**Milestone Report 2:**

------------------------------------------------------------------------------------------------------------------------

The Dataset is about fake news and can be found on kaggle:
https://www.kaggle.com/c/fake-news/data

------------------------------------------------------------------------------------------------------------------------

## Modeling:

In the last step, a statistical test was performed to see if there is a significant difference in the means of text polarity between articles that are fake and all text articles? Based on the statistical analysis of a one sample t-test, there was a significant difference between the mean text polarity of all articles and the mean text polarity of fake news articles.

This next step, which is the crux of our analysis, requires training models to predict whether a news article is fake or real. Before we can train the models, we need to convert the text into suitable input for these algorithms. To do this, we will use two methods that are called Count Vectorizer and TF-IDF Vectorizer.

The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. Conversely, TF-IDF (Term Frequency – Inverse Document Frequency) are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents. The TfidfVectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents.

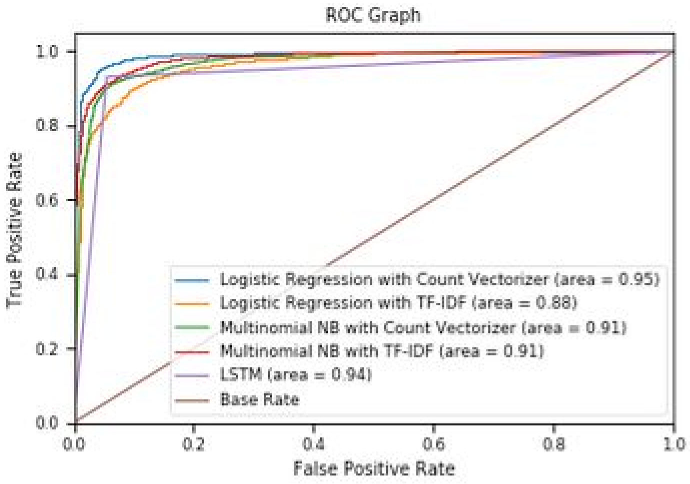
Each of these two methods will act as input for both a Multinomial Naive Bayes Classifier and a Logistic Regression Classifier. Logistic Regression was used as the benchmark model. A benchmark model is a model that is used for reference to compare how better other models are against it. No hyper-parameter tuning was performed for logistic regression and it was trained with a C value of 0.01. Alternatively, for the Multinomial NB model, hyper-parameter tuning was performed and found alpha = 0.1 to be the best value in both the CountVectorizer and TF-IDF Vectorizer models.

The LSTM (Long Short Term Memory) Neural Networks, are a particular type of recurrent neural networks. "*In a simple way, LSTM networks have some internal contextual state cells that act as long-term or short-term memory cells. The output of the LSTM network is modulated by the state of these cells. This is a very important property when we need the prediction of the neural network to depend on the historical context of inputs, rather than only on the very last input."*

The LSTM  model was created with the maximum number of words to be used to be 50,000 and the max number of words in each text to be 250. The LSTM model had a SpatialDropout1D value of 0.2, dropout value of 0.1 and recurrent dropout value of 0.1.

AUC - ROC curve is a performance measurement for classification problems at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much a model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between articles that are real or fake. Below is a table of the classifiers used and their respective AUC-ROC scores.

| Classifier | AUC-ROC Score | Accuracy |
|---|---|---|
| Multinomial NB (Count Vectorizer) | 0.91 | 0.91 |
| Multinomial NB (TF-IDF) | 0.91 | 0.91 |
| Logistic Regression (Count Vectorizer) | 0.95 | 0.95 |
| Logistic Regression (TF-IDF) | 0.88 | 0.88 |
| LSTM | 0.94 | 0.94 |



