FLIP ROBO

Fake News Project

Submitted By:

Purva Sonsare

ACKNOWLEDGMENT

This Project would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I would like to thank Flip-Robo Technologies Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & my SME of Flip-Robo's Mohd. Kashif for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

INTRODUCTION

Problem Statement:

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

Conceptual Background of the Domain Problem

Fake news is defined as a made-up story with an intention to deceive or to mislead. The rate of production of fake news has increased exponentially. In the past news obtained from newspaper, radio or TV were considered as the best and authentic source of information about the real world and ongoing situations but now everything has changed. In the run of popularity and ill mind set the media houses and social media are spreading fake news. It's becoming harder and harder to say whether a piece of news is real or fabricated.

The effect of fake news can be seen everywhere. The fake news leads to communal disturbance, character assassination, mental trauma, sometimes it is used as a weapon to achieve some illicit plans etc. these are like wild fire which spread too quickly and difficult to control. Which creates difficulty in differentiating between fake news and authentic news.

Review of Literature

You can find many datasets for fake news detection on Kaggle or many other sites. I download these datasets from Kaggle. There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. You have to insert one label column zero for fake news and one for true news. We are combined both datasets using pandas built-in function.

Motivation for the Problem Undertaken

Technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) tools offer great promise for researchers to build systems which could automatically detect fake news. However, detecting fake news is a challenging task to accomplish as it requires models to summarize the news and compare it to the actual news in order to classify it as fake. Moreover, the task of comparing proposed news with the original news itself is a daunting task as it's highly subjective and opinionated.

The goal is to build a prototype to classify the news as fake or not fakes in order to bring awareness and reduce unwanted chaos.

Analytical Problem Framing

Model Building Phase

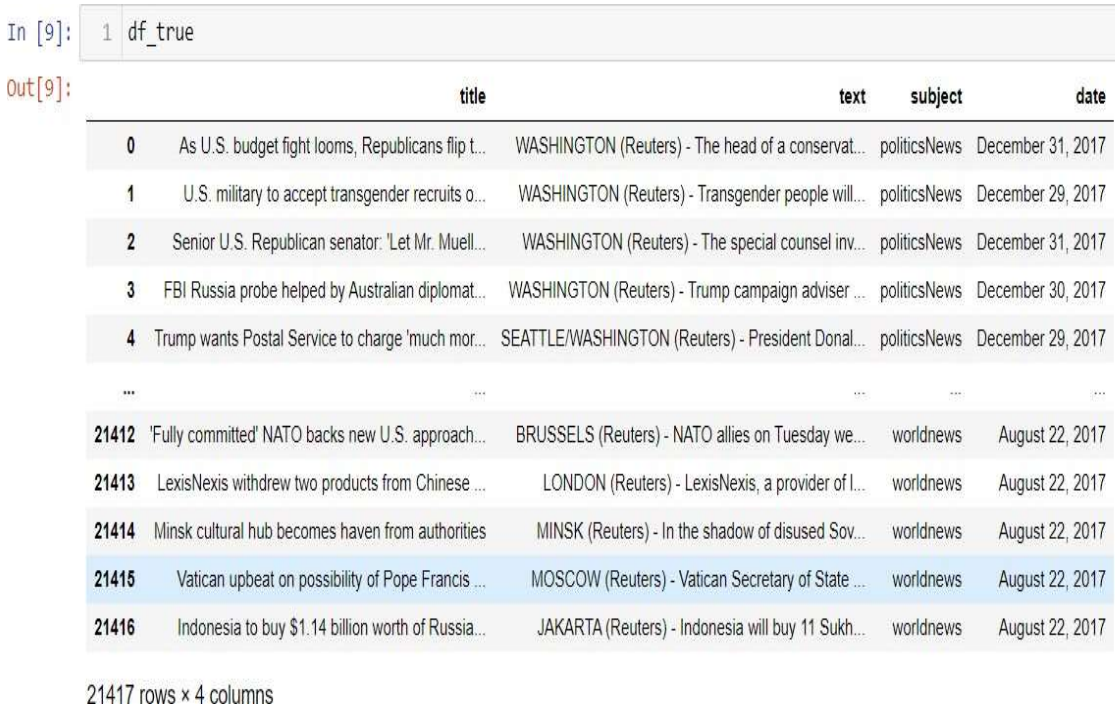
You need to build a machine learning model. Before model building do all data pre-processing steps involving NLP. Try different models with different hyper parameters and select the best model. Follow the complete life cycle of data science. Include all the steps like-

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

Data Sources and their formats

You can find many datasets for fake news detection on Kaggle or many other sites. I download these datasets from Kaggle. There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. You have to insert one label column zero for fake news and one for true news. We are combined both datasets using pandas built-in function.

The data is provided in the CSV file.

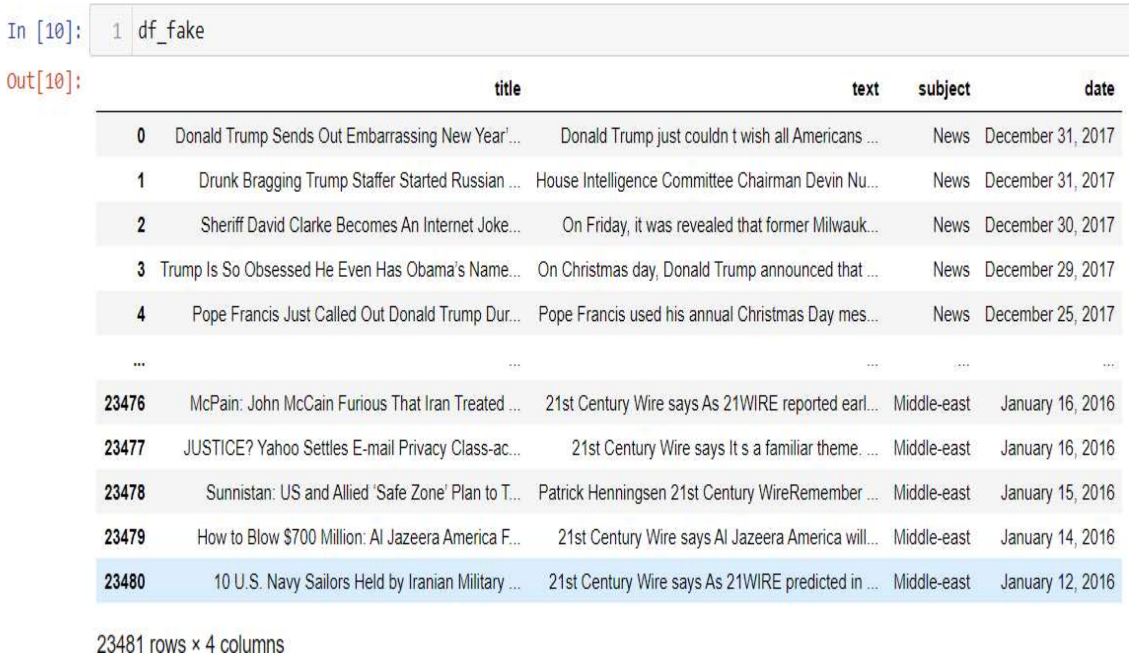


```
In [9]: df_true
```

Out[9]:

	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
...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017

21417 rows × 4 columns



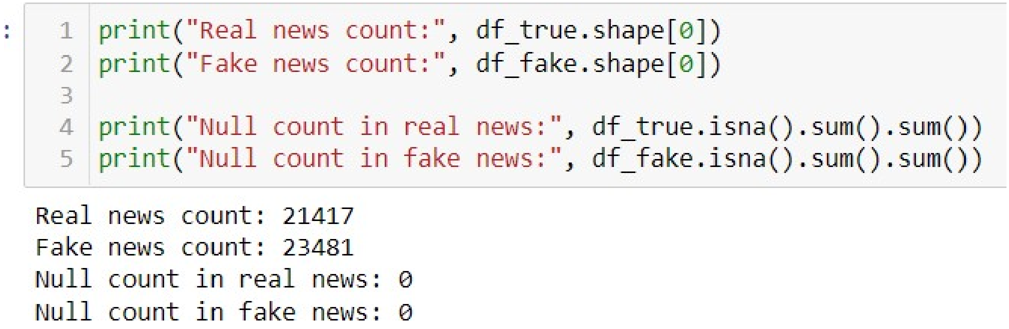
```
In [10]: df_fake
```

Out[10]:

	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
...	...	...	...	...
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016

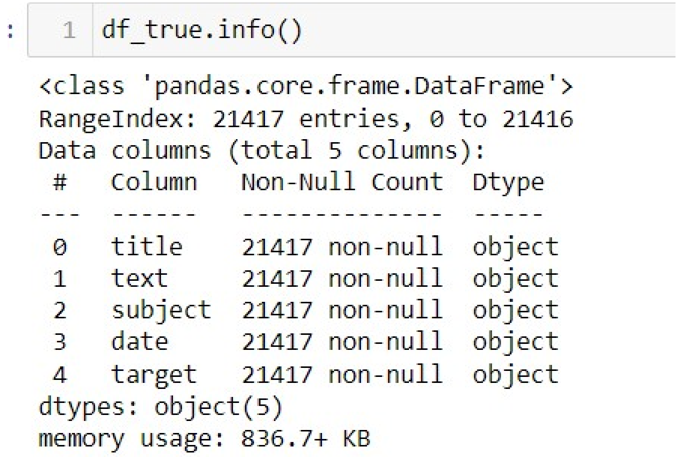
23481 rows × 4 columns

Exploring the Data separately



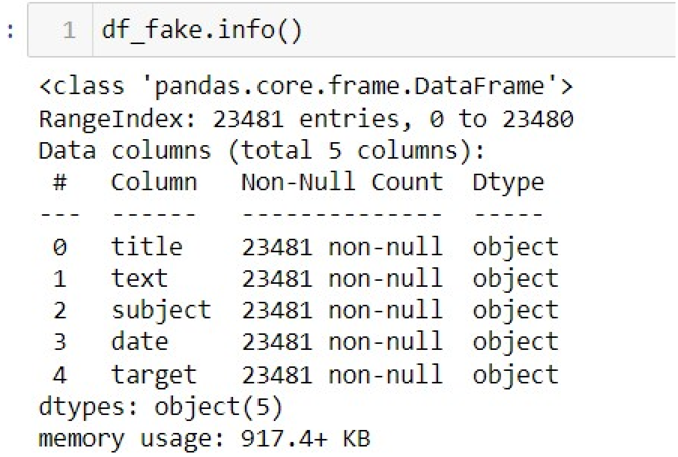
```
print("Real news count:", df_true.shape[0])
print("Fake news count:", df_fake.shape[0])

print("Null count in real news:", df_true.isna().sum().sum())
print("Null count in fake news:", df_fake.isna().sum().sum())
```

```
Real news count: 21417
Fake news count: 23481
Null count in real news: 0
Null count in fake news: 0
```



```
df_true.info()
```

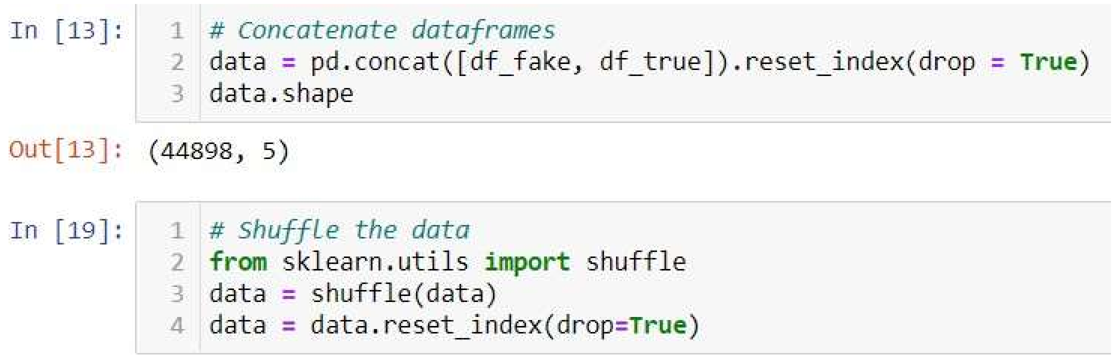
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
 4   target   21417 non-null  object
dtypes: object(5)
memory usage: 836.7+ KB
```



```
df_fake.info()
```

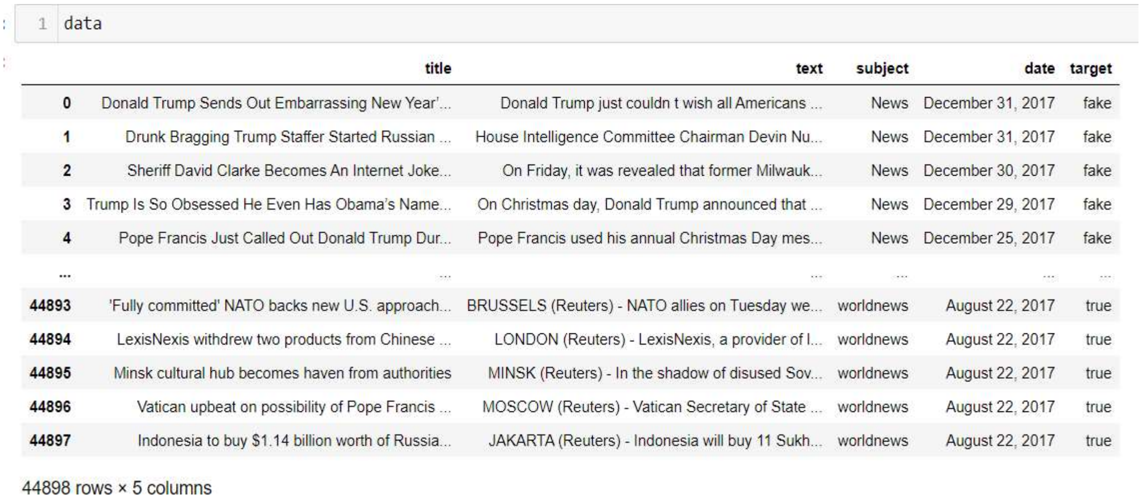
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
 4   target   23481 non-null  object
dtypes: object(5)
memory usage: 917.4+ KB
```

Combining the two Dataset



```
In [13]:  # Concatenate dataframes
          data = pd.concat([df_fake, df_true]).reset_index(drop = True)
          data.shape

Out[13]: (44898, 5)
```

```
In [19]:  # Shuffle the data
          from sklearn.utils import shuffle
          data = shuffle(data)
          data = data.reset_index(drop=True)
```



```
data
```

	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	fake
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	fake
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	fake
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	fake
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	fake
...	...	...	...	...	...
44893	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	true
44894	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	true
44895	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	true
44896	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	true
44897	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	true

44898 rows × 5 columns

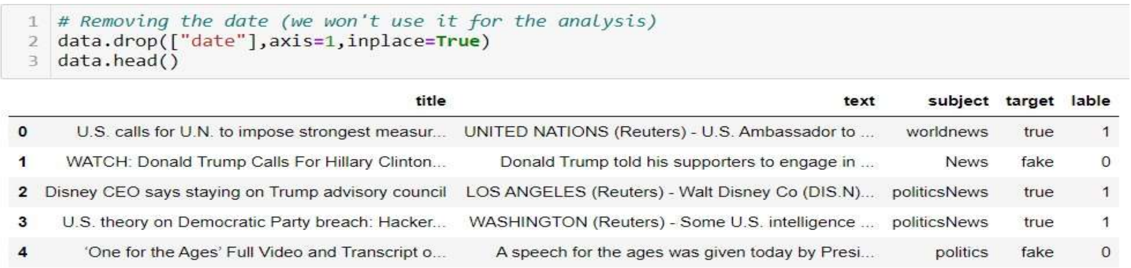
We concatenated the two data into one Data-Frame so that we could easily do any EDA and preprocessing comfortably and will find the training the data will be easy as well.

Data Pre-Processing

Impute missing values

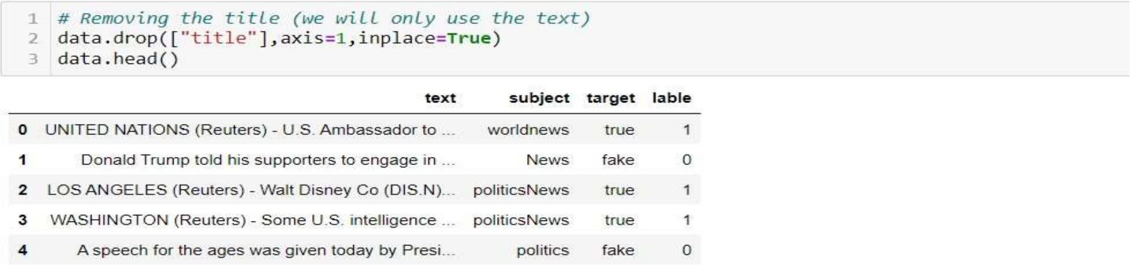
Dropping columns which has missing values.

Removing the date (we won't use it for the analysis), also, Removing the title (we will only use the text).



```python
# Removing the date (we won't use it for the analysis)
data.drop(["date"],axis=1,inplace=True)
data.head()
```

	title	text	subject	target	lable
0	U.S. calls for U.N. to impose strongest measur...	UNITED NATIONS (Reuters) - U.S. Ambassador to ...	worldnews	true	1
1	WATCH: Donald Trump Calls For Hillary Clinton...	Donald Trump told his supporters to engage in ...	News	fake	0
2	Disney CEO says staying on Trump advisory council	LOS ANGELES (Reuters) - Walt Disney Co (DIS.N)...	politicsNews	true	1
3	U.S. theory on Democratic Party breach: Hacker...	WASHINGTON (Reuters) - Some U.S. intelligence ...	politicsNews	true	1
4	'One for the Ages' Full Video and Transcript o...	A speech for the ages was given today by Presi...	politics	fake	0



```python
# Removing the title (we will only use the text)
data.drop(["title"],axis=1,inplace=True)
data.head()
```

	text	subject	target	lable
0	UNITED NATIONS (Reuters) - U.S. Ambassador to ...	worldnews	true	1
1	Donald Trump told his supporters to engage in ...	News	fake	0
2	LOS ANGELES (Reuters) - Walt Disney Co (DIS.N)...	politicsNews	true	1
3	WASHINGTON (Reuters) - Some U.S. intelligence ...	politicsNews	true	1
4	A speech for the ages was given today by Presi...	politics	fake	0

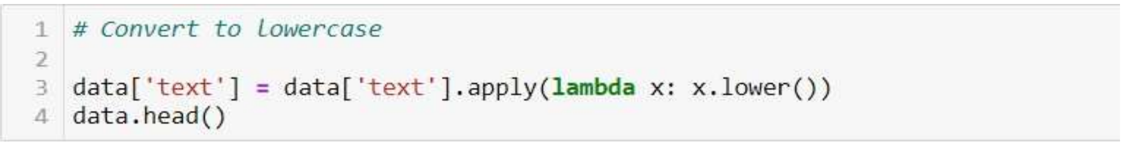
Pre-processing using NLP

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. This data is usually not necessary or helpful when it comes to analysing data because it may hinder the process or provide inaccurate results.

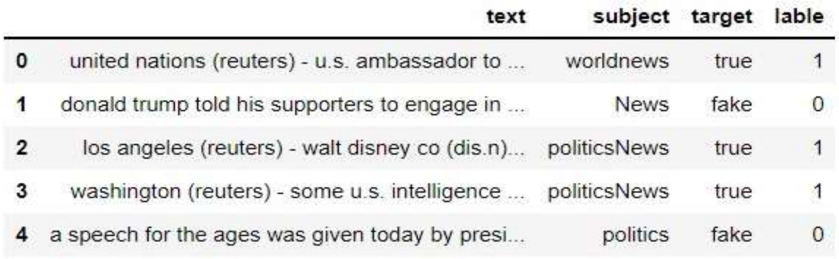
Before cleaning the data, a new column is created named 'length_before_cleaning' which shows the total length of the news respectively before cleaning the text.
The following steps were taken in order to clean the text:

- Transform the text into lower case.
- Replaced the email addresses with the text 'emailaddress'
- Replaced the URLs with the text 'webaddress'
- Removed the HTML tags
- Removed the numbers
- Removed extra newlines
- Removed the punctuations
- Removed the unwanted white spaces
- Removed the remaining tokens that are not alphabetic
- Removed the stop words

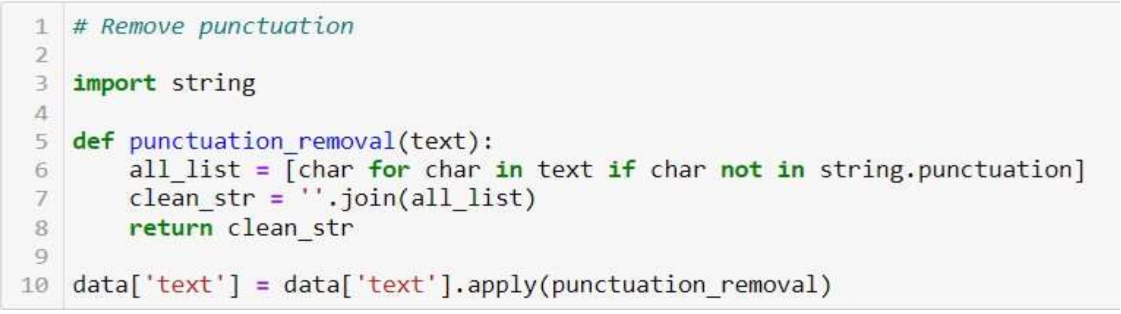
Convert to lowercase and Remove punctuation



```python
# Convert to lowercase

data['text'] = data['text'].apply(lambda x: x.lower())
data.head()
```

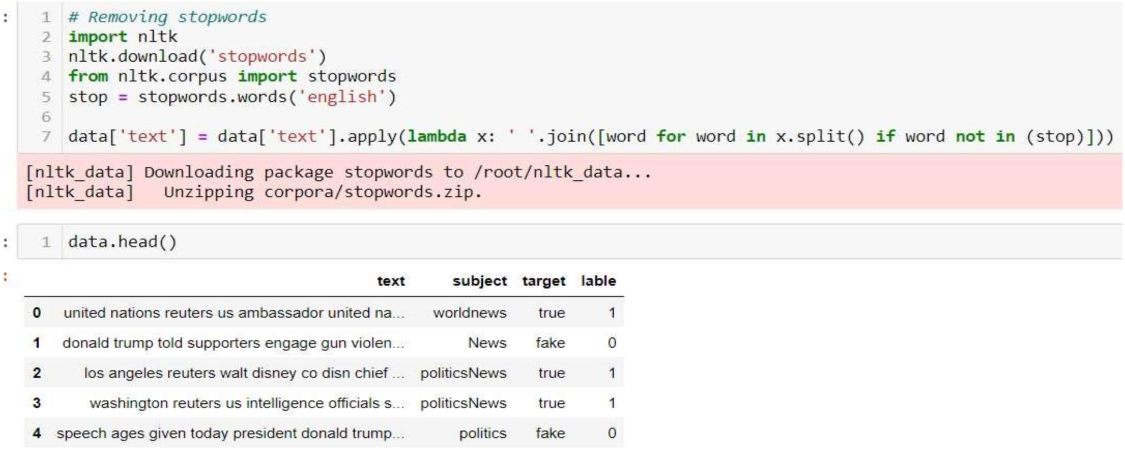


	text	subject	target	lable
0	united nations (reuters) - u.s. ambassador to ...	worldnews	true	1
1	donald trump told his supporters to engage in ...	News	fake	0
2	los angeles (reuters) - walt disney co (dis.n)...	politicsNews	true	1
3	washington (reuters) - some u.s. intelligence ...	politicsNews	true	1
4	a speech for the ages was given today by presi...	politics	fake	0



```python
# Remove punctuation

import string

def punctuation_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
    clean_str = ''.join(all_list)
    return clean_str

data['text'] = data['text'].apply(punctuation_removal)
```

Removing Stop-Words



```python
# Removing stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')

data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```python
data.head()
```

	text	subject	target	lable
0	united nations reuters us ambassador united na...	worldnews	true	1
1	donald trump told supporters engage gun violen...	News	fake	0
2	los angeles reuters walt disney co disn chief ...	politicsNews	true	1
3	washington reuters us intelligence officials s...	politicsNews	true	1
4	speech ages given today president donald trump...	politics	fake	0

Tokenization

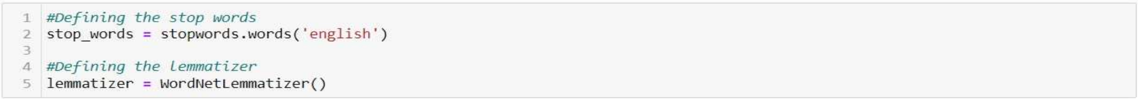
Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis.
After cleaning the text, each comment i.e., the corpus is split into words. Thus, the text is tokenized into words using word_tokenize().

