no_punct_num'].apply(
        lambda x: ' '.join([word for word in x.split() if word not in stop_w

    # Lemmatize words
    lemmatizer = WordNetLemmatizer()
    dataframe['lemmatized_text'] = dataframe['no_stopwords'].apply(
        lambda x: ' '.join(
            [lemmatizer.lemmatize(word, get_wordnet_pos(word)) for word in x
        )
    )

    # Remove extra spaces
    dataframe['clean_text'] = dataframe['lemmatized_text'].apply(
        lambda x: re.sub(r'\s+', ' ', x).strip())

    return dataframe


train_data = clean_text(train_data, 'Text')

print(train_data[['Text', 'clean_text']].head())
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/evelynhaskins/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/evelynhaskins/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     /Users/evelynhaskins/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/evelynhaskins/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

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

                                          clean_text
0  worldcom exboss launch defence lawyer defend f...
1  german business confidence slide german busine...
2  bbc poll indicates economic gloom citizen majo...
3  lifestyle governs mobile choice faster well fu...
4  enron boss payout eighteen former enron direct...
```

In [95]:

```
x = train_data['Text']
y = train_data['CategoryId']
```

Mapping frequently used words to 1 and the rest to 0

In [111…

```
from sklearn.feature_extraction.text import CountVectorizer
x = np.array(train_data.iloc[:,0].values)
y = np.array(train_data.CategoryId.values)
cv = CountVectorizer(max_features = 5000)
x = cv.fit_transform(train_data.Text).toarray()
print("X.shape = ",x.shape)
print("y.shape = ",y.shape)
```

```
X.shape =  (1490, 5000)
y.shape =  (1490,)
```

Out[111…

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0]])
```

In [97]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, r
print(len(x_train))
print(len(x_test))
```

```
1043
447
```

Fitting Supervised Models

In [98]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

svm_model = SVC(kernel='linear', random_state=0)
svm_model.fit(x_train, y_train)

y_pred = svm_model.predict(x_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')  # Weighted
recall = recall_score(y_test, y_pred, average='weighted')
f1score = f1_score(y_test, y_pred, average='weighted')

print('Support Vector Machine:')
print(('Test Accuracy', round(accuracy, 2)))
print(('Precision', round(precision, 2)))
print(('Recall', round(recall, 2)))
print(('F1', round(f1score, 2)))
```

```
Support Vector Machine:
('Test Accuracy', 0.96)
('Precision', np.float64(0.96))
('Recall', np.float64(0.96))
('F1', np.float64(0.96))
```

In [99]:

```
from sklearn.linear_model import LogisticRegression

logreg_model = LogisticRegression(random_state=0, max_iter=1000)
logreg_model.fit(x_train, y_train)

y_pred = logreg_model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')  # Weighted
recall = recall_score(y_test, y_pred, average='weighted')
f1score = f1_score(y_test, y_pred, average='weighted')

print(('Test Accuracy', round(accuracy, 2)))
print(('Precision', round(precision, 2)))
print(('Recall', round(recall, 2)))
print(('F1', round(f1score, 2)))
```

```
('Test Accuracy', 0.96)
('Precision', np.float64(0.96))
('Recall', np.float64(0.96))
('F1', np.float64(0.96))
```

In [ ]:

```
joblib.dump(logreg_model, 'logistic_regression_model.pkl')
```

In [100…

```
from sklearn.tree import DecisionTreeClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=0)
rf_model.fit(x_train, y_train)

y_pred = rf_model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1score = f1_score(y_test, y_pred, average='weighted')

print(('Test Accuracy', round(accuracy, 2)))
print(('Precision', round(precision, 2)))
print(('Recall', round(recall, 2)))
print(('F1', round(f1score, 2)))
```

```
('Test Accuracy', 0.95)
('Precision', np.float64(0.95))
('Recall', np.float64(0.95))
('F1', np.float64(0.95))
```

In [124…

```
from sklearn.tree import DecisionTreeClassifier

dt_model = DecisionTreeClassifier(random_state=0)
dt_model.fit(x_train, y_train)

y_pred = dt_model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1score = f1_score(y_test, y_pred, average='weighted')

print(('Test Accuracy', round(accuracy, 2)))
print(('Precision', round(precision, 2)))
print(('Recall', round(recall, 2)))
print(('F1', round(f1score, 2)))
```

```
('Test Accuracy', 0.82)
('Precision', np.float64(0.82))
('Recall', np.float64(0.82))
('F1', np.float64(0.82))
```

In [102…

```
from sklearn.neighbors import KNeighborsClassifier

knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(x_train, y_train)

y_pred = knn_model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1score = f1_score(y_test, y_pred, average='weighted')

print(('Test Accuracy', round(accuracy, 2)))
print(('Precision', round(precision, 2)))
print(('Recall', round(recall, 2)))
print(('F1', round(f1score, 2)))
```

```
('Test Accuracy', 0.72)
('Precision', np.float64(0.74))
('Recall', np.float64(0.72))
('F1', np.float64(0.72))
```

Comparing Model Accuracy

In [103…

```
import pandas as pd
from sklearn.metrics import accuracy_score, precision_score, recall_score, f
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier

perform_list = []

models = {
    "SVM": SVC(),
    "Logistic Regression": LogisticRegression(),
    "Random Forest": RandomForestClassifier(),
    "Decision Tree": DecisionTreeClassifier(),
    "KNN": KNeighborsClassifier(n_neighbors=3)
}

for model_name, model in models.items():
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)

    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1score = f1_score(y_test, y_pred, average='weighted')

    perform_list.append([model_name, round(accuracy, 2), round(precision, 2)

model_performance = pd.DataFrame(data=perform_list, columns=['Model', 'Test

model_performance = model_performance[['Model', 'Test Accuracy', 'Precision'

print(model_performance)
```

```
/Users/evelynhaskins/.pyenv/versions/3.10.12/lib/python3.10/site-packages/sk
learn/linear_model/_logistic.py:469: ConvergenceWarning: lbfgs failed to con
verge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regre
ssion
  n_iter_i = _check_optimize_result(
```

```
                 Model  Test Accuracy  Precision  Recall    F1
0                  SVM           0.92       0.92    0.92  0.92
1  Logistic Regression           0.97       0.97    0.97  0.97
2        Random Forest           0.95       0.95    0.95  0.95
3        Decision Tree           0.82       0.82    0.82  0.82
4                  KNN           0.72       0.74    0.72  0.72
```

Testing model on new articles that have no predefined category

Cleaning test data - do I have to do this? Is this more effecient?

```
In [112…
test_dataset = clean_text(test_data, 'Text')

print(test_dataset[['Text', 'clean_text']].head())
```

```
                                                Text  \
0  qpr keeper day heads for preston queens park r...
1  software watching while you work software that...
2  d arcy injury adds to ireland woe gordon d arc...
3  india s reliance family feud heats up the ongo...
4  boro suffer morrison injury blow middlesbrough...

                                          clean_text
0  qpr keeper day head preston queen park ranger ...
1  software watch work software monitor every key...
2  arcy injury add ireland woe gordon arcy rule i...
3  india reliance family feud heat ongoing public...
4  boro suffer morrison injury blow middlesbrough...
```

Shifting it to the 0 and 1 format for word fequency

```
In [117…
from sklearn.feature_extraction.text import CountVectorizer
x = np.array(test_dataset.iloc[:,0].values)
cv = CountVectorizer(max_features = 5000)
x = cv.fit_transform(test_dataset.Text).toarray()
print("X.shape = ",x.shape)
x
```

```
X.shape =  (735, 5000)
```

```
Out[117…
 array([[0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        ...,
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]])
```

Lets see if it works!

```
In [123…
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
import joblib

logreg_model = joblib.load('logistic_regression_model.pkl')
predictions = logreg_model.predict(x)
test_dataset['predicted_category'] = predictions
print(test_dataset.iloc[200:221][['Text', 'predicted_category']])
```

```
                                                  Text  predicted_category
200  campbell lifts lid on united feud arsenal s so...                   3
201  brown visits slum on africa trip chancellor go...                   3
202  industrial output falls in japan japanese indu...                   3
203  cult band kasabian surge forward indie dance b...                   3
204  high fuel prices hit ba s profits british airw...                   3
205  turkey knocks six zeros off lira turkey is to ...                   3
206  arsenal through on penalties arsenal win 4-2 o...                   3
207  playstation 3 chip to be unveiled details of t...                   0
208  ba to suspend two saudi services british airwa...                   3
209  anelka apologises for criticism manchester cit...                   3
210  patti smith to host arts festival rock star pa...                   3
211  church urges nelly show boycott church ministe...                   3
212  beattie return calms attack fears everton stri...                   3
213  bookmakers back aviator for oscar the aviator ...                   3
214  radcliffe eyes hard line on drugs paula radcli...                   3
215  what price for  trusted pc security   you can ...                   0
216  minimum rate for foster parents foster carers ...                   3
217  bening makes awards breakthrough film actress ...                   4
218  sculthorpe wants lions captaincy paul sculthor...                   3
219  fry set for role in hitchhiker s actor stephen...                   3
220  murray returns to scotland fold euan murray ha...                   3
```

```
In [ ]:
```