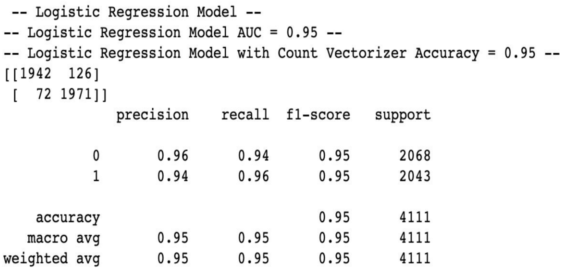
Logistic Regression with Count Vectorizer to tokenize the data gave us the best results. It achieved an accuracy of 95% and an AUC-ROC score of 95%. Although the LSTM model achieved a higher AUC-ROC score, it would be an overkill to use it since Logistic Regression performs better in accuracy and has a great AUC-ROC score. Also, Logistic Regression is less computationally expensive.

The worst model performance was also by Logistic Regression with TF-IDF Vectorizer achieving an accuracy and AUC-ROC score of 0.88.

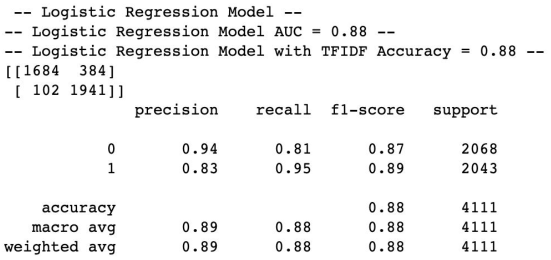
### Model Evaluation:

Apart from just using accuracy as an evaluation metric, we also use the AUC-ROC score which plots the True Positive Rate against the False Positive Rate. We should also take into consideration the False Positive and False Negative Errors to evaluate our models' performances.

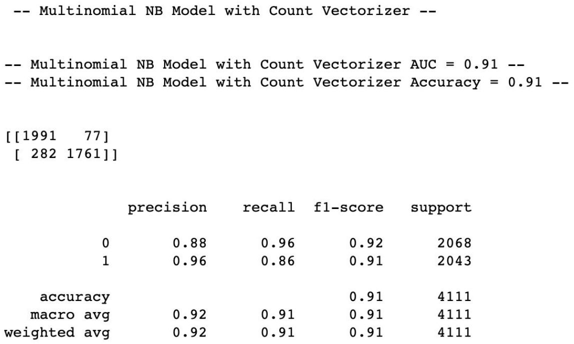
False Positives (Type I Error): You predict that the article is Fake but is Real.
False Negatives (Type II Error): You predict that the article is Real but is Fake.



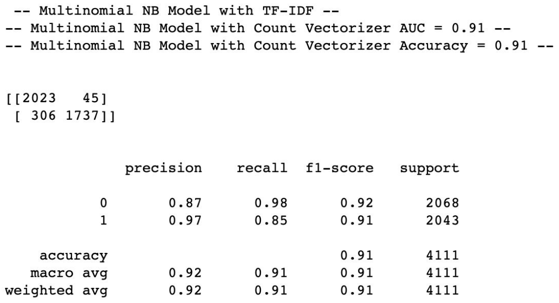
```
 -- Logistic Regression Model --
-- Logistic Regression Model AUC = 0.95 --
-- Logistic Regression Model with Count Vectorizer Accuracy = 0.95 --
[[1942  126]
 [  72 1971]]
              precision    recall  f1-score   support

           0       0.96      0.94      0.95      2068
           1       0.94      0.96      0.95      2043

    accuracy                           0.95      4111
   macro avg       0.95      0.95      0.95      4111
weighted avg       0.95      0.95      0.95      4111
```



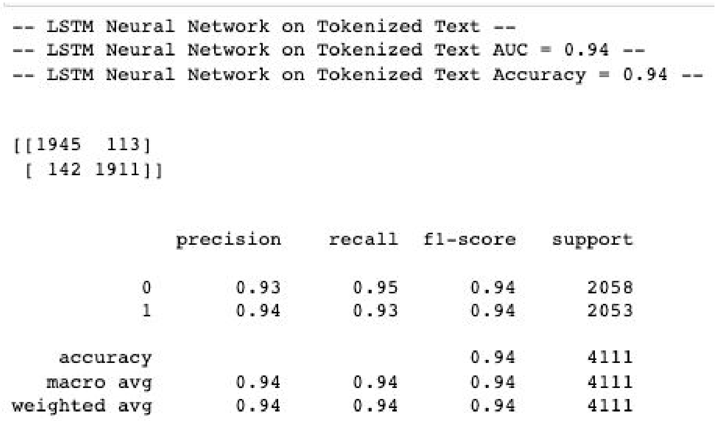
```
 -- Logistic Regression Model --
-- Logistic Regression Model AUC = 0.88 --
-- Logistic Regression Model with TFIDF Accuracy = 0.88 --
[[1684  384]
 [ 102 1941]]
              precision    recall  f1-score   support

           0       0.94      0.81      0.87      2068
           1       0.83      0.95      0.89      2043

    accuracy                           0.88      4111
   macro avg       0.89      0.88      0.88      4111
weighted avg       0.89      0.88      0.88      4111
```



```
 -- Multinomial NB Model with Count Vectorizer --


-- Multinomial NB Model with Count Vectorizer AUC = 0.91 --
-- Multinomial NB Model with Count Vectorizer Accuracy = 0.91 --


[[1991   77]
 [ 282 1761]]


              precision    recall  f1-score   support

           0       0.88      0.96      0.92      2068
           1       0.96      0.86      0.91      2043

    accuracy                           0.91      4111
   macro avg       0.92      0.91      0.91      4111
weighted avg       0.92      0.91      0.91      4111
```



```
 -- Multinomial NB Model with TF-IDF --
-- Multinomial NB Model with Count Vectorizer AUC = 0.91 --
-- Multinomial NB Model with Count Vectorizer Accuracy = 0.91 --


[[2023   45]
 [ 306 1737]]


              precision    recall  f1-score   support

           0       0.87      0.98      0.92      2068
           1       0.97      0.85      0.91      2043

    accuracy                           0.91      4111
   macro avg       0.92      0.91      0.91      4111
weighted avg       0.92      0.91      0.91      4111
```



```
-- LSTM Neural Network on Tokenized Text --
-- LSTM Neural Network on Tokenized Text AUC = 0.94 --
-- LSTM Neural Network on Tokenized Text Accuracy = 0.94 --


[[1945  113]
 [ 142 1911]]

              precision    recall  f1-score   support

           0       0.93      0.95      0.94      2058
           1       0.94      0.93      0.94      2053

    accuracy                           0.94      4111
   macro avg       0.94      0.94      0.94      4111
weighted avg       0.94      0.94      0.94      4111
```

## Interpreting the Data:

## Summary:

The distribution of text polarity of all articles (both fake and real) seemed evenly distributed while some articles were completely negative and completely positive in their polarity. Similarly for real articles, the text polarity appeared evenly distributed and contained articles with completely negative or positive polarity. For fake articles, the distribution of text polarity also appeared to be normally distributed. However, an interesting thing to note is there are no articles that have perfectly negative or positive text polarity (i.e. -1 and +1).

A heatmap was used to check for any correlations between the different features in the dataset. There did not seem to be any strong correlations in the dataset. The greatest correlation value in the dataset was -0.12 between length and label, indicating a very weak negative correlation.

In all news articles, without removal of stop words, the top word that occured was ‘the’. With the removal of stop words, the top word that occured was ‘said’. Similarly, without removal of stop words, the top bigram that occured was ‘of the’. With removal of stop words, the top bigram that occured was ‘mr trump’. Lastly, in all news articles, without removal of stop words the top trigram that occured was ‘the united states’. With the removal of stop words, the top trigram that occured was ‘new york times’.

In all fake articles, without removal of stop words, the top word that occured was ‘the’. With the removal of stop words, the top word that occured was ‘said’. Similarly, without removal of stop words, the top bigram that occured was ‘of the’. With removal of stop words, the top bigram that occured was ‘mr trump’. Lastly, in all news articles, without removal of stop words the top trigram that occured was ‘the united states’. With the removal of stop words, the top trigram that occured was ‘new york times’.

In all real articles, without removal of stop words, the top word that occured was ‘the’. With the removal of stop words, the top word that occured was ‘trump’. Similarly, without removal of stop words, the top bigram that occured was ‘of the’. With removal of stop words, the top bigram that occured was ‘mr trump’. Lastly, in all news articles, without removal of stop words the top trigram that occured was ‘the united states’. With the removal of stop words, the top trigram that occured was ‘new york times’.

## Potential Solutions:

Now that we have built our classifier and also trained a LSTM network, both of which can be improved over time as we get more data and tweak our model settings, we can incorporate this model into the social media website. This can be done on the backend, where the model can be deployed to run against news stories that people share on the platform and flag the content as 'fake' or 'potential unverified news'. We can also discuss with the data engineering team what the best way to incorporate the model into the site would be and to make it easy, we can use kubernetes to productionize the model so newer versions of the model can replace the older version and there is no downtime or disruption of service from the users standpoint.

It is important that rather than just predicting if the news article is fake or real, we also have an estimate of the probability that the news article is fake or real. We could flag the users that share fake articles consistently, and shut down these accounts in hopes of retaining the loyal user base of the social media platform.

We can run into two sorts of problems with this approach. Firstly, consider that the news article is flagged as fake by the model, but in reality the news article is real. This is called a false positive and this mistake could be expensive, as it could lead to embarrassment for the social media company and also potential loss of users. Though, this could be rectified quickly by building a mechanism to report articles that have been falsely flagged as fake. This could be one of the many directions the evolution of this project can head in.

On the contrary, consider that the news article is flagged as real but in reality is actually fake, and continues to get shared unchecked with more and more people being exposed to the fake news while thinking it is real. This is called a false negative and it has the ability to seed distrust and polarize online users on sensitive issues using racist and hyper-nationalistic rhetoric. This is more harmful to the validity of our model as well as the reputation of the social media company, as it leads to people distrusting the platform and its ability to separate truth from lies. To mitigate this issue, a mechanism should be built that allows people to report if the article is fake and then a team assigned to fact checking the article make the ultimate decision based on their analysis. These are just some solutions that can be implemented, neither are they meant to be perfect nor do they claim to be.

Since, this is a relatively new area and we will learn more about how to counter the different tactics employed by individuals and states to spread disinformation on social media platforms.

To summarize all of the above mentioned points:

Solution 1:

We can rank the articles with a probability estimate of it being real or fake, indicating how confident we think a particular article is real or fake. This will provide the readers some context and enough to decide for themselves to be skeptical of the news or not. The social media platform isn't making the decision for their users but giving them the information to decide for themselves so it does not feel like the user is being coerced into believing something or not.

Solution 2: Provide free training to the users of the social media platform to identify fake news. For example, tips and resources can be shared that discuss common ways one can fact check a news article. It can be as simple as copy pasting information from the article into google, and seeing if the information adds up.

## Where do we go from here?

This problem is about equipping the social media company with actionable knowledge regarding their ability to identify fake news on their platform and countering it. When modeling the data, we should not use the predictive metric as our final solution. Instead, we should use the information we get from modeling and arm the social media platform users so they can carry out informed decision making.

Once the model is deployed, as more and more articles get shared on the platform, our dataset will grow. This will help make our model more accurate over time by allowing us to even test out different techniques or other models that may perform better or are more computationally efficient for our use case. For now, this model will do as it is better to have some knowledge about an article's validity than to be completely in the dark in this age of misinformation.

**References:**

**https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd**