Is it Fake News or Real News?

University of Connecticut

MS in Business Analytics and Project Management

OPIM-5512- Data Science with Python

Professor Ramesh Shankar

Abhinav Dubey, Himadri Chobisa, Kaileen Pfeiffer and Pratibha Purohit

GROUP 7

Contents

Executive Summary

In recent years, we have witnessed a rise in fake news, i.e., false pieces of information created with the intention of deception. The dissemination of this type of news poses a serious threat to cohesion and social well-being, since it fosters political polarization, generates the distrust of people with respect to their leaders and creates grave social issues. The huge amount of news that is disseminated through social media makes manual verification unfeasible, which has promoted the design and implementation of automatic systems for fake news detection. The creators of fake news use various stylistic tricks to promote the success of their creations, with one of them being to excite the sentiments of the recipients. This has led to text analytics in charge of determining the polarity and strength of opinions expressed in a text, to be used in fake news detection approaches, either as a basis of the system or as a complementary element.

As the name suggests, a deliberately misleading story is fake news. Fake news is basically the misinformation spread via various communication channels as small private conversations to as huge as print media, electronic media, social media etc. Fake news has become such a big menace in recent times, that it is very much capable of swaying the public's opinions at large, the latest being the wide spread of misinformation related to Covid-19 vaccines. Hence, to ascertain the veracity of a piece of news has become one of the greatest concerns as well as necessity of the current times. It has been now proved that fake news is designed deliberately with vested interests, sometimes for personal/political/business gains by organizations, and many times by websites to increase their web traffic dramatically.

A fake news story can affect the perception of a business. Detecting fake news can help a business release a quick response plan, create an information hub for the public, and create proactive communications to mitigate speculations that could impact their business before it becomes widespread. Understanding the parameters which influence the news can help companies, organizations and even the government come up with a contingency/mitigation plan to tackle any such situation.

Our approach for the dataset in hand was to detect any features which can distinguish between a fake news and true news after proper trimming of data with various NLP techniques, and ultimately developing a model which can accurately determine the outcome of the future news article. Majority of analysis was done with Python with some initial data cleaning in excel. A number of python libraries were employed under the NLTK package and others to parse, trim, clean and modify the data and create comparative bar graphs, word clouds, ngram of words etc.

to understand the data better and eventual vectorization of clean text was done, in order to run models to classify the data as true or fake.

Out of the three categorical models deployed, linear regression gave the best results with highest accuracy (99.34%) and precision (0.99). Based on the model results, we recommend that businesses and organizations at large should take advantage of the data at their disposal with machine learning techniques and come up with solution-oriented models and algorithms to proactively fight the web of fake news. We also recommend that all websites, newspapers etc. should have a fact checker with every post so that viewers get an idea about the same. In short, NLP is the most widely used and reliable technique to analyse true vs fake news.

Problem Statement

A fake news story can affect the perception of people, leading them to believe in something that may or may not be true. This can impact the reputation of a business or cause unreasonable customer expectations. Detection of fake news shared over social platforms is critical to mitigating its spread. There is a lot of activity online at any given moment; in just one minute there are 4.45 billion users online, 87,500 tweets published, 600,000 Facebook posts, and tons more. This illustrates just how quickly news, real or fake, can spread. Instead of relying purely on a hunch, a machine learning algorithm will be developed to identify when a news article may be fake. The algorithm can be used by companies that can detect early on what might be real vs. fake news about their business. They can use this information to build a clear response plan, create an up-to-date hub of information for the public to access, and create proactive or reactive communications to effectively mitigate speculations, misinformation, and topics that can potentially ruin their business. This is a data-driven algorithm trained with confirmed fake vs. real news articles that provides credibility to its ability to accurately predict fake news.

Literature

Other data analysts have also explored algorithms that could be used to predict real vs. fake content. With this dataset specifically, there were many discussions on Kaggle about what others were doing with the information or questions they had. Many of the discussion threads were surrounding the topic of, “what actually is considered fake news?”. Some questioned, is it just an embellishment? Does an opinion make it real or fake? Is the quote being used out of context? One author, Rohit Kulkarni wrote the statement, “Jimmy Wales is a whale”. Now, most would think this is a false statement because how could Jimmy Wales be a whale? This is of course under the assumption that Jimmy Wales is a human, but what if Jimmy Wales was the name of the whale? Then, this would be a true statement. Essentially, the point Rohit was trying to make is that it can be very difficult to analyze a sentence and deem it true vs. fake without understanding the context in which it was spoken or written.

Sage Journal published the article, “Big Data and Quality Data for Fake News and Misinformation Detection”. The article focused on various techniques that could be used to detect real vs. fake news. The first was a feature-based approach, which is an extraction and analysis of linguistic cues for identifying specific targets; like real vs. fake text. It uses techniques that look at n-grams, subjectivity, and polarity that have yielded interpretable results. The second approach they used was a deep learning technique, which is a machine learning method on artificial neural networks with representation learning. There were two main techniques within deep learning that the research explored, recurrent and convolutional neural networks. Recurrent networks work best for short text semantics and convolutional networks work best on long text semantics, where the presence or absence of features is a more distinguishing factor than their location or order. The author concluded that they needed better quality datasets to confidently state that their algorithm was accurately detecting real vs. fake information. They actually put out a “call to arms” requesting that companies provide them with datasets that had confirmed real vs. fake news to better train their models. So, the quality of data seems to be a common theme across what other analysts have observed.

Data Description

The dataset was pulled online from Kaggle. The information was split into two datasets - one with fake news articles and one with true news articles. There were about 40,000 records of information between the two datasets. The datasets were published between 2015-2017. This was a presidential election year, so the content in the data is largely dominated by political content. There were 4 variables in each dataset identifying the article title, subject, text and published date.

Methodology

Libraries Imported

There are over 137,000 python libraries present today. Python libraries play a vital role in developing machine learning, data science, data visualization, image, data manipulation application and more. To solve our classification problem to identify fake news or true news we imported the following libraries for our project. **(Appendix: Figure 1. a)**

1. Pandas
2. Numpy
3. NLTK
4. SeaBorn
5. Matplotlib
6. WordCloud
7. BeautifulSoup
8. Scikit Learn

Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors, and is likely to contain many errors. To make the process easier first both the datasets were checked for missing values and carriage return issues. Issues were found and removed around 150 rows in the dataset which had missing values and carriage return issues. After removing the missing values we reviewed the True News dataset and found that it had publishers' names like WASHINGTON (Reuters), etc at the beginning of each article. Including publisher names can make the model biased so they were removed from the dataset. Machines don't understand text data, they understand 1s and 0s. So a target class column was created in both the datasets in true and in fake news. True news was denoted by 1s and fake news was denoted by 0s. After creating the target class column we merged both the dataset.

After merging the True News and Fake news Dataset a new dataset was created with 43,000 data entries. In a classification problem the dataset should be balanced as an imbalanced dataset can dramatically impact our ability to gain a large enough or representative sample of examples from the minority class. So checked target class distribution in the merged dataset by plotting a bar chart representing the count of Fake News and True News Data Entries. The dataset came out to be somewhat balanced between fake vs. real news - there are about 23,000 fake news articles and 21,000 true news articles. Since the dataset is relatively balanced, balancing techniques like Synthetic Minority Oversampling Technique(SMOTE) was not applied. **(Appendix: Figure 2. a)**

In the dataset there was a “Subject” variable that categorizes each news article. To check the breakdown of articles into various categories we created a bar graph in which blue bars represent the subjects contained in the true news set like Political and U.S. news and the red bars represent the subjects contained in the fake news set - things like, Government news and mid-east news. Since we were not concerned with the categories for our project, So the subject column from our dataset was removed. **(Appendix: Figure 2. b)**

As our aim is to know if the news is fake or true as a dimension reduction step the Title and text column were merged together. So in the merged dataset we have one column containing title and text column and another is target column with true news as 1s and fake news as 0s.

Cleaning the Text data

Before building models, some text data cleaning is required as our predictor variable has text in all the data entries. Text data contains a lot of noise either in the form of symbols or in the form of URLs, punctuations, stopwords, numeric values and duplicate values. Therefore, it becomes necessary to clean the text, not just for making it more understandable but also for getting better insights. These impediments were removed from the text by using different Python libraries and packages like beautifulsoup, regular expressions library and NLTK

First beautifulsoup python library was used to parse the human-readable text from the existing text. Parsing the text helped us in removing the HTML tags from the text. Once the text was parsed the predictor variable was checked for the duplicate entries and removed around 5K of the duplicate entries. Removing duplicates is important as they lead to bad reporting and skewed metrics. Used a regular expression library to remove the existing numeric values, punctuations and URLS.

In the text there were many words which can be distracting and non-informative (or non-discriminative). These words are called Stop Words. Removing stop words can potentially help improve the performance as there are fewer and only meaningful tokens left. Thus, it could increase classification accuracy. Based on this concept, using the NLTK package removed a list of generic stop words from the English vocabulary. After text data cleaning the final dataset has around 35,000 rows of data to do data visualization and data modelling.

Data Visualization

Various data visualization techniques were used to better understand what the data consisted of. First, a WordCloud library was imported to identify which words were being used more or less frequently in the dataset. Words like, “US”, “Trump”, and “People” were frequently used in both the true and fake news datasets. Since this data was pulled during the election period, it makes sense that the datasets were dominated by political words. **(Appendix: Figure 3. a and Figure 3. b)**

Next, a few different Parteo charts of the data were put together to visualize the size of the information. Four plots were put together which included the number of characters, number of words, average word length, and N-Gram ranking of each article. The plots yielded similar results. For example, the average number of characters in a true news article was ~2,500. In comparison, a fake news article had an average of ~5,000 characters. A similar conclusion was made when comparing the number of words and average word length between the true and fake news articles. The true news articles on average had fewer words and word lengths in each article. **(Appendix: Figure 3. c and Figure 3. e)** Lastly, an N-Gram ranking technique was used to identify how many times a word is used independently, in a pair, or in three’s in what’s referred to as a Unigram, Bigram, Trigram, respectively. As the number of words used consecutively increases, the amount of times those words appear together decreases. In Unigram, the word “Trump” is used most frequently, over 140,00 times throughout the entire dataset. However, as that word gets paired with “President Donald Trump ", the number of times these words appear together decreases, to only about 7,000 times across the dataset. These visualizations helped show what type of information appeared in the dataset most frequently, so that the information did not have to be manually reviewed, row by row. **(Appendix: Figure 3. f and Figure 3. h)**

Predictive Analytics

After data processing and cleaning, we performed predictive analytics to predict if the news is fake or real. The dataset was split into Training and Test. We assigned 80% data to the Training set and 20% data to the Test set.

The classifiers do not directly work with the raw text data hence it has to be converted into numbers. TF-IDF method was followed to convert the text data into matrix format. In this method, a document term matrix is generated and each column represents a single unique word.The cell value represents a weighting that highlights the importance of that particular word to the document. Its score represents the relative importance of a term in the document and the entire corpus. TF-IDF score is composed by two terms: the first computes the normalized Term Frequency (TF), the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

TF-IDF formula:

TF = (freq of word X in doc Y)/(total number of words in doc Y)

IDF = (total docs)/(freq of word X across all docs)

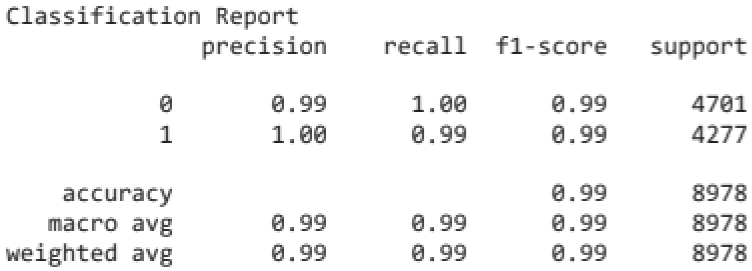
High weight means that the word occurs many times within a few documents and low weight means that the word occurs fewer times in a lot of documents or repeats across multiple documents.

Sklearn’s Tfidf Vectorizer was used for the vectorization portion in Python.

Next, this vector data was used to train different models. Since we have a target variable with 0 and 1 for fake and true news hence we will use Supervised machine learning technique and will work with 3 different models namely Logistic Regression, Naive Bayes & Random forest.

Logistic Regression

Its uses a flexible method to predict the probability of a binary variable like Yes/No, 0/1 or True/False and in our dataset the target variable is a binary variable with value 1 and 0 for true and fake news. The model had an accuracy of 99% and True positive and True negative numbers are quite high compared to False Positive and Negative. **(Appendix: Figure 4. a and Figure 4. b)**



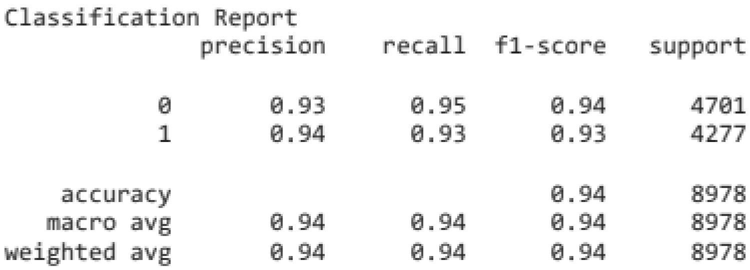
```
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      4701
           1       1.00      0.99      0.99      4277

    accuracy                           0.99      8978
   macro avg       0.99      0.99      0.99      8978
weighted avg       0.99      0.99      0.99      8978
```

Naive Bayes

Naive Bayes is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The model had an accuracy of 94% with a high number of True positives and True negatives. **(Appendix: Figure 4. c and Figure 4. d)**



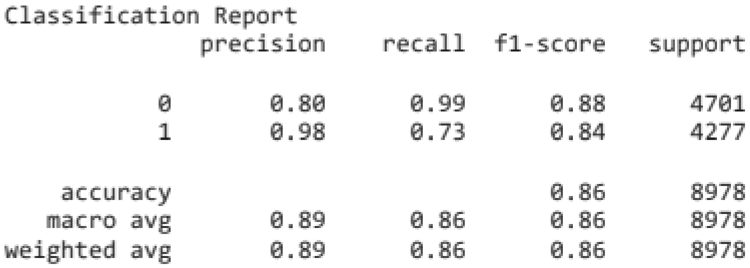
```
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      4701
           1       0.94      0.93      0.93      4277

    accuracy                           0.94      8978
   macro avg       0.94      0.94      0.94      8978
weighted avg       0.94      0.94      0.94      8978
```

Random Forest

Random Forest are ensemble models, particularly bagging models which are part of the tree based model family. They generate multiple different decision trees and after training all trees as forest then it takes the average/mean of all the outcomes from different trees to do the prediction. The accuracy of this model was around 86%. **(Appendix: Figure 4. e and Figure 4. f)**



```
Classification Report
              precision    recall  f1-score   support

           0       0.80      0.99      0.88      4701
           1       0.98      0.73      0.84      4277

    accuracy                           0.86      8978
   macro avg       0.89      0.86      0.86      8978
weighted avg       0.89      0.86      0.86      8978
```

Model Comparison

We compared the models on different parameters like Accuracy, Recall, AUC and Precision. All 3 models performed well and predicted the target variable with high accuracy but Logistic Regression had the highest accuracy among all the 3 models. It had 99% accuracy which is 5% more than Naive Bayes and around 15% more than Random Forest. Precision was also high for

the logistic regression compared to other 2 models. In our prediction Precision is a better parameter as it's a measure of quality. Higher precision means that an algorithm returns more relevant results than irrelevant ones and in our dataset, we want to predict fake and true news correctly instead of marking most news as fake. **(Appendix: Figure 5. a , Figure 5. b , and Figure 5. c)**
Logistics Regression had high values for all the parameters hence it was our final selected model.

Recommendations

Based on the research, analysis and findings performed, we came up with the following recommendations:

1. Online businesses should capitalize on machine learning models to find out if a text is fake or real. They can use the proposed hybrid deep learning model for fake text identification.

2. Regulation is required by governments to stop the advancement of fake texts. This will need appropriate public policy changes. If fake texts are controlled then market fluctuations will be less, leading to a stable economic environment.

3. Data Analytics is the enabler to solve this critical business problem of distinguishing between fake and real texts with high predictive accuracy, hence assisting the policy makers, businesses, communities, consumers and society.

4. Websites can add a ‘fact check’ option with every news and post to check the veracity of news. Multimedia content, particularly image and video, is becoming increasingly important on social media. Fake news is also increasingly accompanied by this type of content, so it will be necessary to enrich detection systems with multimedia analysis methods.

5. The most difficult fake news to detect are those in which falsehood has been subtly introduced, for example, expanding an authentic news piece with the addition of fake data or slightly modifying an authentic news story. In this case, aspect-based sentiment analysis and adversarial training can be of great help.

6. High-performance AI systems, particularly those based on deep learning, behave similar to black boxes that provide good results but can hardly justify a given output in a human-

understandable way. The creation of explainable AI systems is becoming more and more important, and therefore, it is necessary to add mechanisms both to the current analysis methods used and to the resulting fake news detection systems.

Conclusion

The 2016 US presidential election marked a turning point in the interest in fake news, both in society as a whole and in the scientific community in particular. This has led to the creation of a large number of resources, mainly in the form of data sets. The recent rise in the spread and social influence of fake news, driven by the popularization of social networks, has motivated a surge of interest in their automated detection.

One of the interesting findings in our dataset, was that the publisher's name was associated with all the true news articles, on the contrary, none of the fake news articles have publishers or any authentic source mentioned, so in future analysis- the source can be considered an important indicator. Out of various statistical parameters, Precision was best suited for model selection for our dataset as we want to make sure that the true positives (correctly detected true news and fake news) are mostly correct and our models are not biased. Logistic regression results satisfied all these parameters for differentiating true news from the fake ones.

We concluded that the current events (social/political/economic conditions) mostly dominate the news article and hence the most fake articles are also articulated around those only, so for every industry we need to understand the data and do separate repeated analysis to come up with a conclusive recommendation algorithm. Thus, we can say that the research field of fake news detection is currently in its transition from infancy to maturity. In this stage, the most pressing challenges in our view involve the need to guarantee the fairness, accountability, and transparency of systems (ensuring that results are explainable and free from harmful biases); the support for multilingualism and multimedia content; and the detection of fake news generated by subtly modifying authentic stories or by using text-generation algorithms. In short, text analytics and NLP can be used to work with the omnipresent fake news problem.

References

Kaggle Dataset: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

Sage Journal: https://journals.sagepub.com/doi/figure/10.1177/2053951719843310

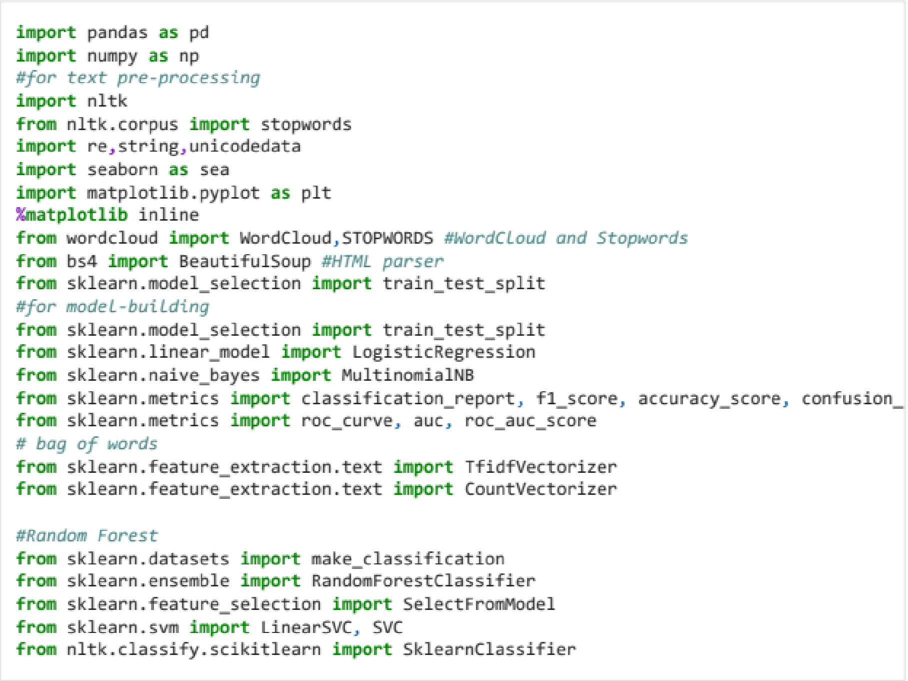
Other Articles we used:

1. https://medium.com/analytics-vidhya/nlp-tutorial-for-text-classification-in-python-8f19cd17b49e
2. https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a
3. https://towardsdatascience.com/text-classification-in-python-dd95d264c802
4. https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

Appendix

1. Libraries Imported

Figure 1. a



```
import pandas as pd
import numpy as np
#for text pre-processing
import nltk
from nltk.corpus import stopwords
import re,string,unicodedata
import seaborn as sea
import matplotlib.pyplot as plt
%matplotlib inline
from wordcloud import WordCloud,STOPWORDS #WordCloud and Stopwords
from bs4 import BeautifulSoup #HTML parser
from sklearn.model_selection import train_test_split
#for model-building
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, f1_score, accuracy_score, confusion_
from sklearn.metrics import roc_curve, auc, roc_auc_score
# bag of words
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer

#Random Forest
from sklearn.datasets import make_classification
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import LinearSVC, SVC
from nltk.classify.scikitlearn import SklearnClassifier
```

2. Data Preprocessing

Figure 2. a

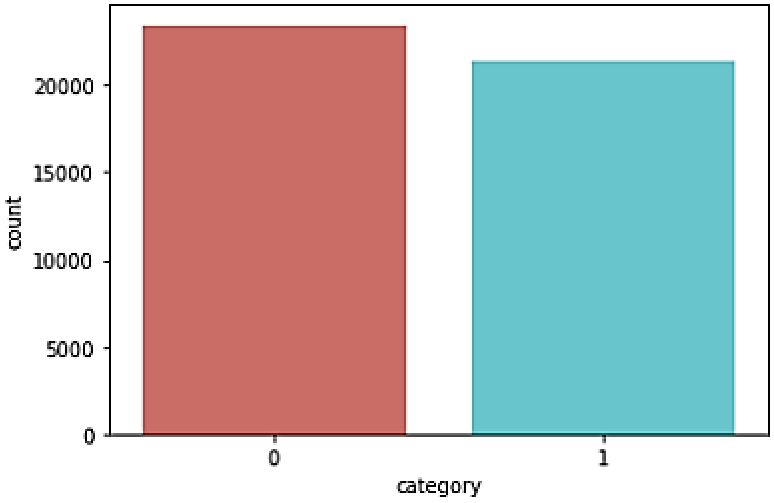
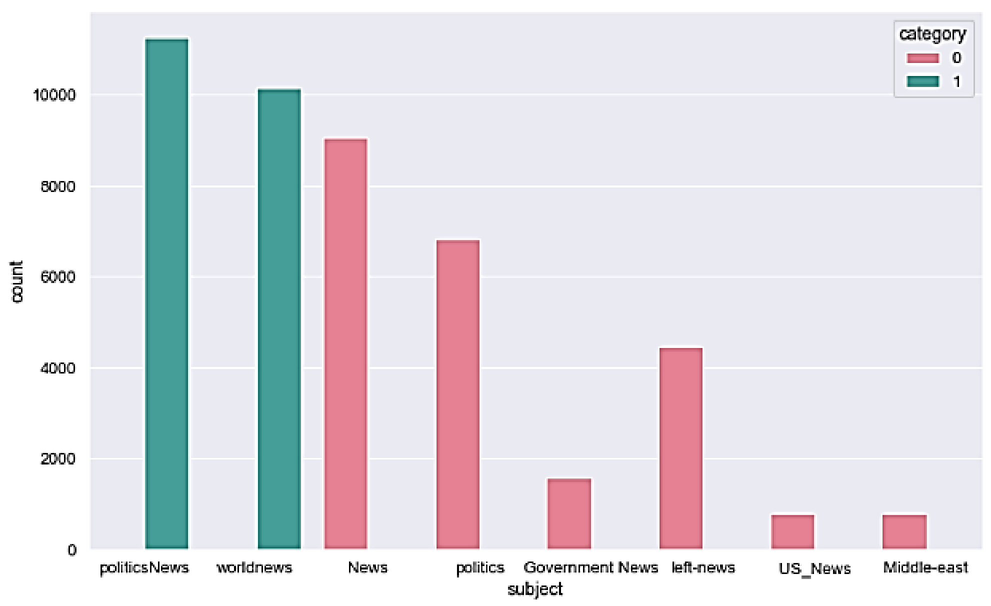
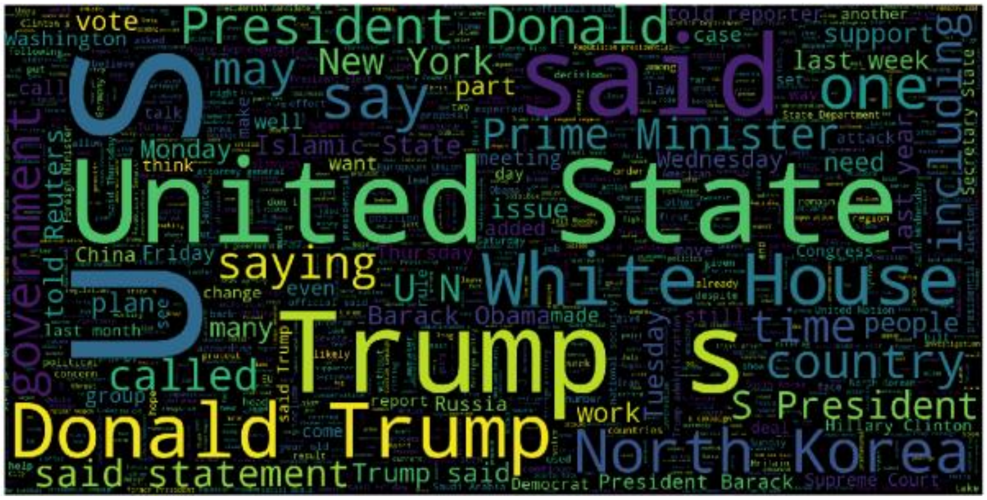


Figure 2. b	

3. Data Visualization

Figure 3. a	

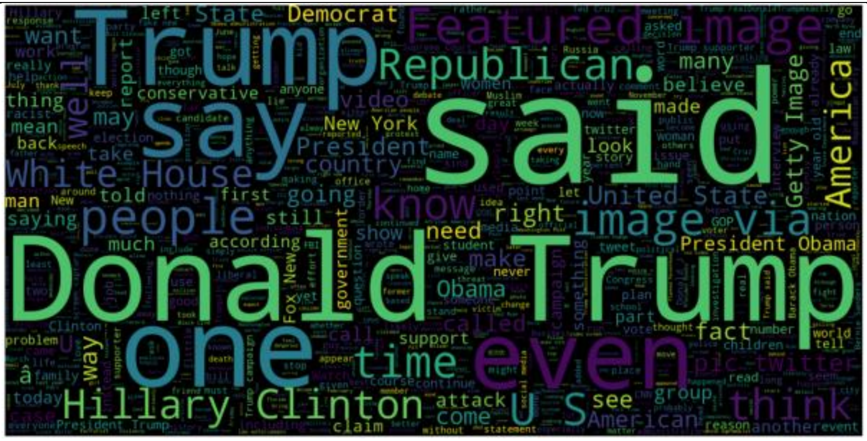
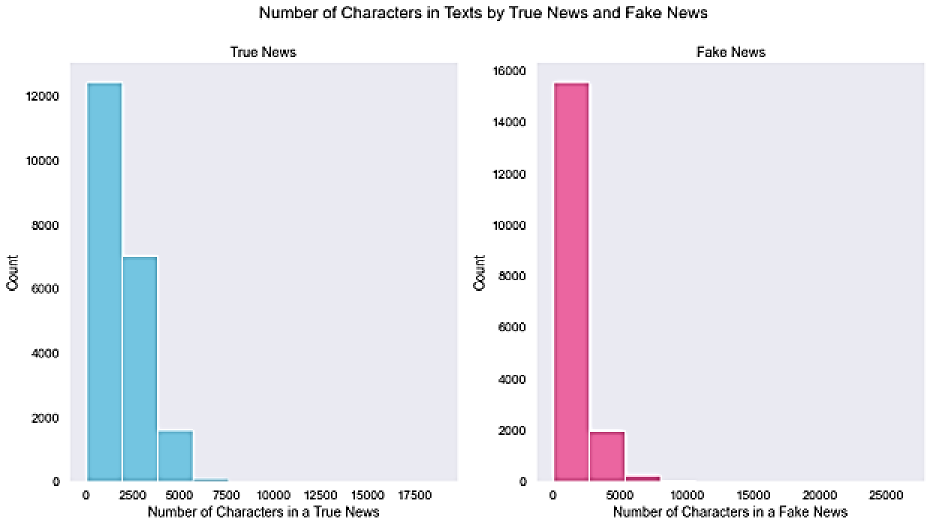
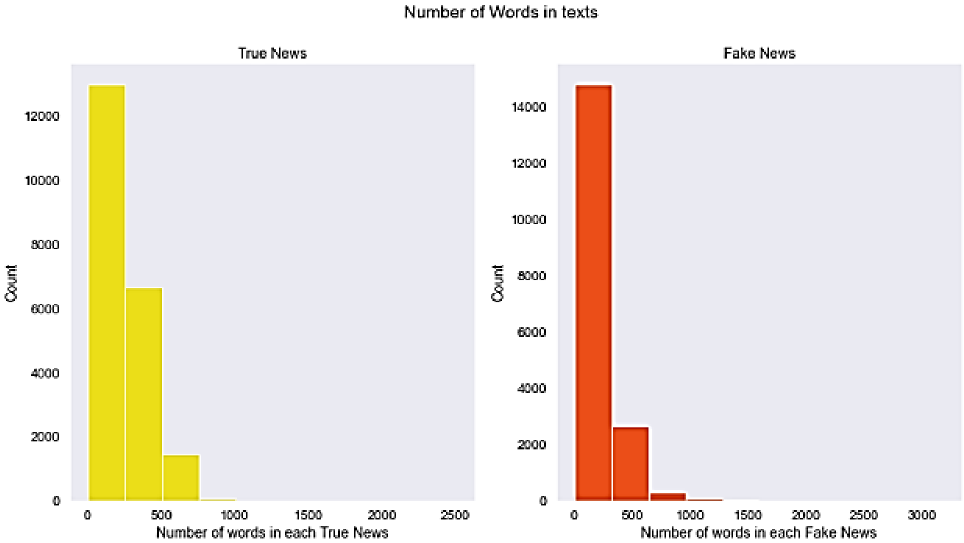
<table>
<tr><td>Figure 3. b</td><td>
</td></tr>
<tr><td>Figure 3. c</td><td>
</td></tr>
<tr><td>Figure 3. d</td><td>
</td></tr>
</table>

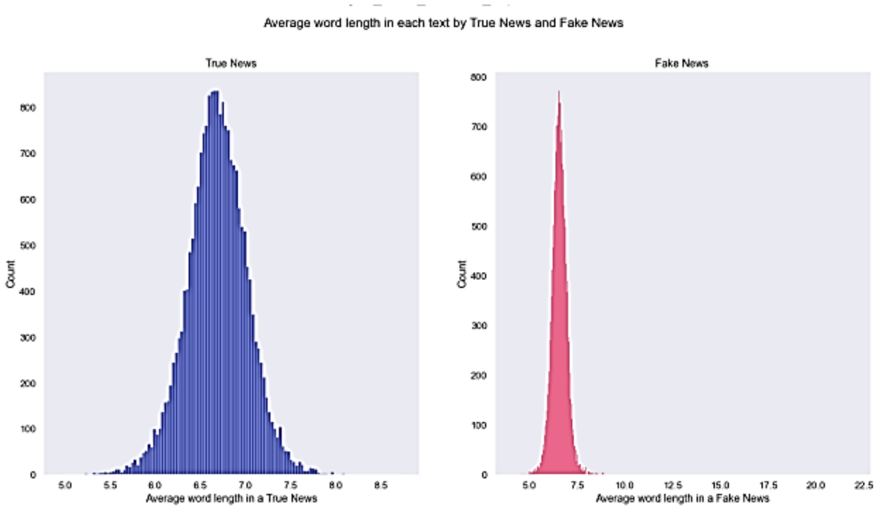
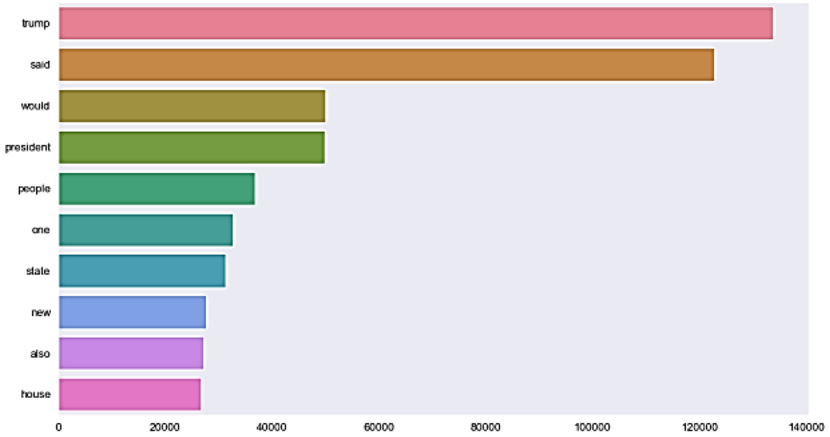
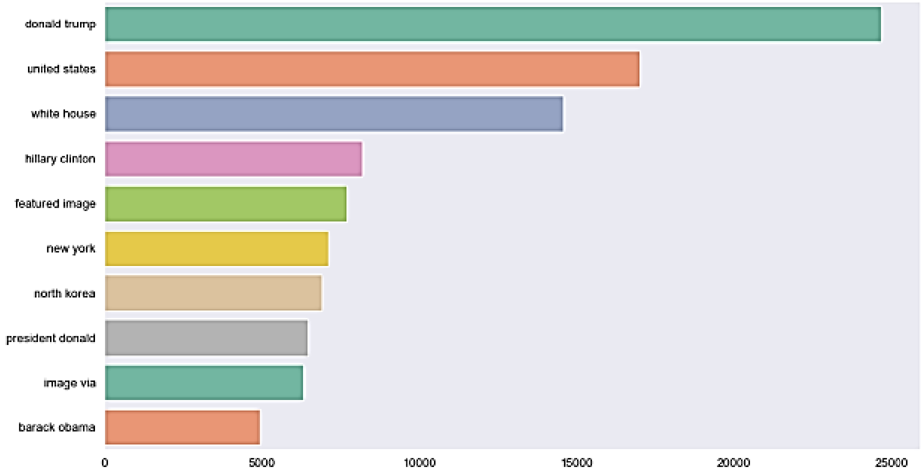
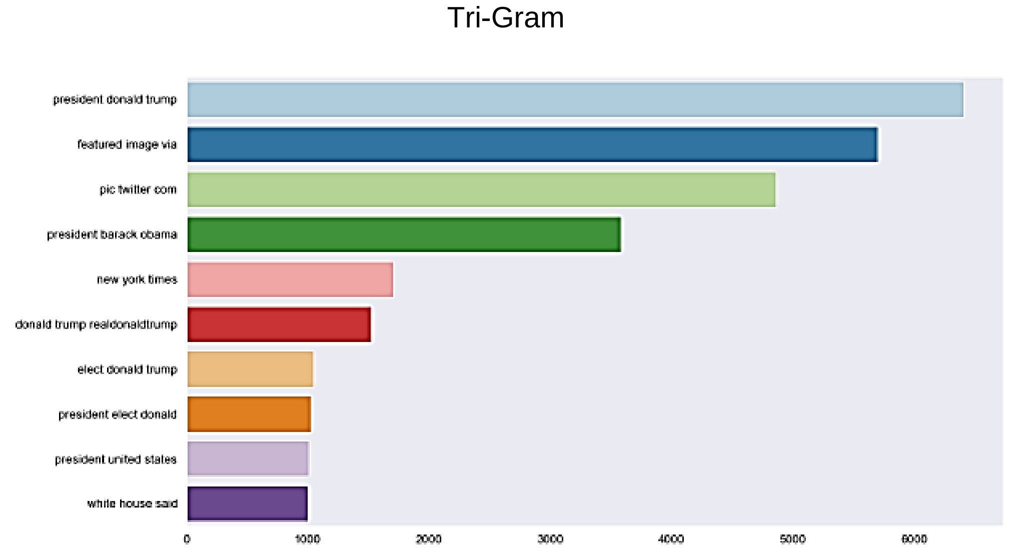
Figure 3. e	
Figure 3. f	Uni-Gram 
Figure 3. g	Bi-Gram 

Figure 3. h



4. Predictive Analytics

Figure 4. a

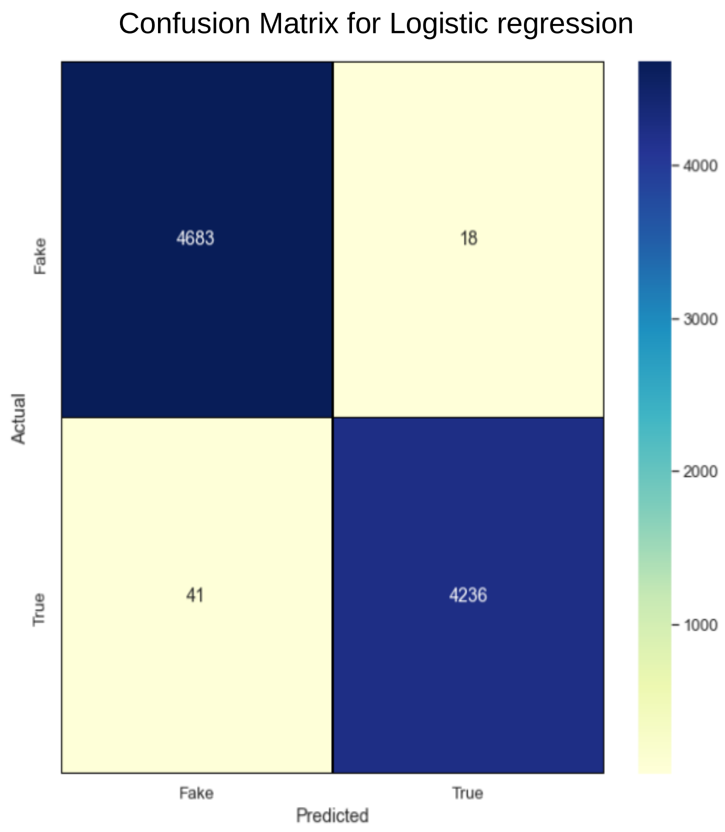


Figure 4. b

ROC Curve for Logistic Regression

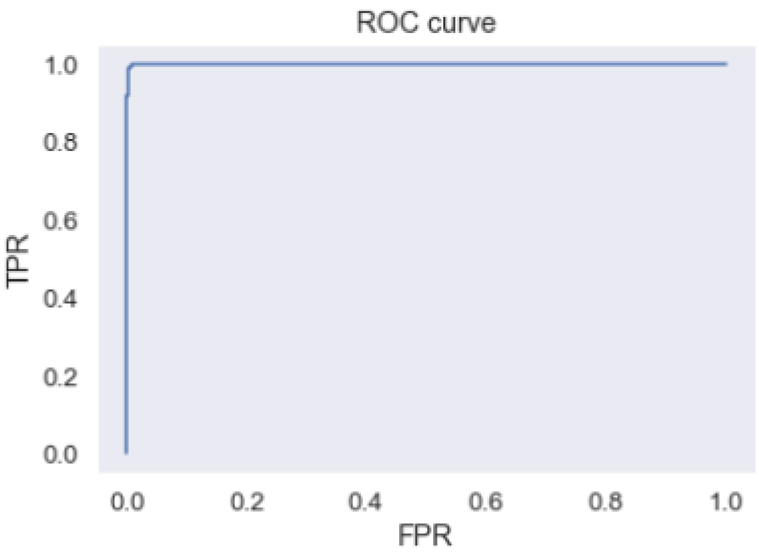


Figure 4. c

Confusion Matrix for Naive Bayes

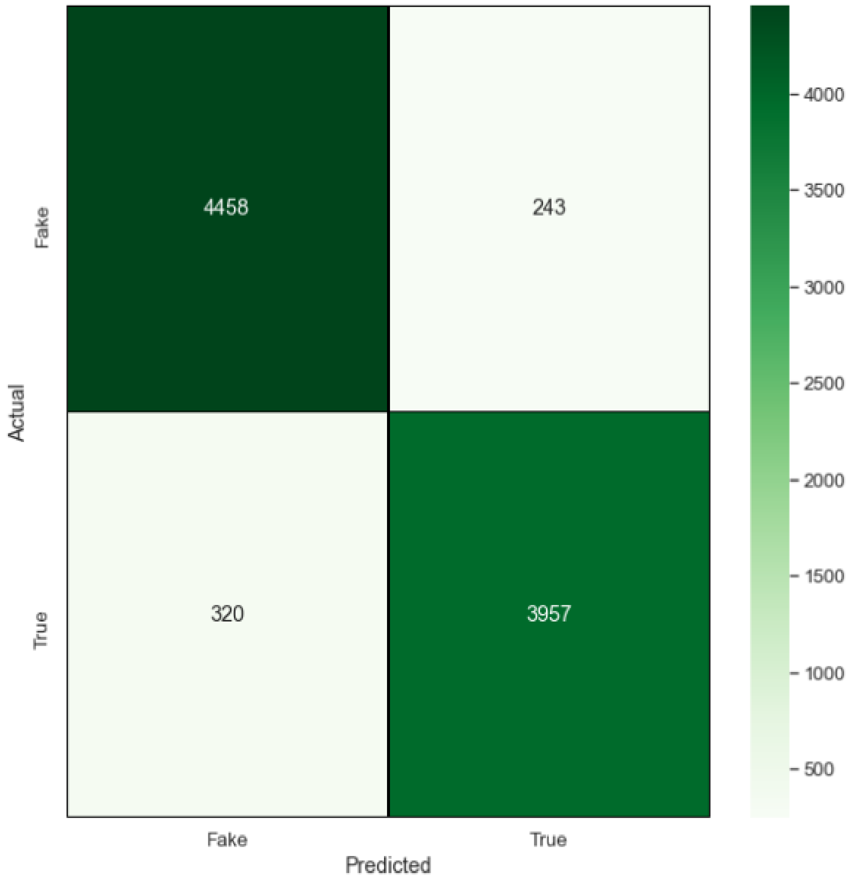


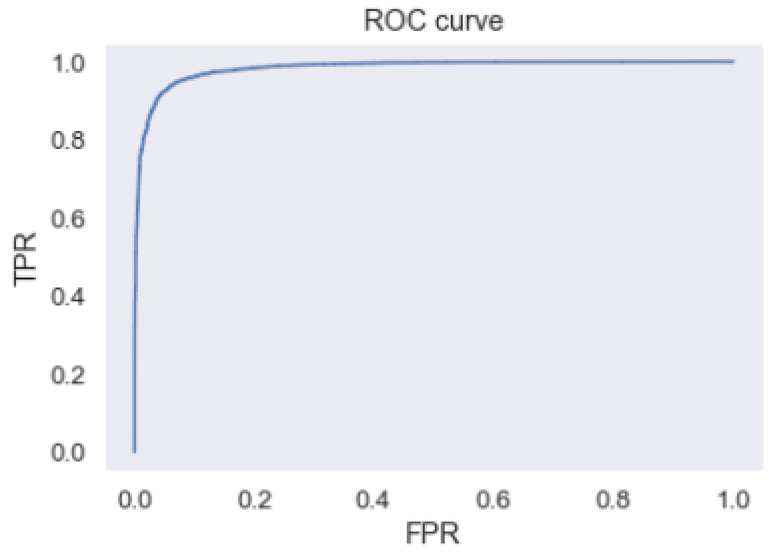
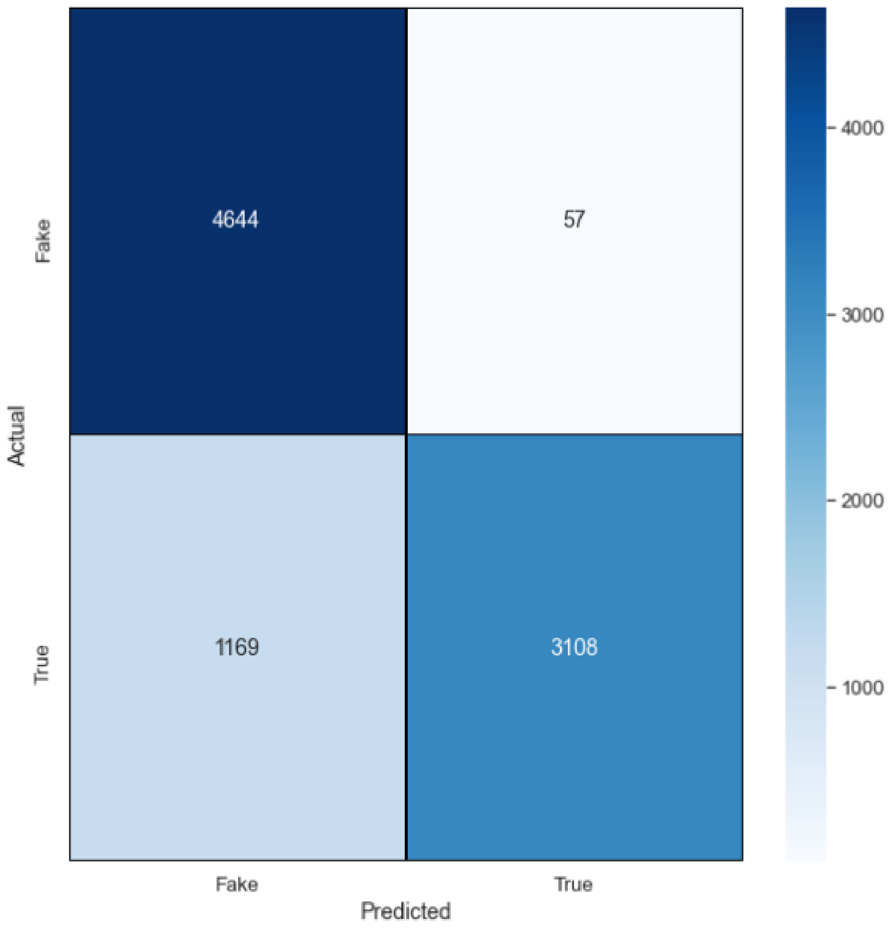
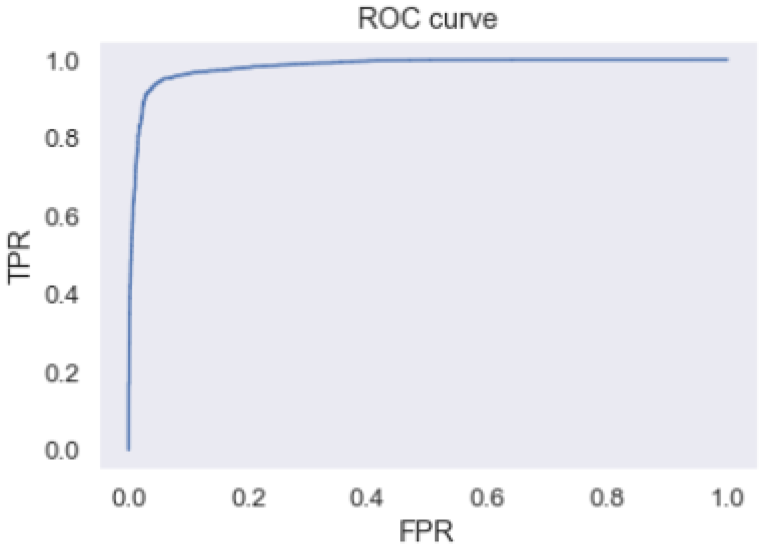
Figure 4. d	ROC Curve for Naive Bayes 
Figure 4. e	Confusion Matrix for Random Forest 

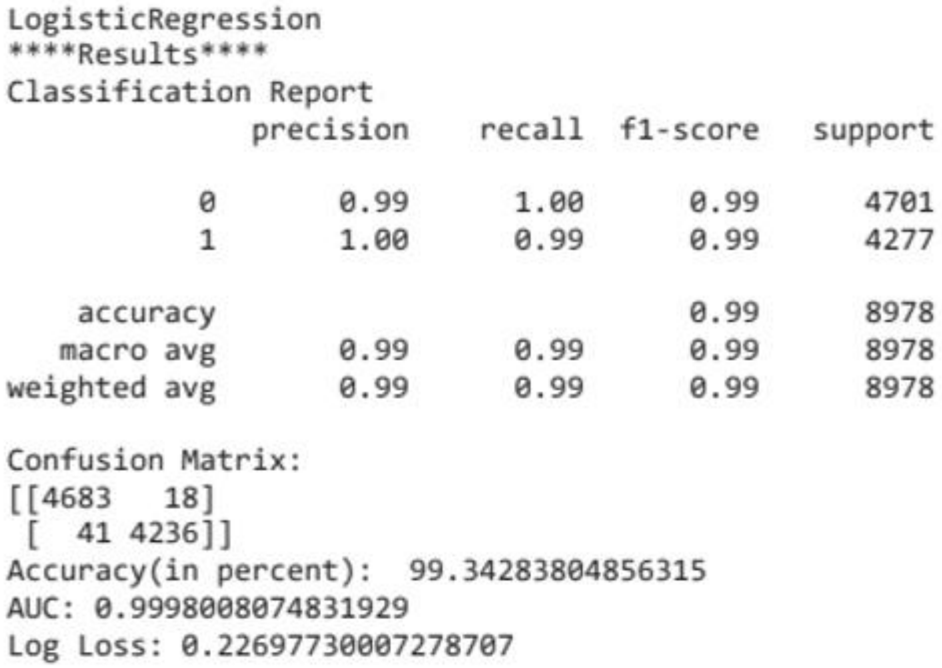
Figure 4. f

ROC Curve for Random Forest



5. Model Comparison

Figure 5. a



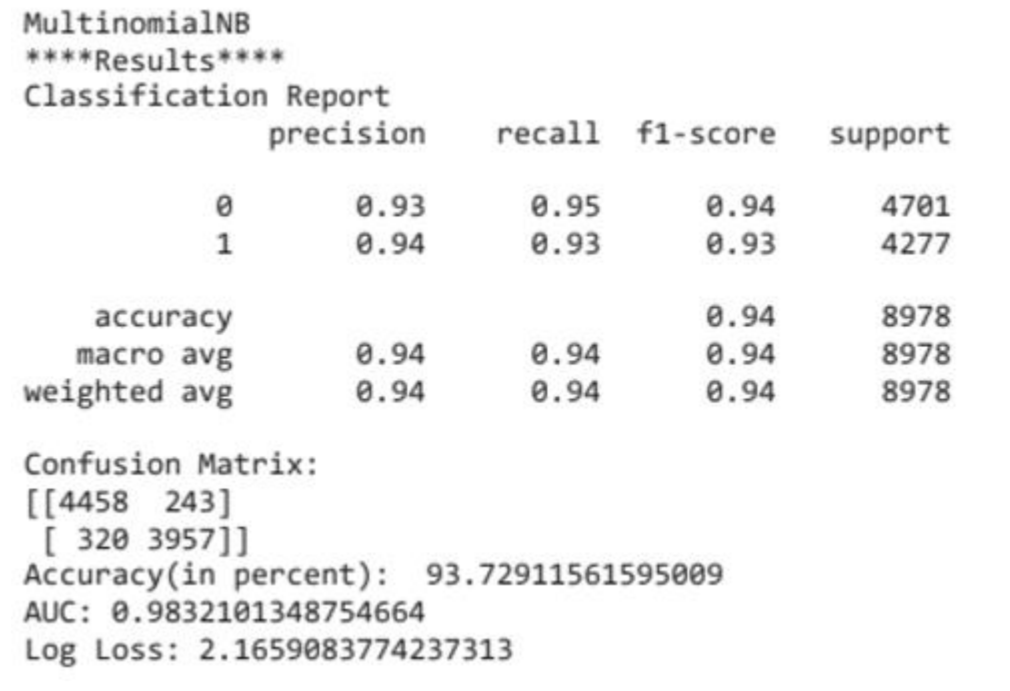
```
LogisticRegression
****Results****
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      4701
           1       1.00      0.99      0.99      4277

    accuracy                           0.99      8978
   macro avg       0.99      0.99      0.99      8978
weighted avg       0.99      0.99      0.99      8978

Confusion Matrix:
[[4683   18]
 [  41 4236]]
Accuracy(in percent):  99.34283804856315
AUC: 0.9998008074831929
Log Loss: 0.22697730007278707
```

Figure 5. b



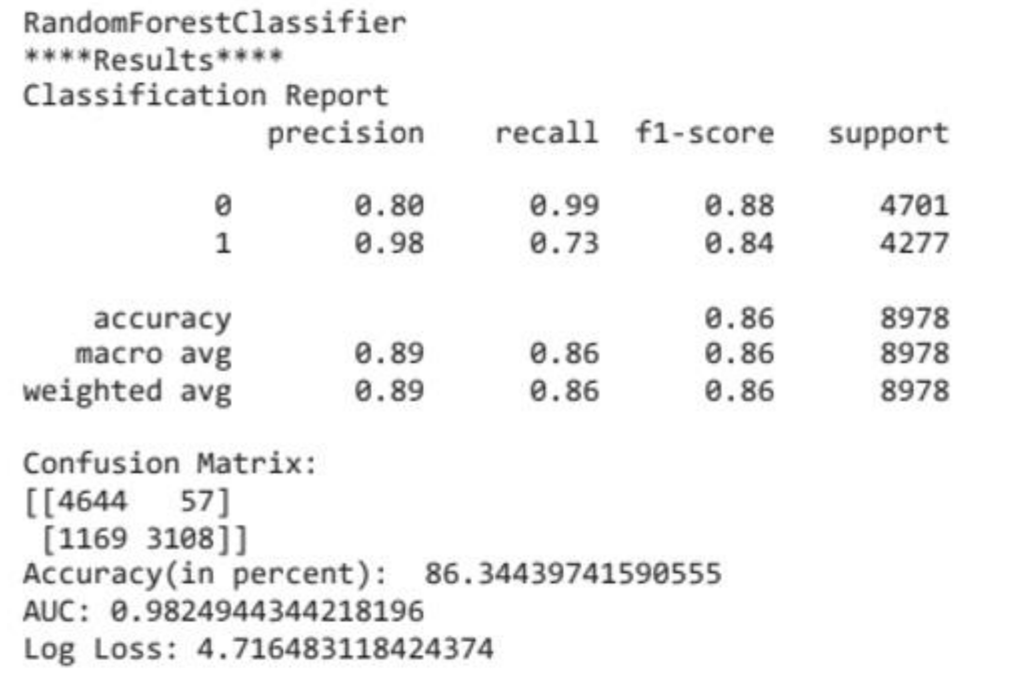
```
MultinomialNB
****Results****
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      4701
           1       0.94      0.93      0.93      4277

    accuracy                           0.94      8978
   macro avg       0.94      0.94      0.94      8978
weighted avg       0.94      0.94      0.94      8978

Confusion Matrix:
[[4458  243]
 [ 320 3957]]
Accuracy(in percent):  93.72911561595009
AUC: 0.9832101348754664
Log Loss: 2.1659083774237313
```

Figure 5. c



```
RandomForestClassifier
****Results****
Classification Report
              precision    recall  f1-score   support

           0       0.80      0.99      0.88      4701
           1       0.98      0.73      0.84      4277

    accuracy                           0.86      8978
   macro avg       0.89      0.86      0.86      8978
weighted avg       0.89      0.86      0.86      8978

Confusion Matrix:
[[4644   57]
 [1169 3108]]
Accuracy(in percent):  86.34439741590555
AUC: 0.9824944344218196
Log Loss: 4.716483118424374
```