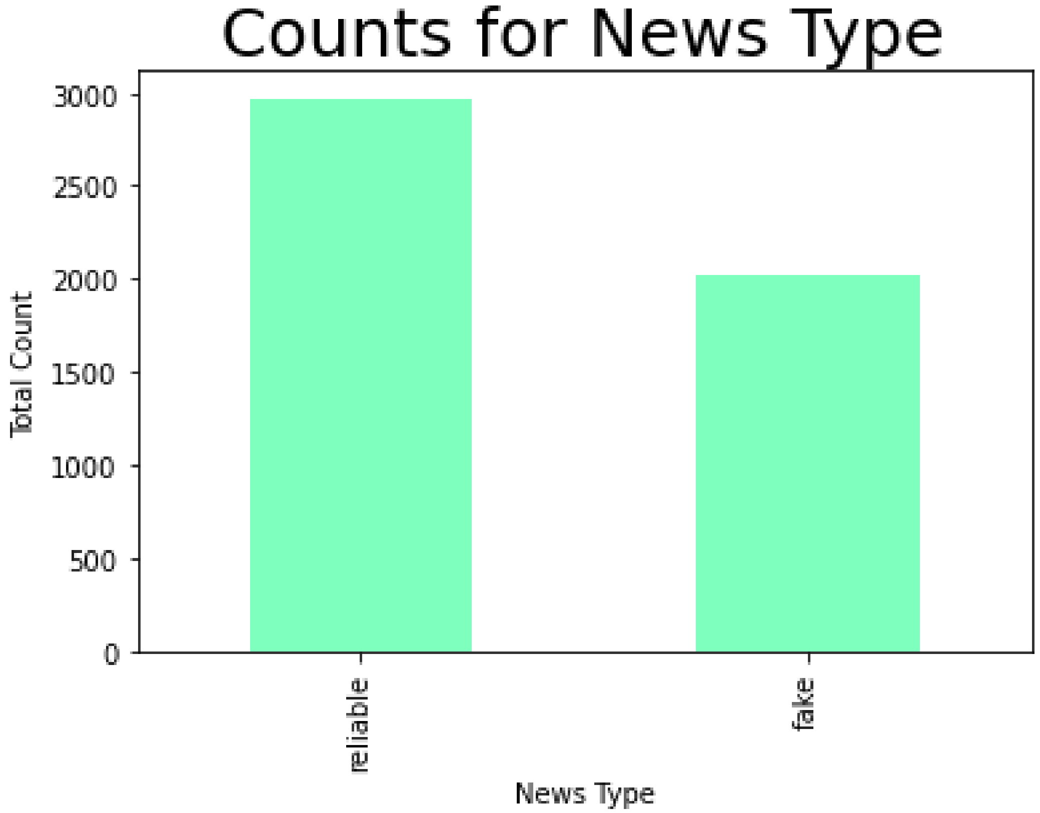
Project Report

The problem of fake online news is a persistent concern in contemporary society, impacting politics, the media, and society. While the Internet enables access to a wealth of information, it is also a medium by which disinformation can be easily spread. In particular, major websites with user-generated content have been met with harsh criticism and calls for legal action due to fake news being circulated on their platforms. Large websites with user-generated content can make use of machine learning to quickly identify sources as being potentially suspect or reliable.

In this project I have classified news into two categories, fake and reliable. I have taken a small dataset with about 5000 rows and there is an even distribution between the two class labels.



For regulatory agencies, disinformation concerning consumer products and health reporting presents a pressing problem, directly associated with goals in informing the public. In politics, Fake news may hold way in influencing elections. Additionally, companies targeted by disinformation have an express interest in identifying and fighting falsehoods disseminated about them and may be interested in the general climate of online disinformation as a whole. Altogether, an analysis online ‘fake news’ is relevant for many organizations.
The hypothetical client for this project is a fact-checking organization, such as PolitiFact or FactCheck.org, that is interested in issues pertinent to automated fake news classification, and capabilities and limitations that machine learning can hold for fact-checking.

Then I used Count vectorizer and tfidf vectorizer for text classification. I preprocessed the text by removing the stop words to remove the low-level information from text to emphasize on the important information.

I used different algorithms test the accuracy of the predictions. I started with using naïve bayes and the got a f1 score of 0.71. I then used some parameters to increase the accuracy of the predictions. By using alpha, the accuracy increased up to 0.76.
I also used support vector machines and got an accuracy of 0.76

```
              precision    recall  f1-score   support

           0       0.75      0.92      0.82       983
           1       0.81      0.54      0.64       663

    accuracy                           0.76      1646
   macro avg       0.78      0.73      0.73      1646
weighted avg       0.77      0.76      0.75      1646
```

I then did some more preprocessing by removing punctuation and used the porter stemming algorithm to remove the common morphological and inflexional endings from the English words. This is the process to normalize English words. This converted the features into words.

The next problem was to reduce the number of features because the number of features as compared to the number of rows was very high. Therefore, I used the parameter max_features in the tfidf vectorizer to limit the number of max features.

```
Name of classifier being used with tfidf vectorizer:

MultinomialNB()

Classification Report :

               precision    recall  f1-score   support

           0       0.73      0.90      0.81       983
           1       0.77      0.50      0.61       663

    accuracy                           0.74      1646
   macro avg       0.75      0.70      0.71      1646
weighted avg       0.75      0.74      0.73      1646


 Accuracy of classifier : 0.741190765492102
```

```
Name of classifier being used with tfidf vectorizer:

LinearSVC()

Classification Report :

               precision    recall  f1-score   support

           0       0.75      0.80      0.77       983
           1       0.67      0.60      0.63       663

    accuracy                           0.72      1646
   macro avg       0.71      0.70      0.70      1646
weighted avg       0.72      0.72      0.72      1646


 Accuracy of classifier : 0.7199270959902795

Name of classifier being used with tfidf vectorizer:

XGBClassifier

Classification Report :

               precision    recall  f1-score   support

           0       0.73      0.88      0.80       983
           1       0.75      0.52      0.61       663

    accuracy                           0.74      1646
   macro avg       0.74      0.70      0.71      1646
weighted avg       0.74      0.74      0.72      1646


 Accuracy of classifier: 0.7363304981773997
```

I also applied logistic regression and decision tree classifier while keeping the max features limited to 1000 and got the following accuracy.

```
Name of classifier being used with tfidf vectorizer:
```

```
Classification Report :

              precision    recall  f1-score   support

           0       0.73      0.88      0.80       983
           1       0.75      0.53      0.62       663

    accuracy                           0.74      1646
   macro avg       0.74      0.70      0.71      1646
weighted avg       0.74      0.74      0.73      1646


Name of classifier being used with tfidf vectorizer:

Classification Report :

              precision    recall  f1-score   support

           0       0.75      0.75      0.75       983
           1       0.63      0.63      0.63       663

    accuracy                           0.70      1646
   macro avg       0.69      0.69      0.69      1646
weighted avg       0.70      0.70      0.70      1646


 Accuracy of classifier : 0.6986634264884569
```