DATA HANDLING

In [1]:

```
import pandas as pd
```

In [2]:

```
# importing true csv file
df1 = pd.read_csv('true.csv')
```

In [3]:

```
# import fake csv file
df2 = pd.read_csv('fake.csv')
```

In [4]:

```
# data exploration of df1
df1.head()
```

Out[4]:

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

In [5]:

```
# data exploration of df2
df2.head()
```

Out[5]:

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

In [6]:

```
# create another column 'target' set to 1 for true news
df1['target'] = 1
```

In [7]:

```
df1.head()
```

Out[7]:

	title	text	subject	date	target
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1

In [8]:

```
# create another column 'target' set to 0 for fake news
df2['target'] = 0
```

In [9]:

```
df2.head()
```

Out[9]:

	title	text	subject	date	target
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0

In [10]:

```
df1.shape
```

Out[10]:

```
(21417, 5)
```

In [11]:

```
df2.shape
```

Out[11]:

```
(23481, 5)
```

In [12]:

```
# merge the two dataframes as one
merged_df = pd.concat([df1, df2], ignore_index = False)
```

In [13]:

```
merged_df.head()
```

Out[13]:

	title	text	subject	date	target
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1

In [14]:

```
merged_df.shape
```

Out[14]:

```
(44898, 5)
```

In [15]:

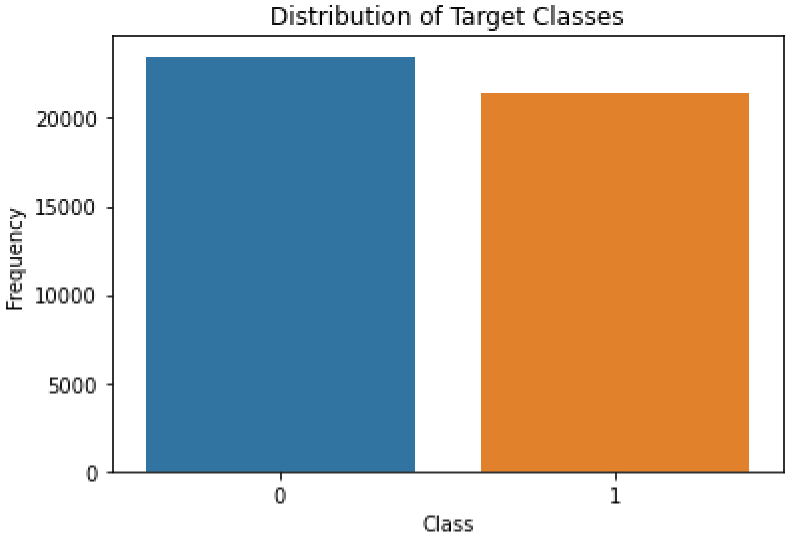
```
# splitting the dataframe to features and target
X = merged_df.iloc[:, 1].values
y = merged_df.iloc[:, -1].values
```

PLOTTING DATA DISTRIBUTION

In [16]:

```
import matplotlib.pyplot as plt
import seaborn as sns

target = y

# Create a count plot
sns.countplot(target)

# Beautify the plot
plt.title('Distribution of Target Classes')
plt.xlabel('Class')
plt.ylabel('Frequency')

# Show the plot
plt.show()
```

```
/Users/nishandhillon/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decor
ators.py:36: FutureWarning: Pass the following variable as a keyword arg: x.
From version 0.12, the only valid positional argument will be `data`, and pas
sing other arguments without an explicit keyword will result in an error or m
isinterpretation.
  warnings.warn(
```



SPLITTING THE DATA INTO TRAINING AND TESTING

In [17]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, trai
```

In [18]:

```
X_train.shape
```

Out[18]:

```
(35918,)
```

APPLYING TFIDF VECTORIZATION

In [19]:

```
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

stopwords = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stopwords)
```

In [20]:

```
# apply tfidf vectorizer
X_train = vectorizer.fit_transform(X_train)  # fit and transform the train da
X_test = vectorizer.transform(X_test)        # transform only the test data
```

NAIVE BAYES

In [21]:

```
from sklearn.naive_bayes import MultinomialNB

naive_bayes = MultinomialNB()
naive_bayes.fit(X_train, y_train)
```

Out[21]:

```
MultinomialNB()
```

In [22]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1
```

```
# make predictions on the test data
y_pred = naive_bayes.predict(X_test)

# print confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
[[4427  269]
 [ 325 3959]]
```

NAIVE BAYES - MODEL ANALYSIS

In [23]:

```
print('accuracy score: ', accuracy_score(y_test, y_pred))

print('\nprecision score (not fake): ', precision_score(y_test, y_pred, pos_l
print('precision score (fake): ', precision_score(y_test, y_pred))

print('\nrecall score: (not fake)', recall_score(y_test, y_pred, pos_label=0)
print('recall score: (fake)', recall_score(y_test, y_pred))

print('\nf1 score: ', f1_score(y_test, y_pred))
```

```
accuracy score:  0.9338530066815145

precision score (not fake):  0.9316077441077442
precision score (fake):  0.9363765373699149

recall score: (not fake) 0.942717206132879
recall score: (fake) 0.9241363211951448

f1 score:  0.9302161654135339
```

In [24]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.93      0.94      0.94      4696
           1       0.94      0.92      0.93      4284

    accuracy                           0.93      8980
   macro avg       0.93      0.93      0.93      8980
weighted avg       0.93      0.93      0.93      8980
```

In [25]:

```
print('spam size in test data:',y_test[y_test==0].shape[0])
print('test size: ', len(y_test))
baseline = y_test[y_test==0].shape[0] / y_test.shape[0]
print(baseline)
```

```
spam size in test data: 4696
test size:  8980
0.5229398663697105
```

LOGISTIC REGRESSION

In [26]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

Out[26]:
```
LogisticRegression(random_state=0)
```

In [27]:
```
y_pred = classifier.predict(X_test)
```

LOGISTIC REGRESSION - MODEL ANALYSIS

In [28]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[4625   71]
 [  43 4241]]
```

Out[28]:
```
0.987305122494432
```

In [29]:
```
print('accuracy score: ', accuracy_score(y_test, y_pred))

print('\nprecision score (not fake): ', precision_score(y_test, y_pred, pos_l
print('precision score (fake): ', precision_score(y_test, y_pred))

print('\nrecall score: (not fake)', recall_score(y_test, y_pred, pos_label=0)
print('recall score: (fake)', recall_score(y_test, y_pred))

print('\nf1 score: ', f1_score(y_test, y_pred))
```

```
accuracy score:  0.987305122494432

precision score (not fake):  0.9907883461868038
precision score (fake):  0.983534322820037

recall score: (not fake) 0.9848807495741057
recall score: (fake) 0.9899626517273576

f1 score:  0.9867380176826431
```

NEURAL NETWORKS

In [30]:
```
from sklearn.neural_network import MLPClassifier
```

In [31]:
```
# Create an instance of the MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(10, 10, 10), max_iter=1000, random_st
```

In [32]:
```
# Train the model
mlp.fit(X_train, y_train)
```

Out[32]:
```
MLPClassifier(hidden_layer_sizes=(10, 10, 10), max_iter=1000, random_state=4
2)
```

In [33]:
```
# Predictions
y_pred = mlp.predict(X_test)
```

NEURAL NETWORKS - MODEL ANALYSIS

In [34]:

```
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[4651   45]
 [  49 4235]]
```

Out[34]:

```
0.989532293986637
```

In [35]:

```
print('accuracy score: ', accuracy_score(y_test, y_pred))

print('\nprecision score (not fake): ', precision_score(y_test, y_pred, pos_l
print('precision score (fake): ', precision_score(y_test, y_pred))

print('\nrecall score: (not fake)', recall_score(y_test, y_pred, pos_label=0)
print('recall score: (fake)', recall_score(y_test, y_pred))

print('\nf1 score: ', f1_score(y_test, y_pred))
```

```
accuracy score:  0.989532293986637

precision score (not fake):  0.9895744680851064
precision score (fake):  0.9894859813084113

recall score: (not fake) 0.9904173764906303
recall score: (fake) 0.988562091503268

f1 score:  0.9890238206445587
```

ANALYSIS OF PERFORMANCE OF DIFFERENT APPROACHES

To analyze the performance of three different approaches to classification—Naive Bayes, Logistic Regression, and Neural Networks—based on the provided metrics, it's important to consider what each metric signifies and how it relates to the overall performance of the model. The metrics given are accuracy, precision (for both "not fake" and "fake" classes), recall (for both "not fake" and "fake" classes), and the F1 score.

Accuracy

Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. It gives a quick snapshot of the model's overall performance but doesn't account for the balance between classes.

Naive Bayes: 93.39%

Logistic Regression: 98.73%

Neural Networks: 98.95%

Analysis: Neural Networks achieve the highest accuracy, closely followed by Logistic Regression, indicating a superior overall classification performance. Naive Bayes lags behind, which is typical for data with complex patterns that Naive Bayes can't capture due to its assumption of feature independence.

Precision

Precision measures the accuracy of the positive predictions (i.e., the percentage of predicted positives that are actually positive).

Naive Bayes: 93.16% (not fake), 93.64% (fake)

Logistic Regression: 99.08% (not fake), 98.35% (fake)

Neural Networks: 98.96% (not fake), 98.95% (fake)

Analysis: Logistic Regression shows slightly better precision for the "not fake" class than Neural Networks, indicating it's more reliable when identifying non-fake instances. However, both models show high precision, suggesting few false positives are made.

Recall

Recall measures the ability of the model to find all the actual positives (i.e., the percentage of actual positives that were correctly identified).

Naive Bayes: 94.27% (not fake), 92.41% (fake)

Logistic Regression: 98.49% (not fake), 98.99% (fake)

Neural Networks: 99.04% (not fake), 98.86% (fake)

Analysis: Neural Networks demonstrate the highest recall for the "not fake" class, indicating they are the most capable of identifying all non-fake instances. Logistic Regression shows a slightly better recall for the "fake" class, suggesting a marginal advantage in detecting fake instances.

F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics for situations where an imbalance might exist.

Naive Bayes: 93.02%

Logistic Regression: 98.67%

Neural Networks: 98.90%

Analysis: The F1 scores corroborate the previous findings, with Neural Networks showing the best balance between precision and recall, closely followed by Logistic Regression. Naive Bayes, while respectable, shows a lower ability to balance these metrics effectively.

Conclusion

Neural Networks emerge as the most robust model among the three, showing superior performance across almost all metrics. This is likely due to their ability to capture complex, non-linear relationships in the data. Logistic Regression also performs admirably, outpacing Naive Bayes significantly and coming close to Neural Networks, especially in precision and recall for the "fake" class.

Naive Bayes, despite its simplicity and the assumption of independence among features, offers decent performance, particularly valuable when computational simplicity and speed are important. However, for the highest accuracy, especially in complex tasks such as this likely classification problem, Neural Networks or Logistic Regression are preferable, with Neural Networks having a slight edge in overall performance.

In [ ]: