FLIP ROBO

Fake News Detection Project

SHASHI SAHU

BATCH -30

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Flip Robo Technologies for supporting me throughout the internship and giving me the opportunity to explore the depth of Data Science by providing multiple projects like this, there are multiple people, organizations, youtubers, who guided me in this wonderful journey and few journals which helped me develop my models in this project. I would like to thank following people for the inspiration and help,

- Flip Robo Technologies
- DataTrained Team
- KAGGLE

References:

Research papers and YouTube videos that helped me in this projects is as follows:

- https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/
- https://www.sciencedirect.com/science/article/pii/S1877050918318210
- https://www.youtube.com/watch?v=TJU8NfDdqNQ&ab_channel=VictorLavrenko

INTRODUCTION

- ## Business Problem Framing

  The project is concerned with identifying a solution that could be used to detect and filter out sites containing fake news for purpose of helping users to avoid being lured by clickbait. It is imperative that such solutions are identified as they will prove to be useful to both readers and tech companies involved in the issue.

- ## Conceptual Background of the Domain Problem

  The idea of fake news is not a novel concept. Notably, the idea has been in existence even before the emergence of the Internet as publishers used false and misleading information to further their interests. Following the advent of the web, more and more consumers began forsaking the traditional media channels used to disseminate information for online platforms. Not only does the latter alternative allow users to access a variety of publications in one sitting, but it is also more convenience and faster. The development, however, came with a redefined concept of fake news as content publishers began using what has come to be commonly referred to as a clickbait

- ## Review of Literature

  If we look at some scholar work shows the issue that the fake news has been major concerned amongst scholar from various background. For instance, some authors have observed that fake news is no longer a preserve of the marketing and public relations departments. Instead there is a increasing risk of IT security, therefore, IT department is premised on the idea that it would help avert the various risks associated with the problem. So, if we good deeply into it we could find that the hackers use clickbait with the help of fake news and make some professional of the organization downloads their malicious exploits in their system or leak sensitive information, albeit in an indirect manner. The user may, for instance, be tricked into believing that they are helping to disseminate the news further when, in the actual sense, they are providing the perpetrators with access to their emails, and we can also see that the fake news are worked extensively as they are using videos with original massage and uses their facial structure to replace the massage with false massage they want us to believe, these fake news issues is bigger day by day and we need to implement more our research and extensive knowledge to solve the problem.

- ## Motivation for the Problem Undertaken

  This project was highly motivated project as it includes the real time problem of fake news which if we see are getting bigger, as there various concern as people do good things work hard to build a reputation, and only one false news is enough to ruin it all, it also have inverse effect on the financial market as if we observe there will a good amount of fluctuation on stock markets based on news.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

```
count    20761.000000
mean        72.574924
std         25.011968
min          3.000000
25%         59.000000
50%         75.000000
75%         87.000000
max        456.000000
Name: headline_length, dtype: float64
```

```
count     20761.000000
mean       4552.715380
std        5130.563491
min           1.000000
25%        1628.000000
50%        3361.000000
75%        6275.000000
max      142961.000000
Name: news_length, dtype: float64
```

From the above we can see that length of headlines and news and their statistical summary, so left side table is for headline and we can see that the average length of headlines is 72 and minimum length is 3 and max length is 456, and on right side table is for news, and we can see that the average is 4552, and minimum length is 1 and maximum length is 142961, so some news can be genuine, so we can't consider them outliers.

- Data Sources and their formats.

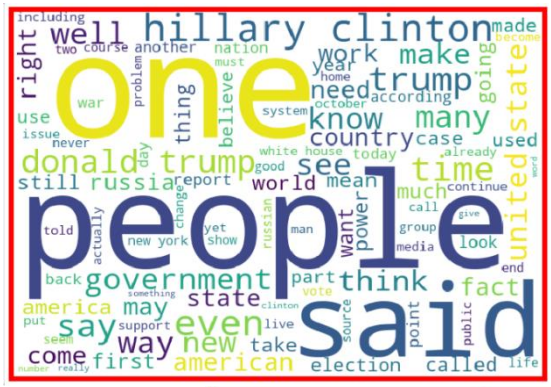
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  20800 non-null  int64 
 1   id          20800 non-null  int64 
 2   headline    20242 non-null  object
 3   written_by  18843 non-null  object
 4   news        20761 non-null  object
 5   label       20800 non-null  int64 
dtypes: int64(3), object(3)
memory usage: 975.1+ KB
```

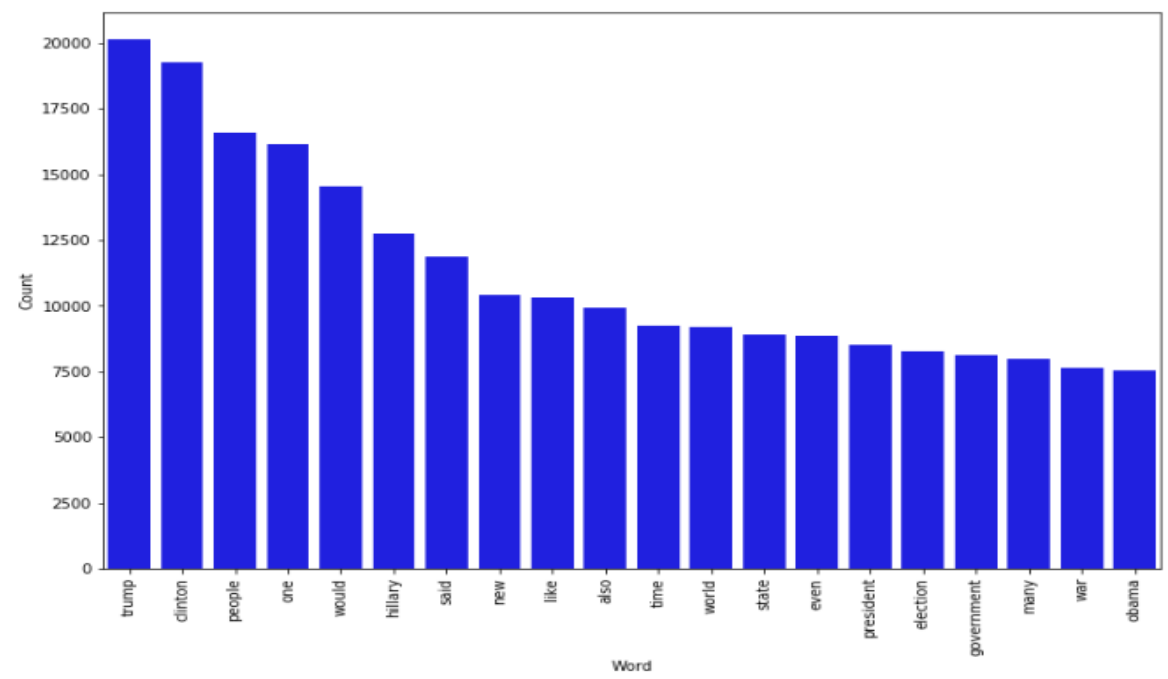
- Data Pre-processing Done

```
## Cleaining the text
from nltk.corpus import wordnet
lemmatizer = WordNetLemmatizer()
stemmer = PorterStemmer()
def clean_text(text):
    text=str(text)
    text = text.lower()
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', text)
    rem_num = re.sub('[0-9]+', '', text)
    tokenizer = RegexpTokenizer(r'\w+')
    tokens = tokenizer.tokenize(rem_num)
    filtered_words = [w for w in tokens if len(w) > 2 if not w in stopwords.words('english')]
    stem_words=[stemmer.stem(w) for w in filtered_words]
    lemma_words=[lemmatizer.lemmatize(w) for w in stem_words]
    return " ".join(filtered_words)
data["Full_News"] = data["news"].apply(lambda x: clean_text(x))
```

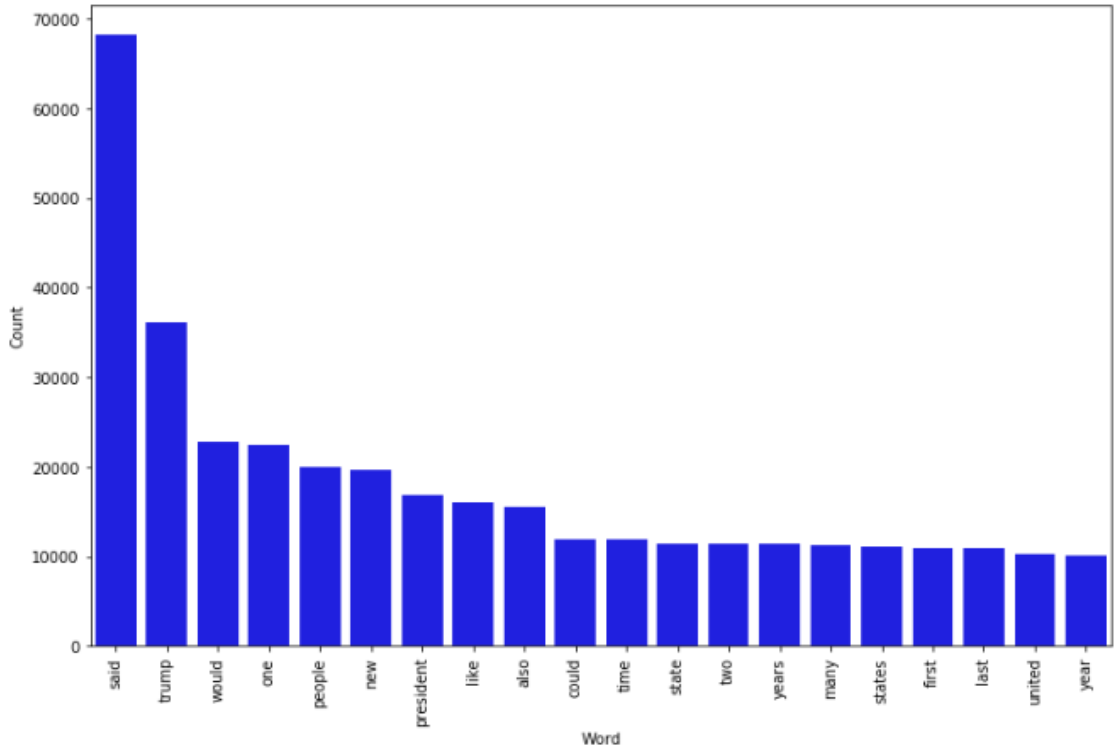
For Data pre-processing we did some data cleaning, where we used wordNet lemmatized and porter Stemmer to clean the words and removed special characters using Regexp Tokenizer and filter the words by removing stop words and then used lemmatizers and joined and return the filtered words.
Used TFIDF vectorizer to convert those text into vectors, and split the data and into test and train and trained various Machine learning algorithms.

- Data Inputs- Logic- Output Relationships





Fake News words Frequency



Real News words Frequency

From the above we can see that most frequent words on both labels and we can observe the words which are leading to fake new are trump, Clinton, people, one, would, etc and words which are leading to real news are said, trump, would, one, people, etc, so we can clearly see that above dataset extensively deals with news around US presidential elections between Trump and Clinton.

- Hardware and Software Requirements and Tools Used

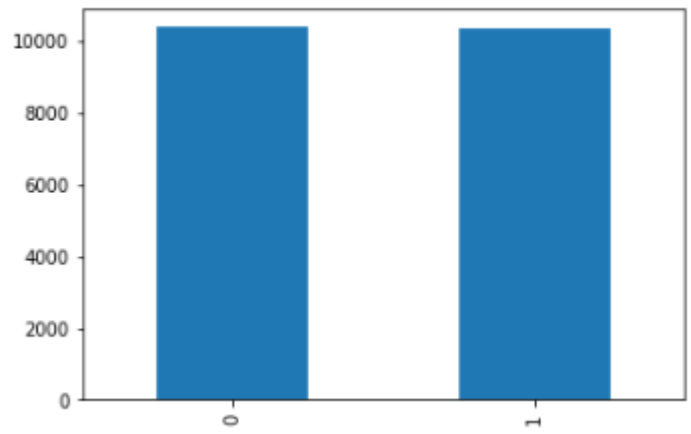
- Hardware: 8GB RAM, 64-bit, i7 processor.
- Software: Excel, Jupyter Notebook, python 3.6.
- Libraries used:

```
In [15]: ## importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from wordcloud import WordCloud
from nltk.corpus import wordnet
import string
import nltk
import ast
import re
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import WhitespaceTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import WordNetLemmatizer,PorterStemmer
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import wordnet
from nltk.corpus import sentiwordnet
from nltk import tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import confusion_matrix,classification_report,auc,roc_curve,roc_auc_score,precision_score,f1_score,accuracy_
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier,ExtraTreesClassifier
from sklearn.model_selection import cross_val_score
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches.

```
<matplotlib.axes._subplots.AxesSubplot at 0x251e04f5c88>
```



From the above we can see that the dataset is balanced which is good as it will help our model to classify more accurately, so we should expect good accuracy

score, and as the volume of data was also good, so was able to implement Passive Aggressive Algorithms.

- Testing of Identified Approaches (Algorithms)

- LR=LogisticRegression()
- DT=DecisionTreeClassifier()
- GNB=MultinomialNB()
- RFC=RandomForestClassifier()
- ETC=ExtraTreesClassifier()
- PAC=PassiveAggressiveClassifier()

- Run and Evaluate selected models

```
models = []
#models.append(('KNeighborsClassifier', KNN))
#models.append(('SVC', SV))
models.append(('LogisticRegression', LR))
models.append(('DecisionTreeClassifier', DT))
models.append(('MultinomialNB', GNB))
models.append(('RandomForestClassifier', RFC))
#models.append(('GradientBoostingClassifier', GBC))
models.append(('ExtraTreesClassifier',ETC))
#models.append(('AdaBoostClassifier',ABC))
models.append(('PasiveAggressiveClassifier',PAC))
```

```
Model = []
score = []
cvs=[]
rocscore=[]
for name,model in models:
    print('**********************',name,'**********************')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    pre=model.predict(x_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('Accuracy_score = ',AS)
    score.append(AS*100)
    print('\n')
    sc = cross_val_score(model, x, y, cv=10, scoring='accuracy').mean()
    print('Cross_Val_Score = ',sc)
    cvs.append(sc*100)
    print('\n')
    false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,pre)
    roc_auc = auc(false_positive_rate, true_positive_rate)
    print ('roc_auc_score = ',roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('classification_report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    print(sns.heatmap(cm,annot=True))
    plt.subplot(912)
    plt.title(name)
    plt.plot(false_positive_rate, true_positive_rate, label='AUC = %0.2f'% roc_auc)
    plt.plot([0,1],[0,1],'r--')
    plt.legend(loc='lower right')
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    print('\n\n')
```

We have created a loop and used models sets and passed them one by one to visualize and display the matrices and accuracy of different models, so that we can create an evaluation matrix of different models.

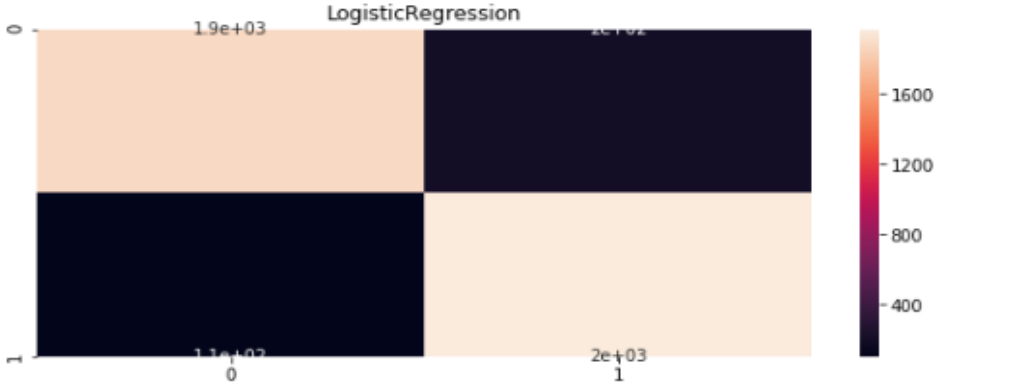
- Key Metrics for success in solving problem under consideration

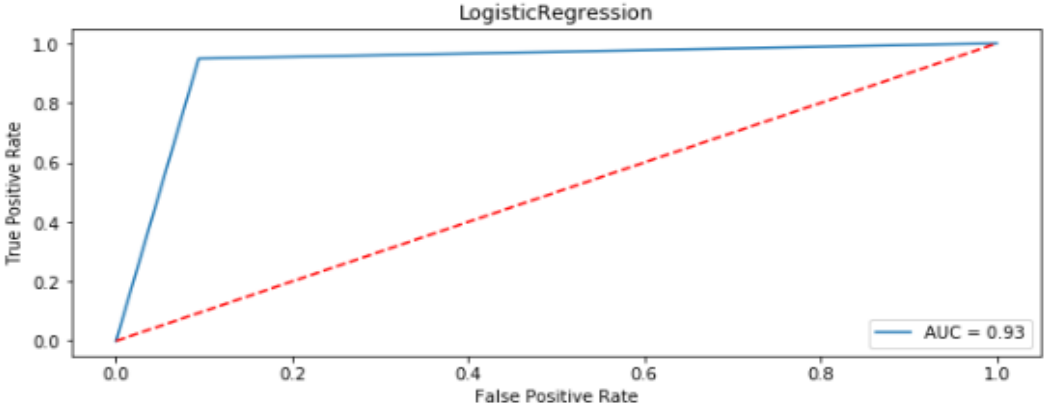
	Model	Accuracy_score	Cross_val_score	Roc_auc_curve
0	LogisticRegression	92.752227	92.596758	92.751711
1	DecisionTreeClassifier	85.287744	86.151972	85.288131
2	MultinomialNB	92.029858	92.211483	92.031521
3	RandomForestClassifier	87.816037	88.844667	87.816152
4	ExtraTreesClassifier	89.092222	89.798252	89.092274
5	PasiveAggressiveClassifier	94.943414	95.241093	94.943147

From the above we can see that there are various models out of which we few gave good accuracy score as more than 90%, however passive aggressive algorithm performed the best then the rest of the models as we achieved 94% which was expected to perform the best out of all classifier algorithms.

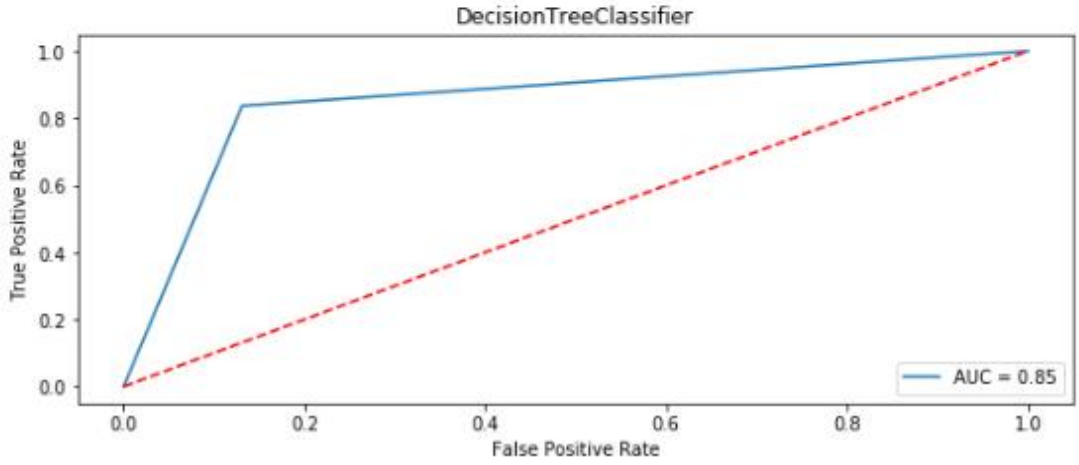
- Visualizations

```
********** LogisticRegression **********

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)

Accuracy_score =  0.9275222730556224

Cross_Val_Score =  0.9259675785863362

roc_auc_score =  0.9275171086577184

classification_report
              precision    recall  f1-score   support

           0       0.95      0.91      0.93      2076
           1       0.91      0.95      0.93      2077

    accuracy                           0.93      4153
   macro avg       0.93      0.93      0.93      4153
weighted avg       0.93      0.93      0.93      4153


[[1881  195]
 [ 106 1971]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```

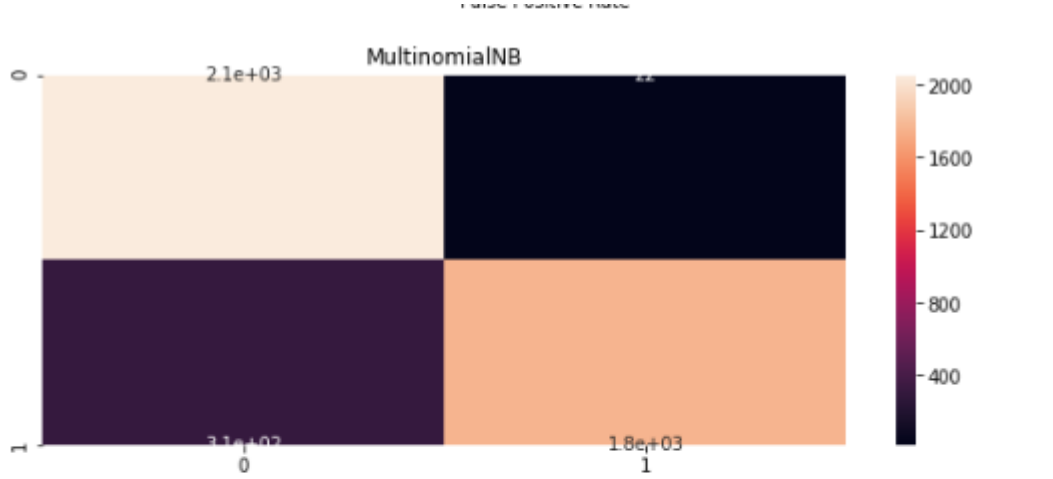


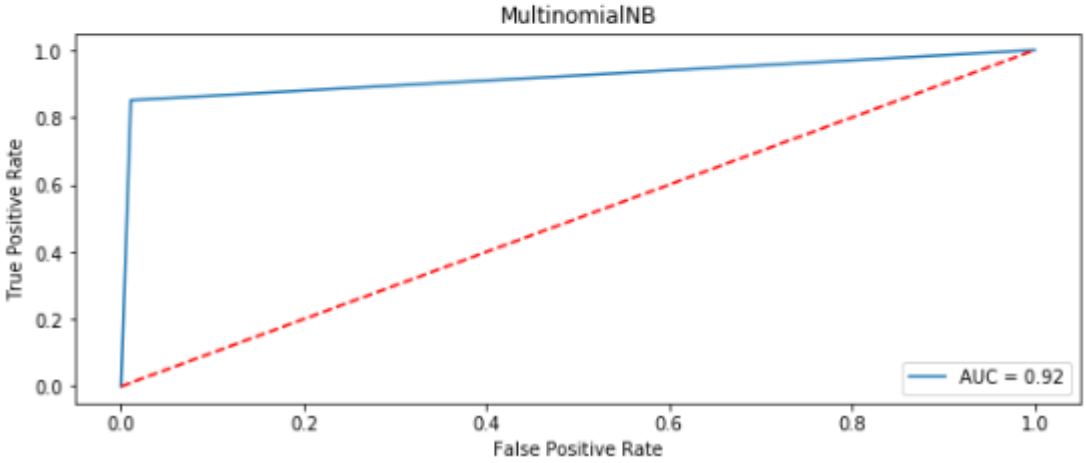



```
********************** DecisionTreeClassifier **********************

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')

Accuracy_score =  0.852877437996629

Cross_Val_Score =  0.861519717426875

roc_auc_score =  0.8528813141081837

classification_report
                precision    recall  f1-score   support

           0       0.84      0.87      0.86      2076
           1       0.86      0.84      0.85      2077

    accuracy                           0.85      4153
   macro avg       0.85      0.85      0.85      4153
weighted avg       0.85      0.85      0.85      4153


[[1804  272]
 [ 339 1738]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```

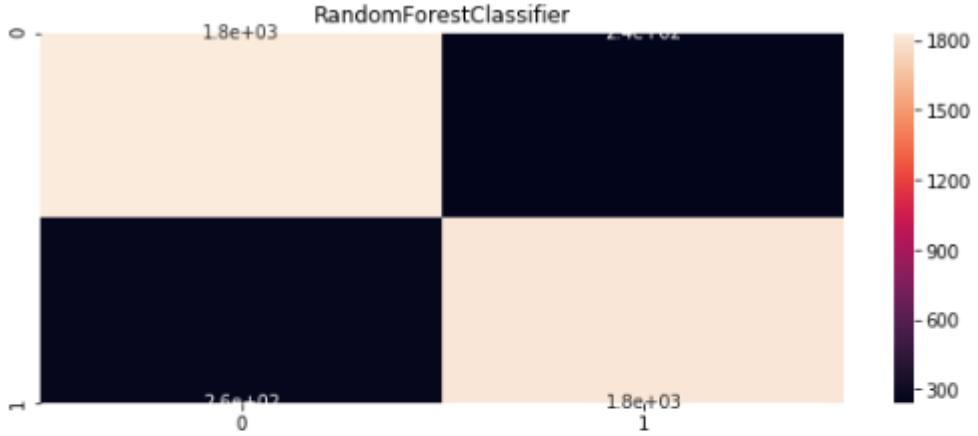


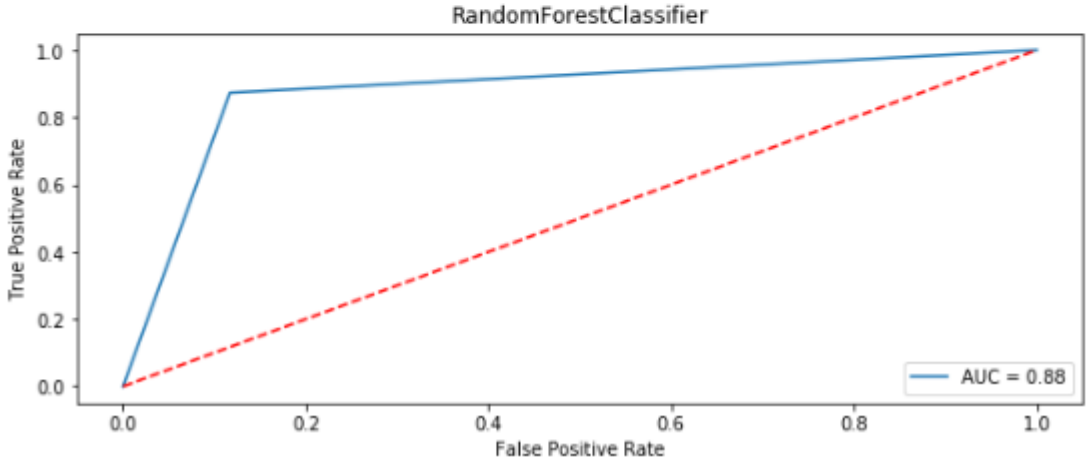

```
********************** MultinomialNB **********************

MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

Accuracy_score =  0.920298579340236

Cross_Val_Score =  0.9221148257105949

roc_auc_score =  0.920315214900697

classification_report
               precision    recall  f1-score   support

           0       0.87      0.99      0.93      2076
           1       0.99      0.85      0.91      2077

    accuracy                           0.92      4153
   macro avg       0.93      0.92      0.92      4153
weighted avg       0.93      0.92      0.92      4153


[[2054   22]
 [ 309 1768]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



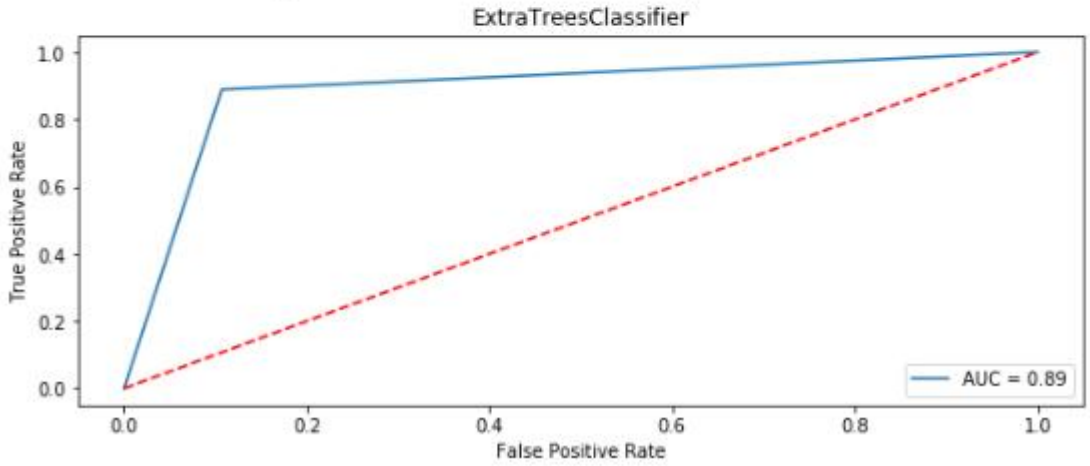



```
********************** RandomForestClassifier **********************

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)

Accuracy_score =  0.8781603660004815

Cross_Val_Score =  0.8884466665814994

roc_auc_score =  0.8781615185307845

classification_report
              precision    recall  f1-score   support

           0       0.87      0.88      0.88      2076
           1       0.88      0.87      0.88      2077

    accuracy                           0.88      4153
   macro avg       0.88      0.88      0.88      4153
weighted avg       0.88      0.88      0.88      4153


[[1833  243]
 [ 263 1814]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```

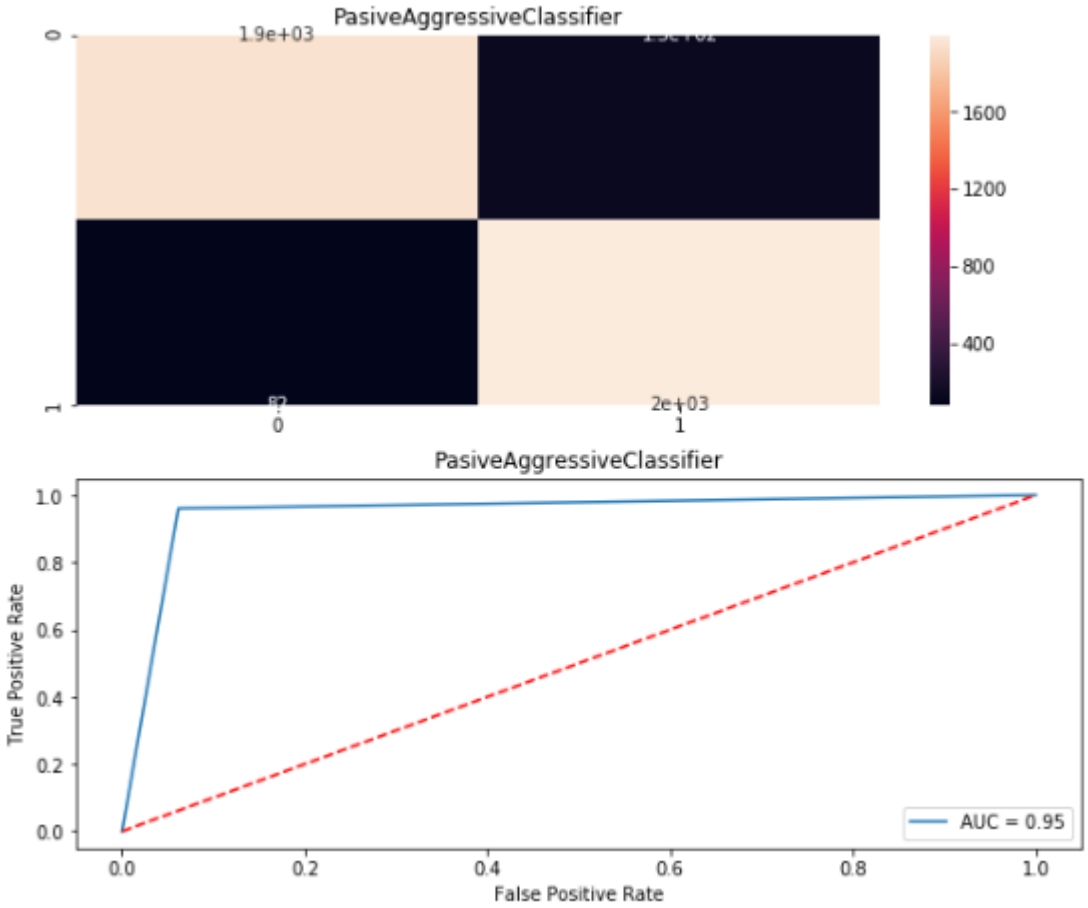





```
********************** ExtraTreesClassifier **********************

ExtraTreesClassifier(bootstrap=False, class_weight=None, criterion='gini',
                     max_depth=None, max_features='auto', max_leaf_nodes=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
                     oob_score=False, random_state=None, verbose=0,
                     warm_start=False)

Accuracy_score =  0.8909222248976644

Cross_Val_Score =  0.8979825235077195

roc_auc_score =  0.8909227403909039

classification_report
               precision    recall  f1-score   support

           0       0.89      0.89      0.89      2076
           1       0.89      0.89      0.89      2077

    accuracy                           0.89      4153
   macro avg       0.89      0.89      0.89      4153
weighted avg       0.89      0.89      0.89      4153


[[1854  222]
 [ 231 1846]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```




```
********************** PasiveAggressiveClassifier **********************

PassiveAggressiveClassifier(C=1.0, average=False, class_weight=None,
                            early_stopping=False, fit_intercept=True,
                            loss='hinge', max_iter=1000, n_iter_no_change=5,
                            n_jobs=None, random_state=None, shuffle=True,
                            tol=0.001, validation_fraction=0.1, verbose=0,
                            warm_start=False)

Accuracy_score =  0.9494341439922948

Cross_Val_Score =  0.9524109294539682

roc_auc_score =  0.9494314739930777

classification_report
               precision    recall  f1-score   support

           0       0.96      0.94      0.95      2076
           1       0.94      0.96      0.95      2077

    accuracy                           0.95      4153
   macro avg       0.95      0.95      0.95      4153
weighted avg       0.95      0.95      0.95      4153


[[1948  128]
 [  82 1995]]

AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



- Interpretation of the Results

  From the above sets of visualizations and matrix, we can see that the Passive Aggressive algorithms performed the best with 94% accuracy score, with cross val score of 95%, and Roc score of 94% and precision score of 96% and f1 score of 95% the max score which was achieved form the dataset provided.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  From the whole evaluation we can see that the maximum number of words in fake news were regarding Trump, and Clinton and we can interpret that it was due to election campaign which was held during US presential election and we know these adverse effects of the voters which were influenced by the fake news and most of the real news had said, trump and president, and fake news which was cleared by trump's campaign, but can hardly see any clarity or real news from the side of Clinton, and due to which the impact we already saw on election results and regarding the election advertisement and news Facebook's CEO Mark Zuckerberg also got extensively question by congress.

- ## Learning Outcomes of the Study in respect of Data Science

  So, from the words frequency chart we can clearly see that most of the news were related to US presential election between Trump and Clinton, and by implementing passive aggressive algorithms we can see that the we have achieved a good score as it calculates the errors and updates its own learning rate which makes our model more reliable.

- ## Limitations of this work and Scope for Future Work

  There were many limitations which training the models as I was trying to train all ensemble algorithms and SVC and KNN but they failed to provide better result and was taking a lot of time more than 72 hours to evaluate due the dataset as it was large.

  Future scope of work will be definitely to implemented hyper tunning the algorithms might give better results, but as per research scholars Passive Aggressive classifier works best for fake news classification.

  However, wanted to try creating training batches as we did for deep learning as it was taking a lot time to train keeping test set completely aside till we train the whole model but due to short time and lack of computational power was unable to implement.