Lemmatization

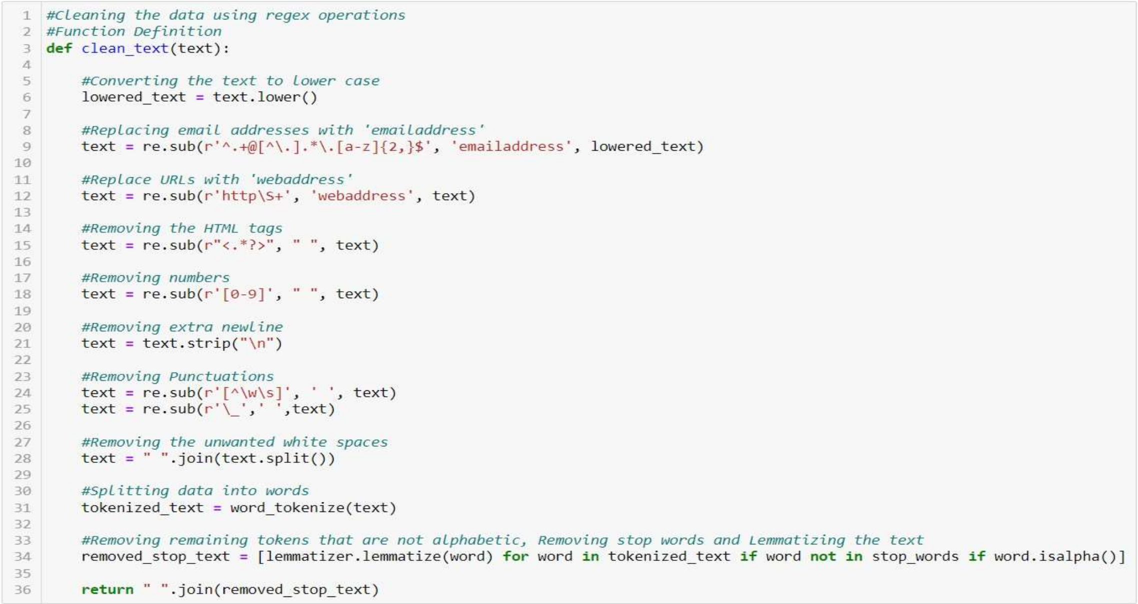
Lemmatization in NLTK refers to the morphological analysis of words, which aims to remove inflectional endings. It helps in returning the base or dictionary form of a word known as the lemma. The NLTK Lemmatization method is based on WorldNet's built-in morph function.Thus, the words are lemmatized using WordNetLemmatizer() after importing the necessary library to perform the same and then creating the instance for it.

All the text cleaning or the above steps are performed by defining a function and applying the same using apply() to the ‘News’ column of the dataset.
Below is the code shown:

```
#Defining the stop words
stop_words = stopwords.words('english')

#Defining the lemmatizer
lemmatizer = WordNetLemmatizer()
```



```python
#Defining the stop words
stop_words = stopwords.words('english')

#Defining the Lemmatizer
lemmatizer = WordNetLemmatizer()
```

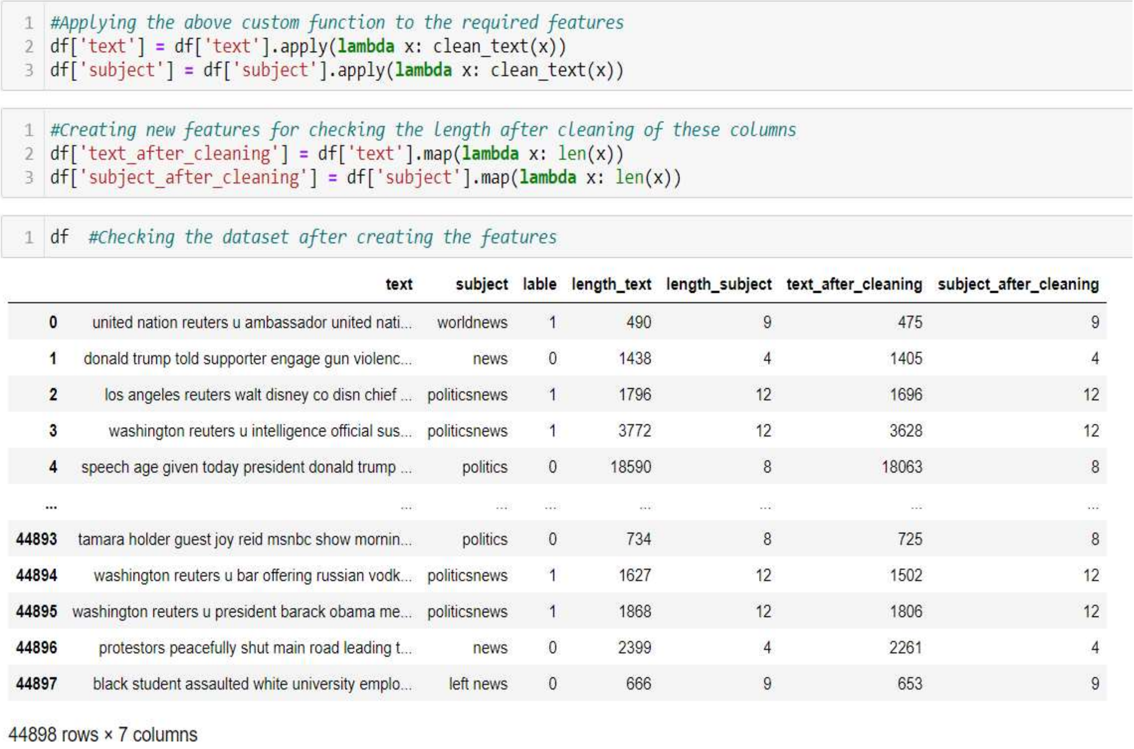


```python
#Cleaning the data using regex operations
#Function Definition
def clean_text(text):

    #Converting the text to lower case
    lowered_text = text.lower()

    #Replacing email addresses with 'emailaddress'
    text = re.sub(r'^.+@[^\.].*\.[a-z]{2,}$', 'emailaddress', lowered_text)

    #Replace URLs with 'webaddress'
    text = re.sub(r'http\S+', 'webaddress', text)

    #Removing the HTML tags
    text = re.sub(r"<.*?>", " ", text)

    #Removing numbers
    text = re.sub(r'[0-9]', " ", text)

    #Removing extra newline
    text = text.strip("\n")

    #Removing Punctuations
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\_',' ',text)

    #Removing the unwanted white spaces
    text = " ".join(text.split())

    #Splitting data into words
    tokenized_text = word_tokenize(text)

    #Removing remaining tokens that are not alphabetic, Removing stop words and Lemmatizing the text
    removed_stop_text = [lemmatizer.lemmatize(word) for word in tokenized_text if word not in stop_words if word.isalpha()]

    return " ".join(removed_stop_text)
```

We also created new features for comparing the original length before cleaning and the new length after cleaning.

We can see that the new length features are created and then added to the dataset. Now, we will calculate the total words removed in all the columns.

We can observe that more number of unwanted words were removed from the dataset and it was done by using regex operations and other NLP techniques.



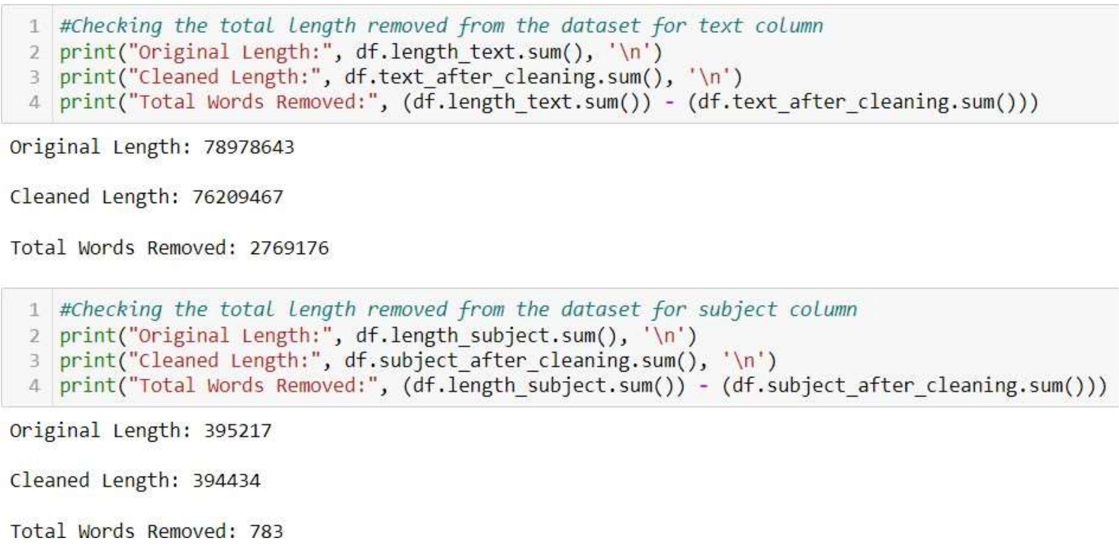
```
#Applying the above custom function to the required features
df['text'] = df['text'].apply(lambda x: clean_text(x))
df['subject'] = df['subject'].apply(lambda x: clean_text(x))
```

```
#Creating new features for checking the length after cleaning of these columns
df['text_after_cleaning'] = df['text'].map(lambda x: len(x))
df['subject_after_cleaning'] = df['subject'].map(lambda x: len(x))
```

```
df  #Checking the dataset after creating the features
```

	text	subject	lable	length_text	length_subject	text_after_cleaning	subject_after_cleaning
0	united nation reuters u ambassador united nati...	worldnews	1	490	9	475	9
1	donald trump told supporter engage gun violenc...	news	0	1438	4	1405	4
2	los angeles reuters walt disney co disn chief ...	politicsnews	1	1796	12	1696	12
3	washington reuters u intelligence official sus...	politicsnews	1	3772	12	3628	12
4	speech age given today president donald trump ...	politics	0	18590	8	18063	8
...	...	...	...	...	...	...	...
44893	tamara holder guest joy reid msnbc show mornin...	politics	0	734	8	725	8
44894	washington reuters u bar offering russian vodk...	politicsnews	1	1627	12	1502	12
44895	washington reuters u president barack obama me...	politicsnews	1	1868	12	1806	12
44896	protestors peacefully shut main road leading t...	news	0	2399	4	2261	4
44897	black student assaulted white university emplo...	left news	0	666	9	653	9

44898 rows × 7 columns



```
#Checking the total length removed from the dataset for text column
print("Original Length:", df.length_text.sum(), '\n')
print("Cleaned Length:", df.text_after_cleaning.sum(), '\n')
print("Total Words Removed:", (df.length_text.sum()) - (df.text_after_cleaning.sum()))
```

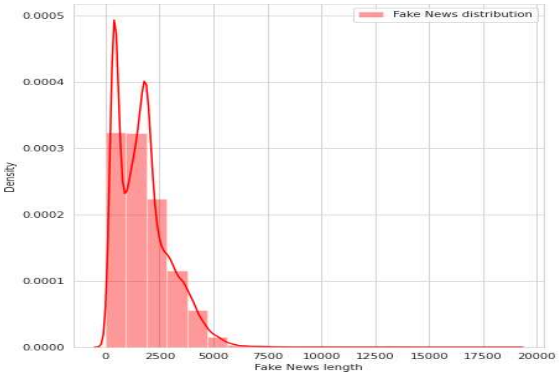
```
Original Length: 78978643

Cleaned Length: 76209467

Total Words Removed: 2769176
```

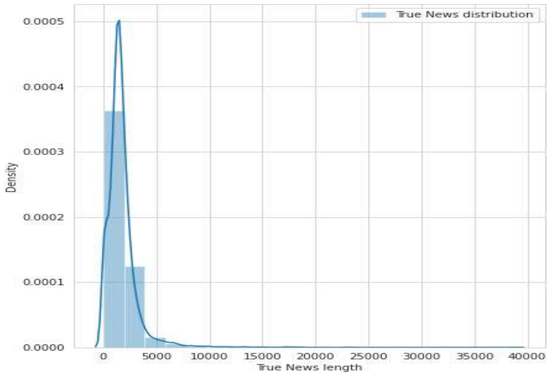
```
#Checking the total length removed from the dataset for subject column
print("Original Length:", df.length_subject.sum(), '\n')
print("Cleaned Length:", df.subject_after_cleaning.sum(), '\n')
print("Total Words Removed:", (df.length_subject.sum()) - (df.subject_after_cleaning.sum()))
```

```
Original Length: 395217

Cleaned Length: 394434

Total Words Removed: 783
```

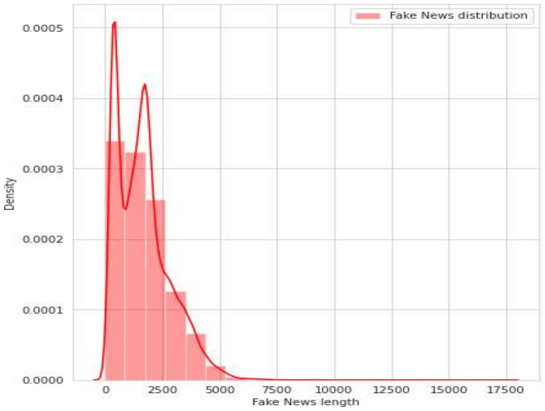
Plotting features before and after cleaning the data

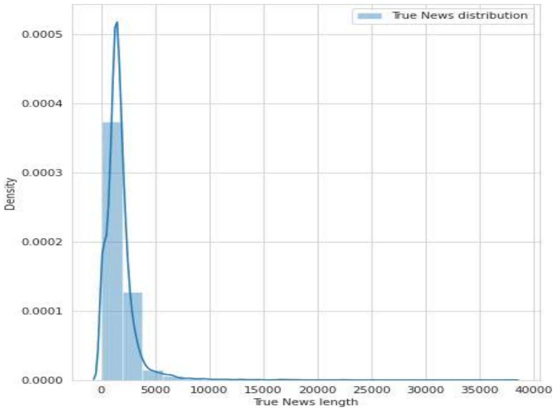
Before cleaning





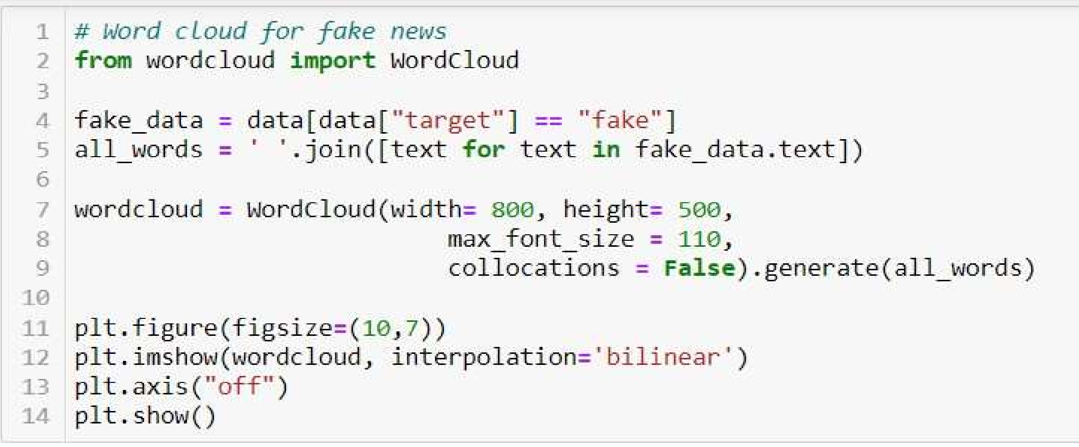
After cleaning



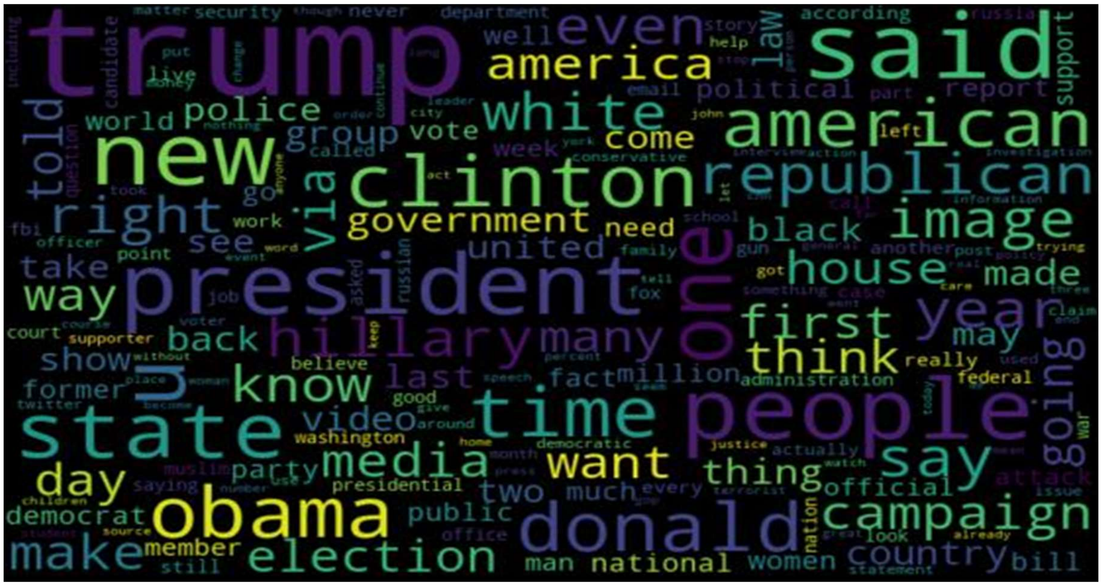


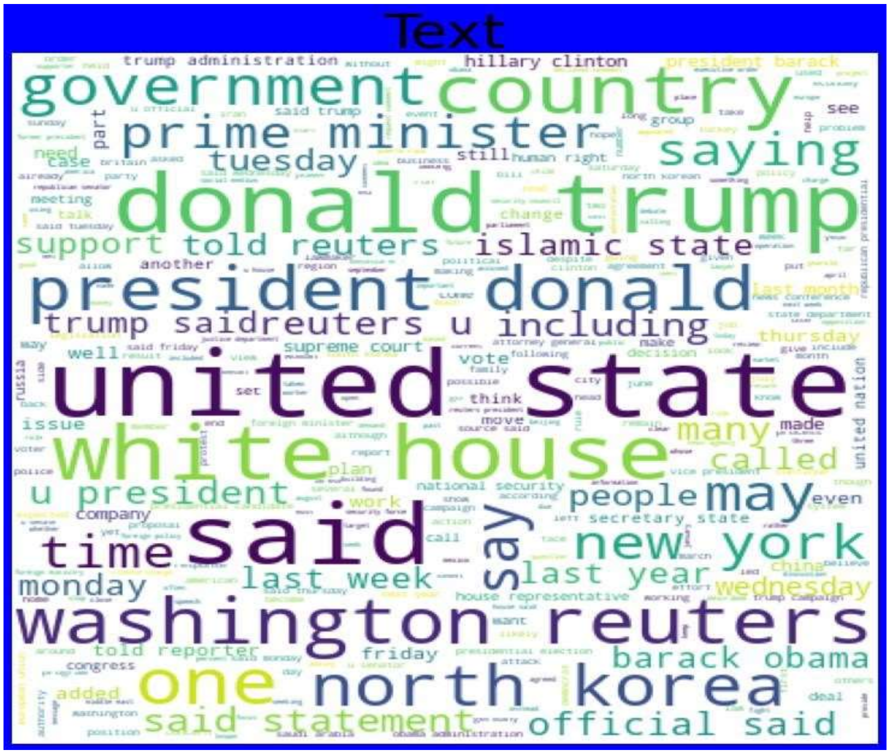
Displaying the Wordcloud

For Fake News



```python
# Word cloud for fake news
from wordcloud import WordCloud

fake_data = data[data["target"] == "fake"]
all_words = ' '.join([text for text in fake_data.text])

wordcloud = WordCloud(width= 800, height= 500,
                          max_font_size = 110,
                          collocations = False).generate(all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

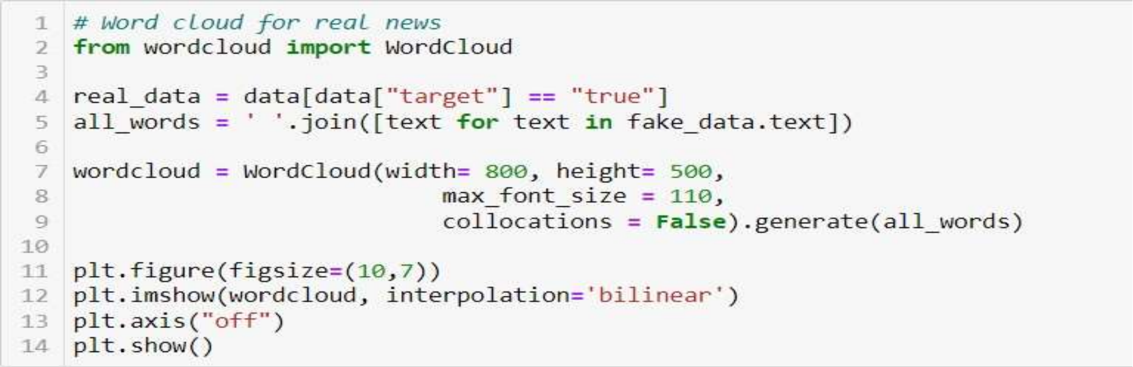



Text

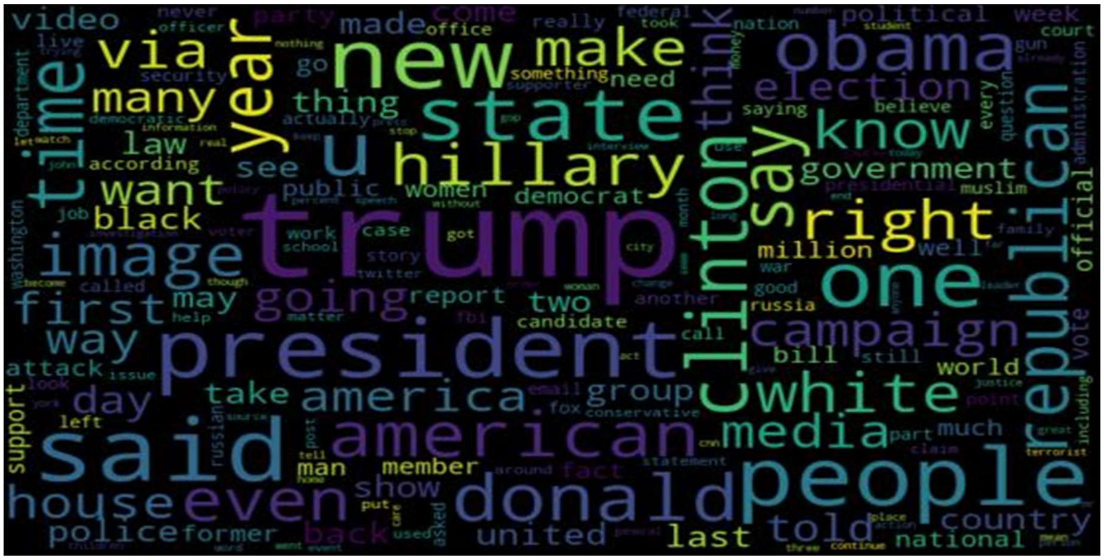
Subject

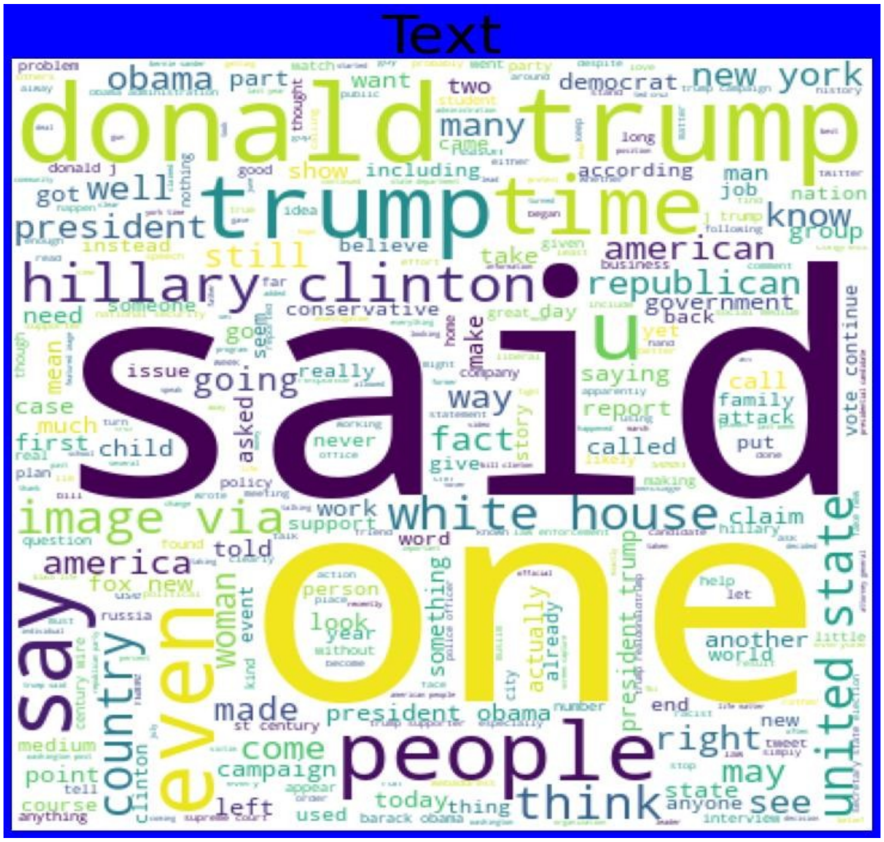
politicsnews
worldnews

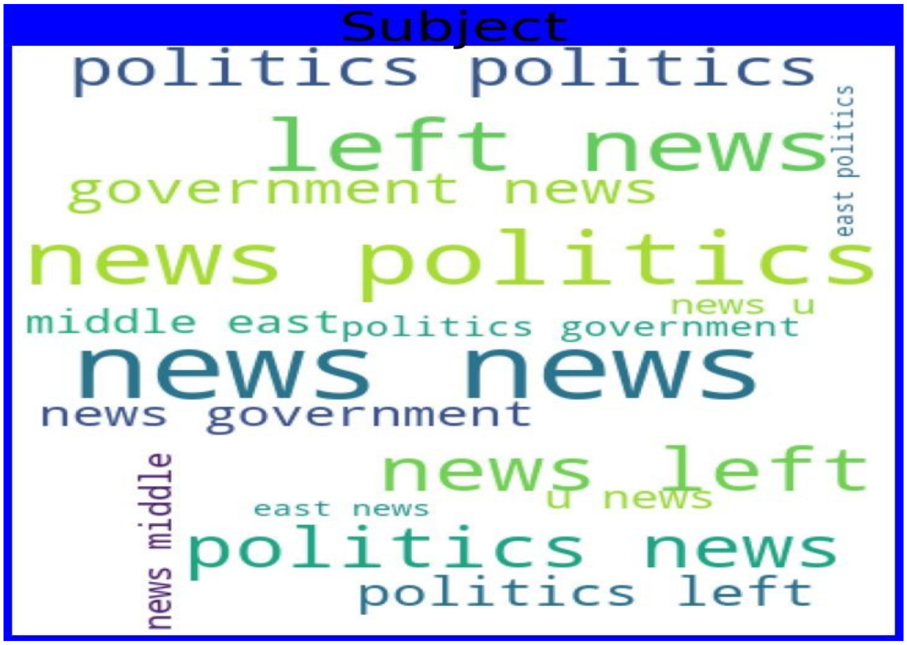
For True News



```
# Word cloud for real news
from wordcloud import WordCloud

real_data = data[data["target"] == "true"]
all_words = ' '.join([text for text in fake_data.text])

wordcloud = WordCloud(width= 800, height= 500,
                          max_font_size = 110,
                          collocations = False).generate(all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```




Text


Subject

Hardware and Software Requirements and Tools Used

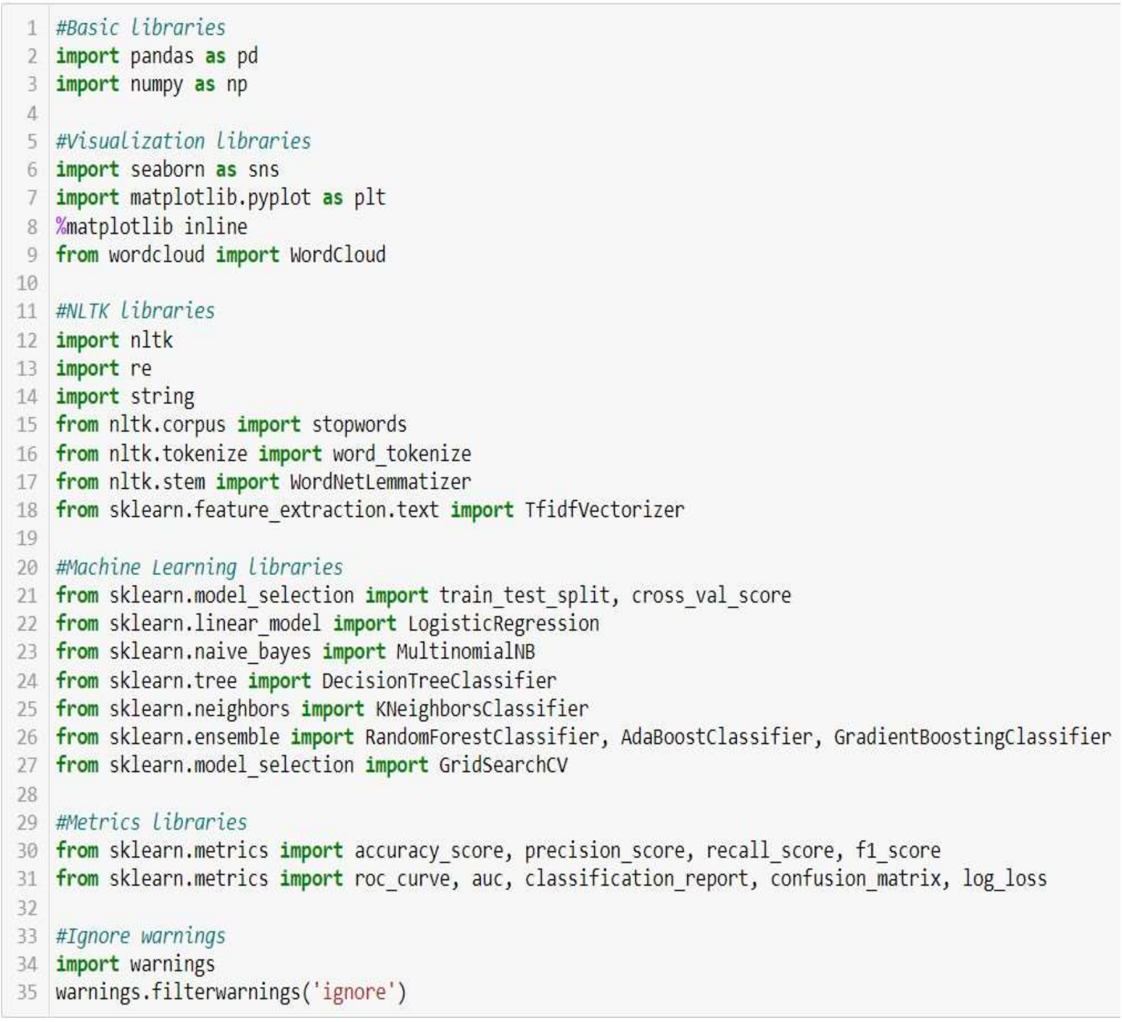
The General Hardware used for this project is :-

8 GB RAM

512GB SSD

Intel i5 processor

For doing this project, the hardware used is a laptop with high end specification and a stable internet connection. While coming to software part, I had used anaconda navigator and in that I have used **Jupyter notebook** to do my python programming and analysis.
For using an CSV file, Microsoft excel is needed. In Jupyter notebook, I had used lots of python libraries to carry out this project and I have mentioned below with proper justification:



```python
#Basic libraries
import pandas as pd
import numpy as np

#Visualization libraries
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from wordcloud import WordCloud

#NLTK libraries
import nltk
import re
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer

#Machine Learning libraries
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV

#Metrics libraries
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import roc_curve, auc, classification_report, confusion_matrix, log_loss

#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

Model/s Development and Evaluation

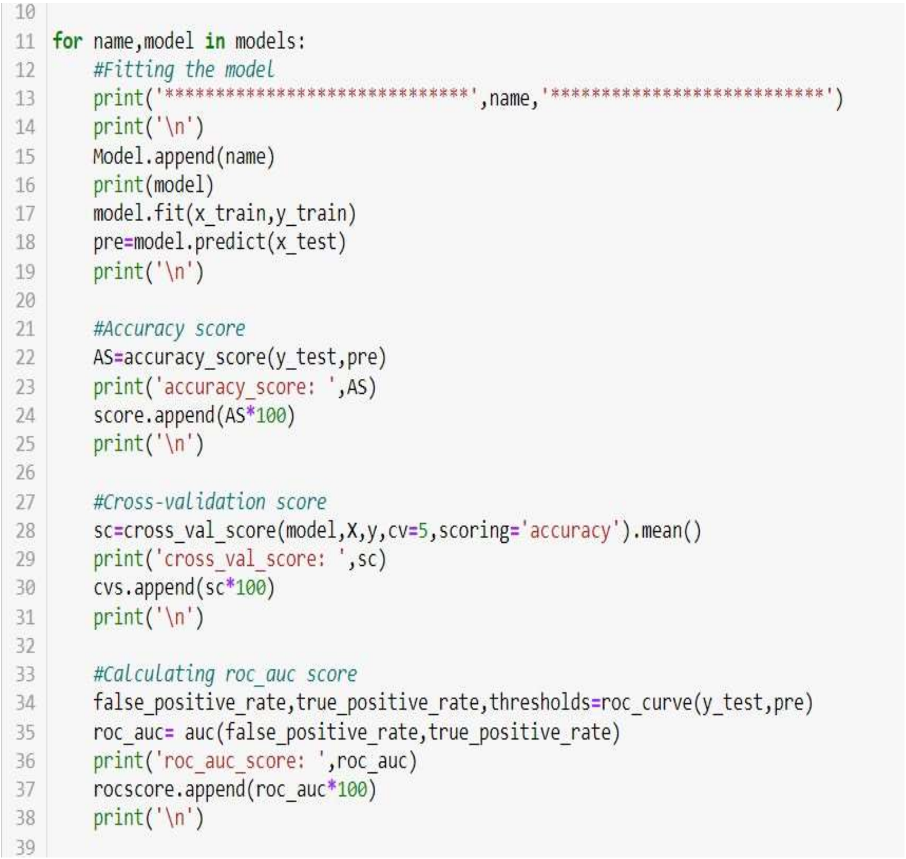
Testing of Identified Approaches (Algorithms)

1) from sklearn.neighbors import KNeighborsClassifier
2) from sklearn.linear_model import LogisticRegression
3) from sklearn.tree import DecisionTreeClassifier
4) from sklearn.naive_bayes import GaussianNB
5) from sklearn.ensemble import RandomForestClassifier
6) from sklearn.preprocessing import StandardScaler
7) from sklearn.metrics import
8) classification_report , confusion_matrix,accuracy_score,roc_curve,auc

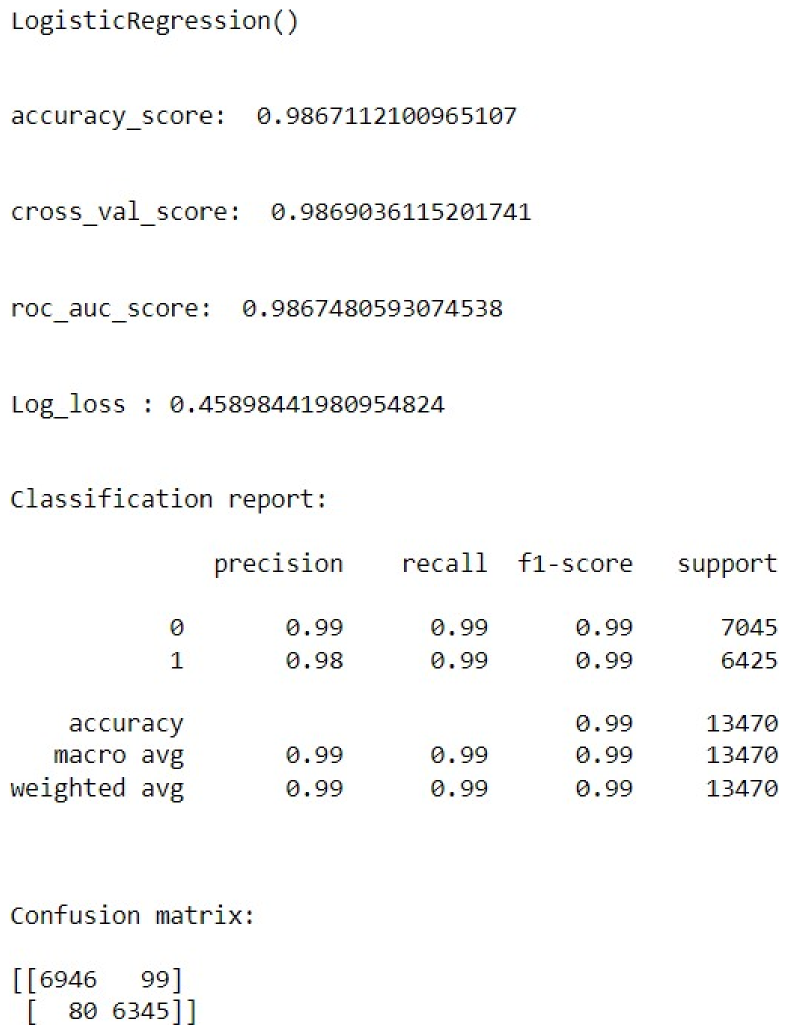
Run and Evaluate selected models

Key Metrics for success in solving problem under consideration

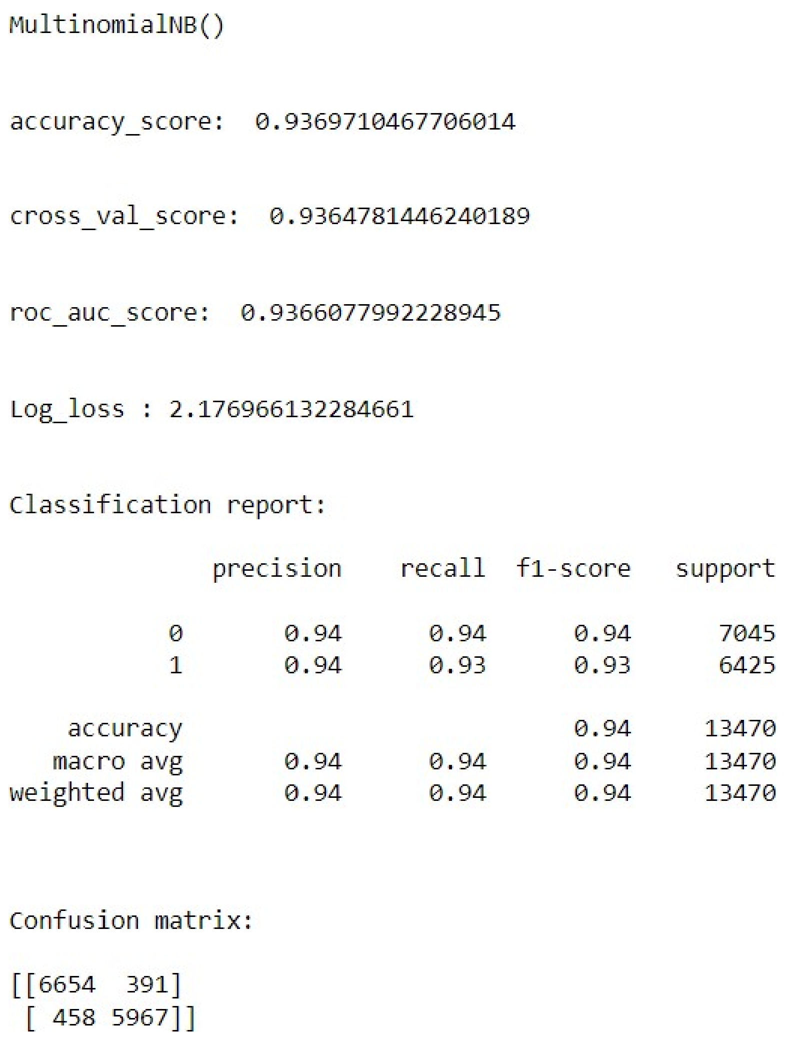
We can observe that I imported the metrics to find the accuracy score, roc_auc_curve, confusion_matrix, classification_report, in order to interpret the models output. Then I also selected the model to find the cross_validation_score and cross validation prediction.



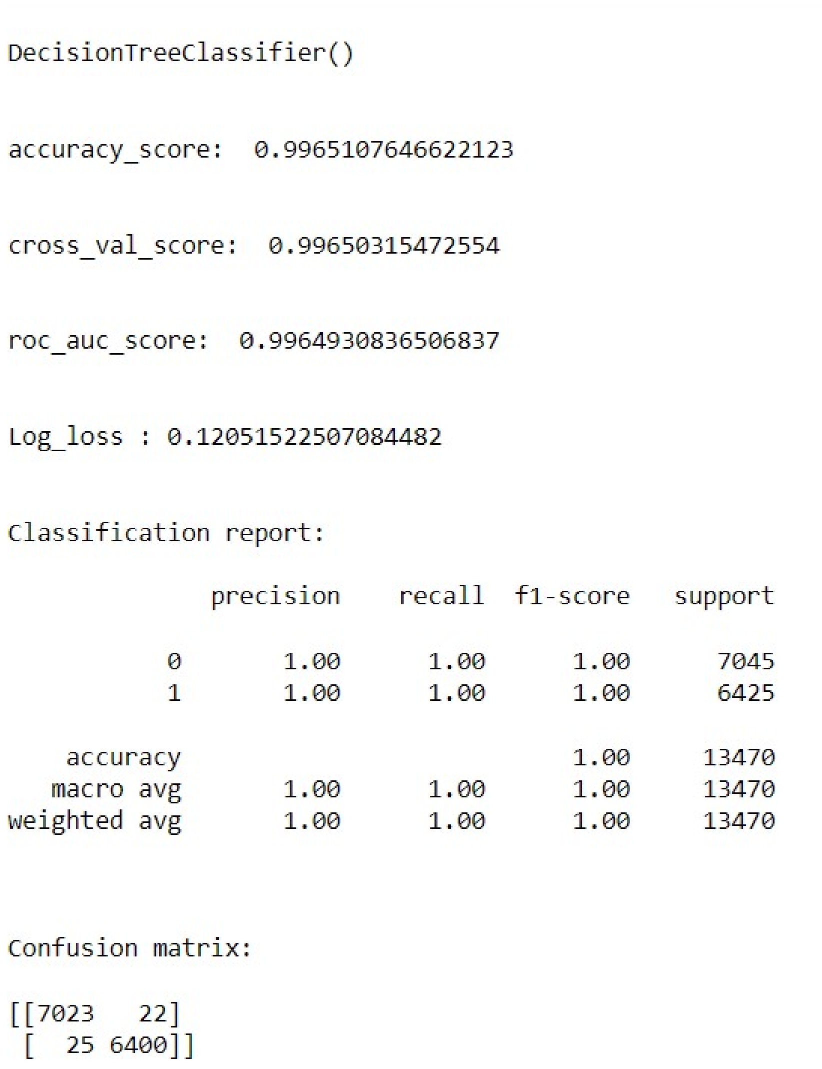
```

for name,model in models:
    #Fitting the model
    print('******************************',name,'***************************')
    print('\n')
    Model.append(name)
    print(model)
    model.fit(x_train,y_train)
    pre=model.predict(x_test)
    print('\n')

    #Accuracy score
    AS=accuracy_score(y_test,pre)
    print('accuracy_score: ',AS)
    score.append(AS*100)
    print('\n')

    #Cross-validation score
    sc=cross_val_score(model,X,y,cv=5,scoring='accuracy').mean()
    print('cross_val_score: ',sc)
    cvs.append(sc*100)
    print('\n')

    #Calculating roc_auc score
    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')

```



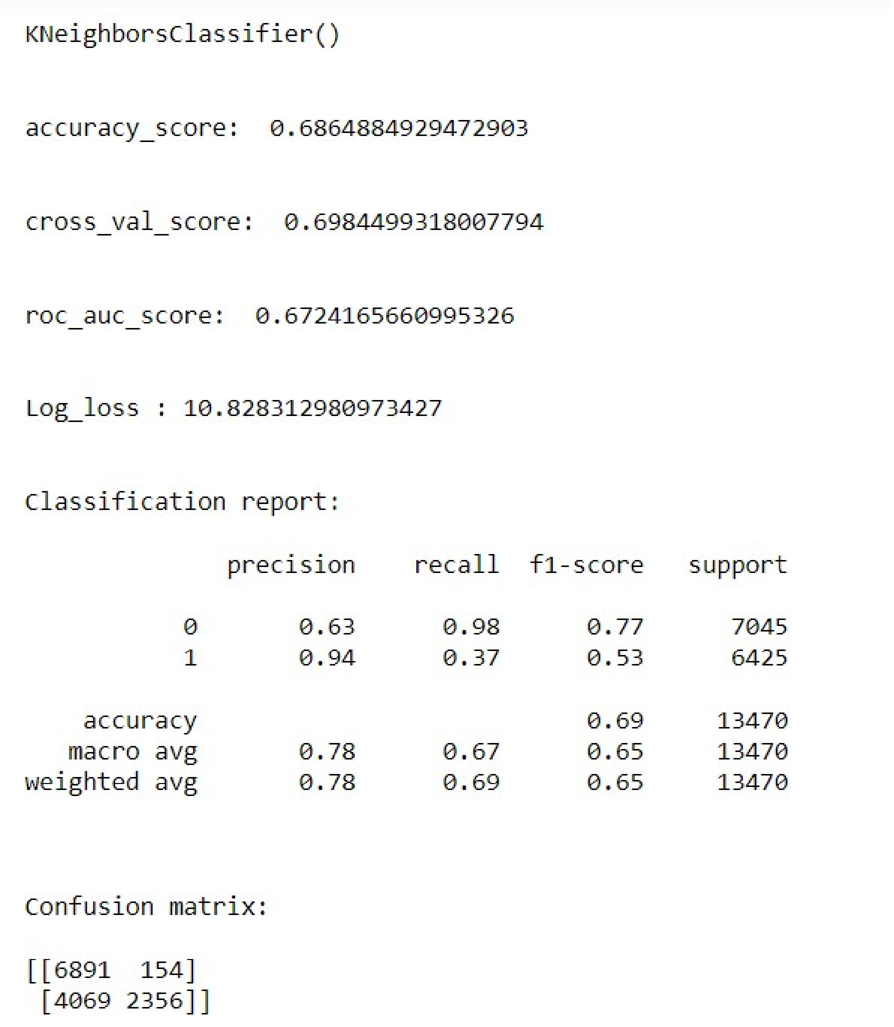
```
LogisticRegression()


accuracy_score:  0.9867112100965107


cross_val_score:  0.9869036115201741


roc_auc_score:  0.9867480593074538


Log_loss : 0.45898441980954824


Classification report:

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      7045
           1       0.98      0.99      0.99      6425

    accuracy                           0.99     13470
   macro avg       0.99      0.99      0.99     13470
weighted avg       0.99      0.99      0.99     13470



Confusion matrix:

[[6946   99]
 [  80 6345]]
```



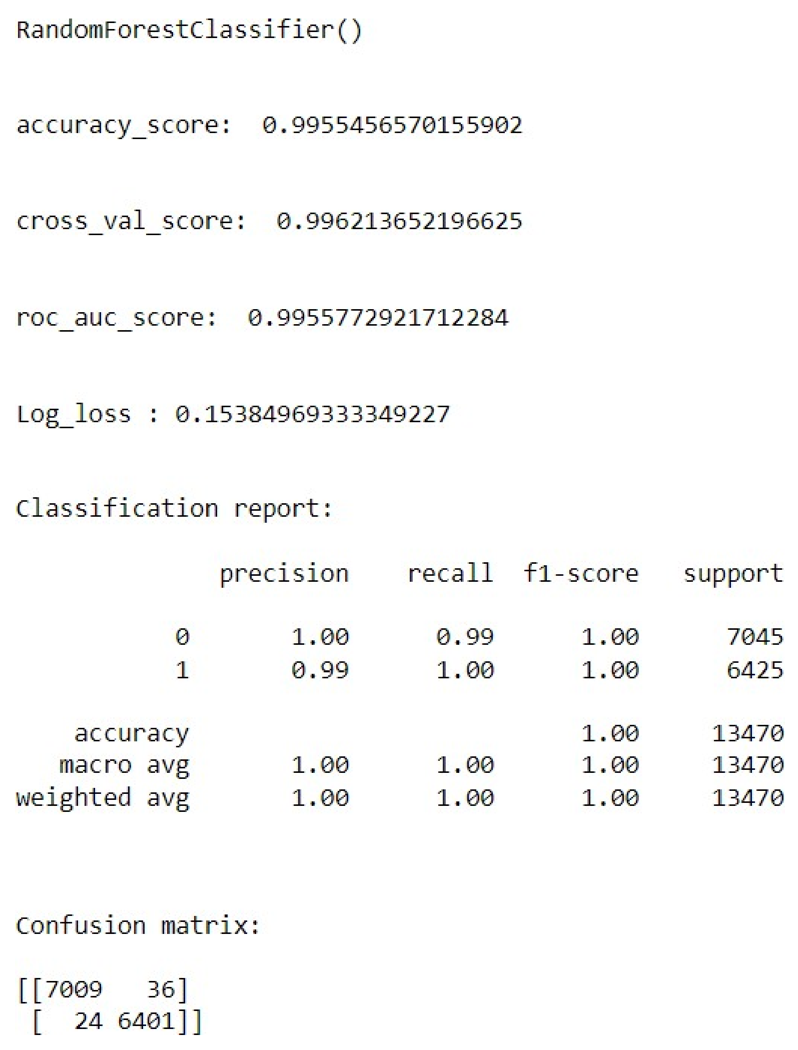
```
MultinomialNB()


accuracy_score:  0.9369710467706014


cross_val_score:  0.9364781446240189


roc_auc_score:  0.9366077992228945


Log_loss : 2.176966132284661


Classification report:

              precision    recall  f1-score   support

           0       0.94      0.94      0.94      7045
           1       0.94      0.93      0.93      6425

    accuracy                           0.94     13470
   macro avg       0.94      0.94      0.94     13470
weighted avg       0.94      0.94      0.94     13470



Confusion matrix:

[[6654  391]
 [ 458 5967]]
```



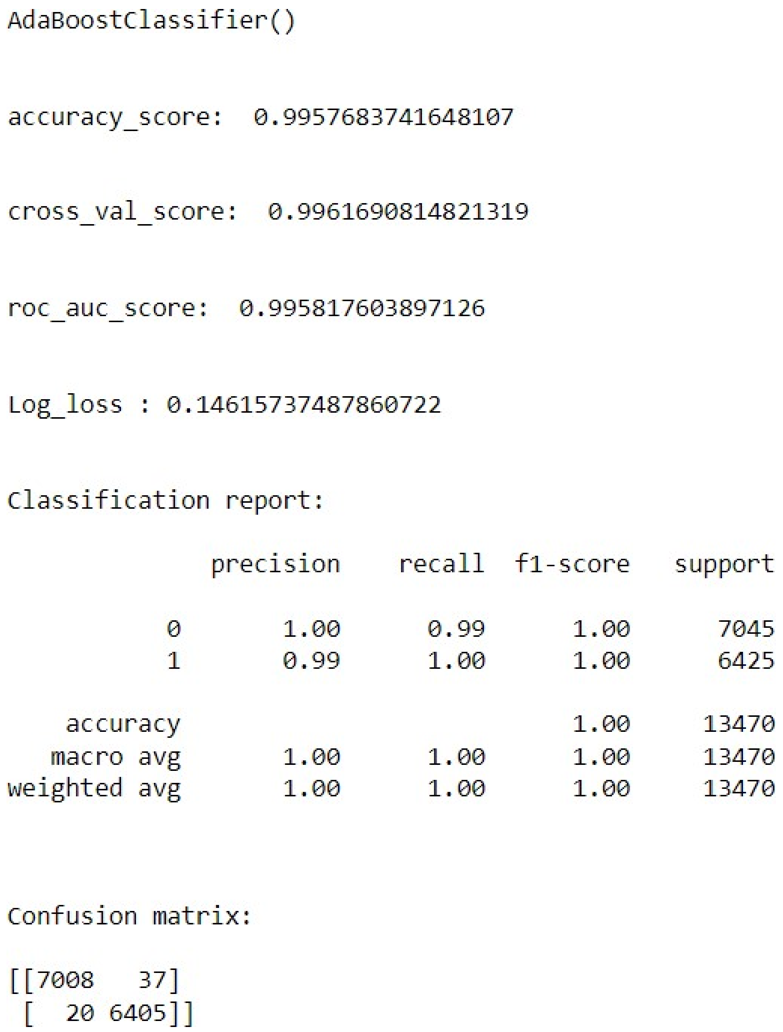
```
DecisionTreeClassifier()


accuracy_score:  0.9965107646622123


cross_val_score:  0.99650315472554


roc_auc_score:  0.9964930836506837


Log_loss : 0.12051522507084482


Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7045
           1       1.00      1.00      1.00      6425

    accuracy                           1.00     13470
   macro avg       1.00      1.00      1.00     13470
weighted avg       1.00      1.00      1.00     13470



Confusion matrix:

[[7023   22]
 [  25 6400]]
```



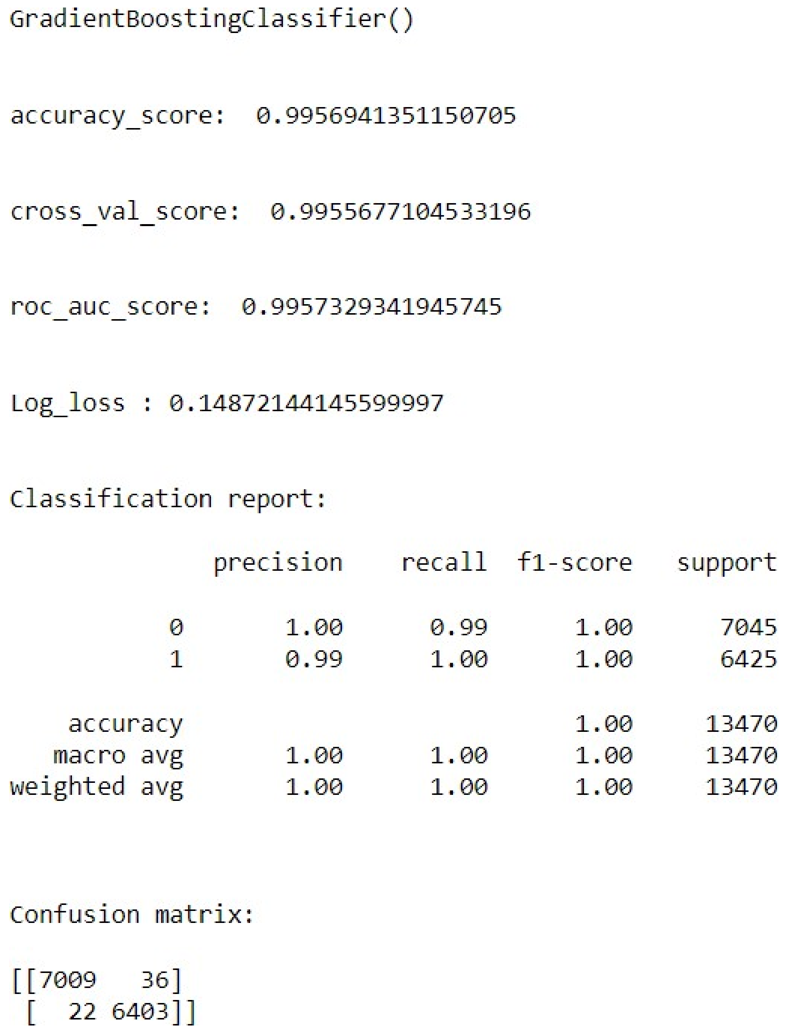
```
KNeighborsClassifier()


accuracy_score:  0.6864884929472903


cross_val_score:  0.6984499318007794


roc_auc_score:  0.6724165660995326


Log_loss : 10.828312980973427


Classification report:
              precision    recall  f1-score   support

           0       0.63      0.98      0.77      7045
           1       0.94      0.37      0.53      6425

    accuracy                           0.69     13470
   macro avg       0.78      0.67      0.65     13470
weighted avg       0.78      0.69      0.65     13470



Confusion matrix:

[[6891  154]
 [4069 2356]]
```



```
RandomForestClassifier()


accuracy_score:  0.9955456570155902


cross_val_score:  0.996213652196625


roc_auc_score:  0.9955772921712284


Log_loss : 0.15384969333349227


Classification report:

              precision    recall  f1-score   support

           0       1.00      0.99      1.00      7045
           1       0.99      1.00      1.00      6425

    accuracy                           1.00     13470
   macro avg       1.00      1.00      1.00     13470
weighted avg       1.00      1.00      1.00     13470



Confusion matrix:

[[7009   36]
 [  24 6401]]
```

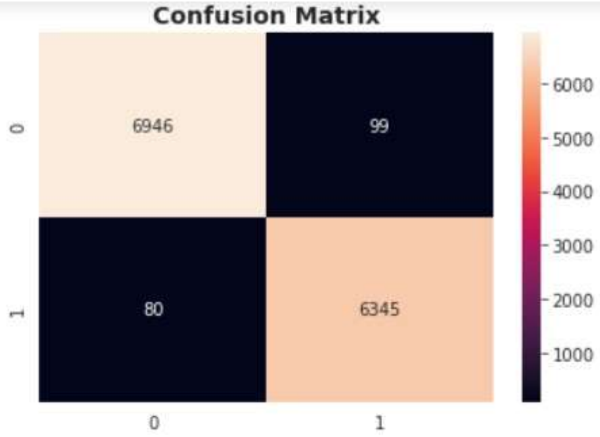


```
AdaBoostClassifier()


accuracy_score:  0.9957683741648107


cross_val_score:  0.9961690814821319


roc_auc_score:  0.995817603897126


Log_loss : 0.14615737487860722


Classification report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      7045
           1       0.99      1.00      1.00      6425

    accuracy                           1.00     13470
   macro avg       1.00      1.00      1.00     13470
weighted avg       1.00      1.00      1.00     13470



Confusion matrix:

[[7008   37]
 [  20 6405]]
```

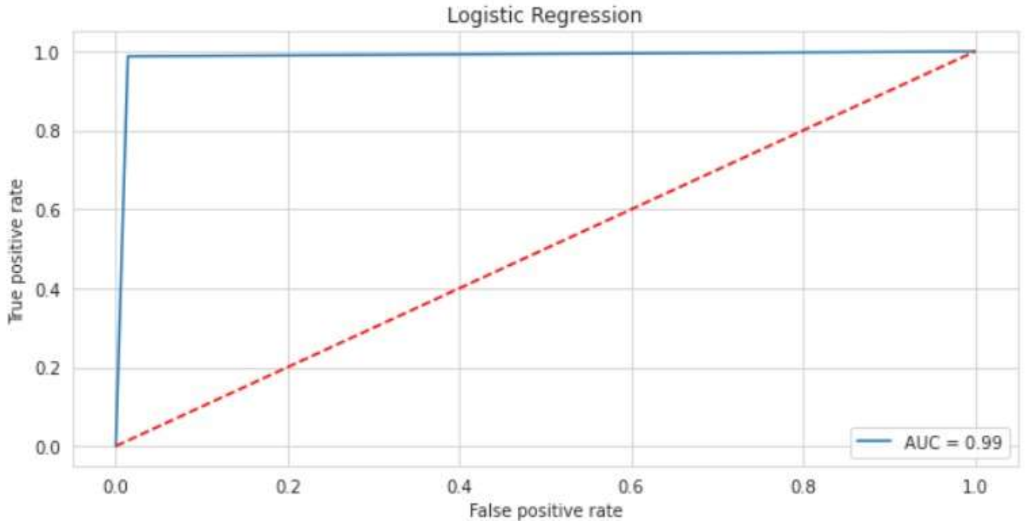


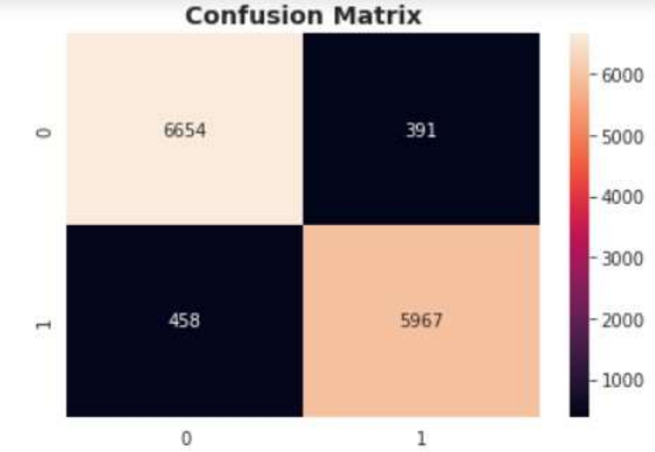
```
GradientBoostingClassifier()


accuracy_score:  0.9956941351150705


cross_val_score:  0.9955677104533196


roc_auc_score:  0.9957329341945745


Log_loss : 0.14872144145599997


Classification report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      7045
           1       0.99      1.00      1.00      6425

    accuracy                           1.00     13470
   macro avg       1.00      1.00      1.00     13470
weighted avg       1.00      1.00      1.00     13470



Confusion matrix:

[[7009   36]
 [  22 6403]]
```

Visualizations

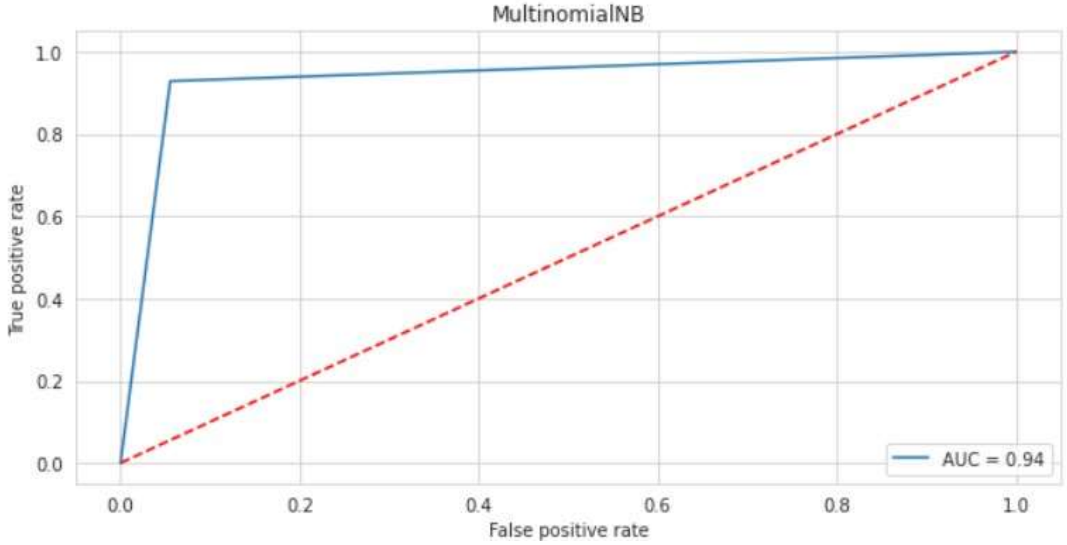


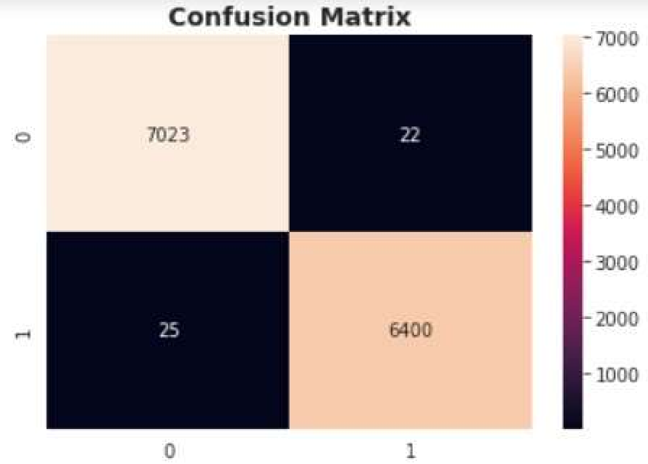
```
AUC_ROC curve:
```



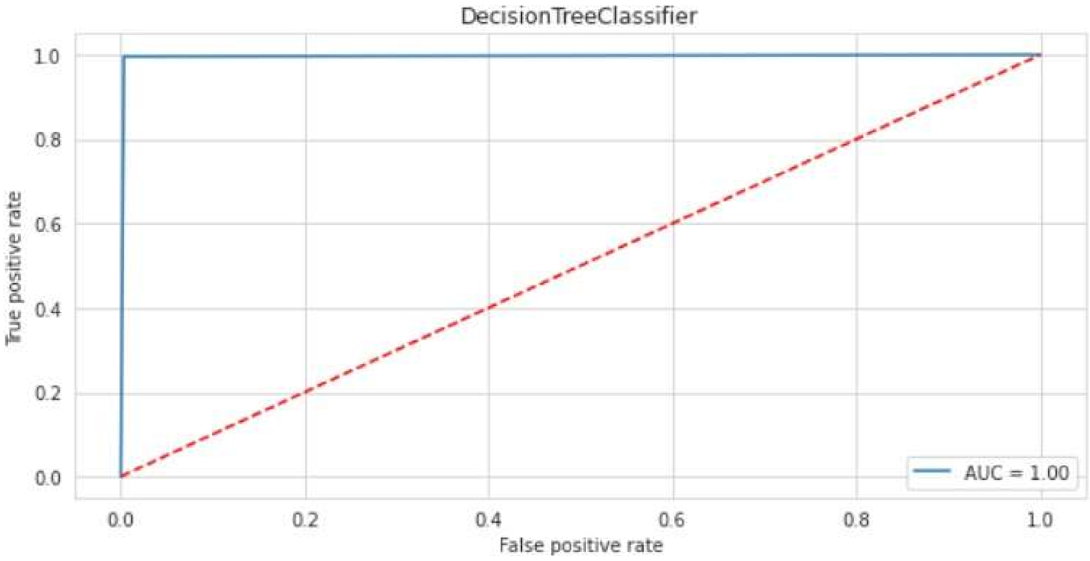


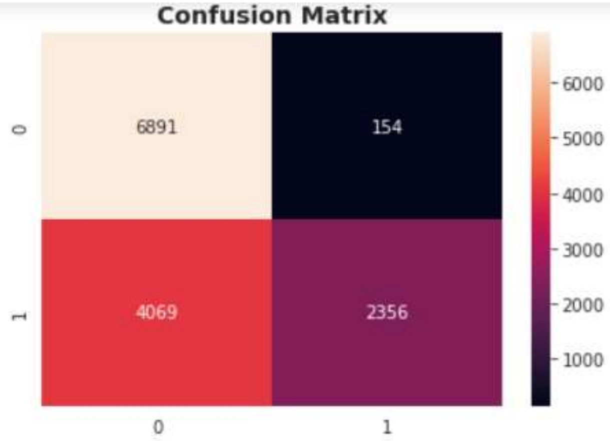
```
AUC_ROC curve:
```



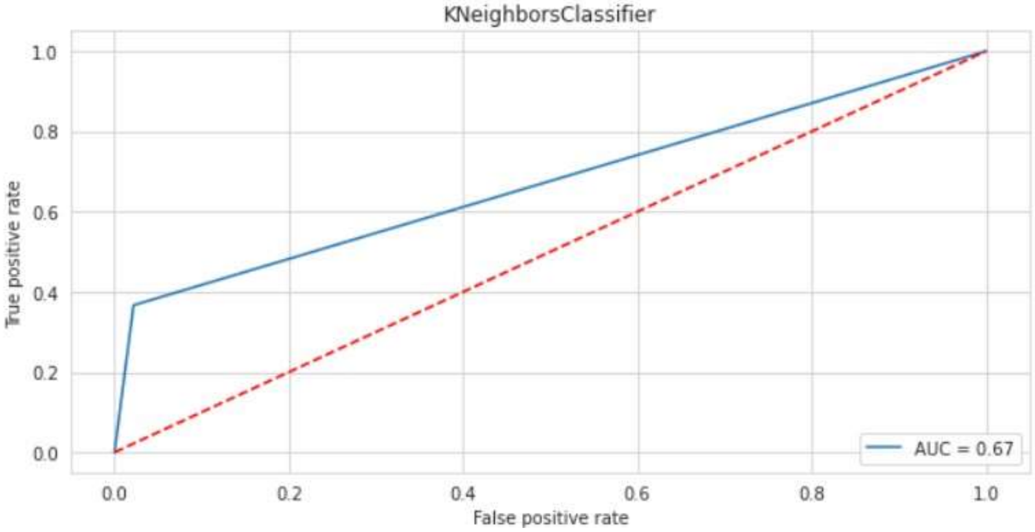


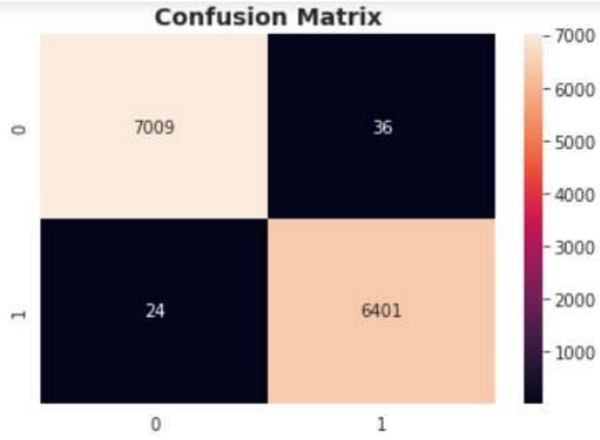
```
AUC_ROC curve:
```



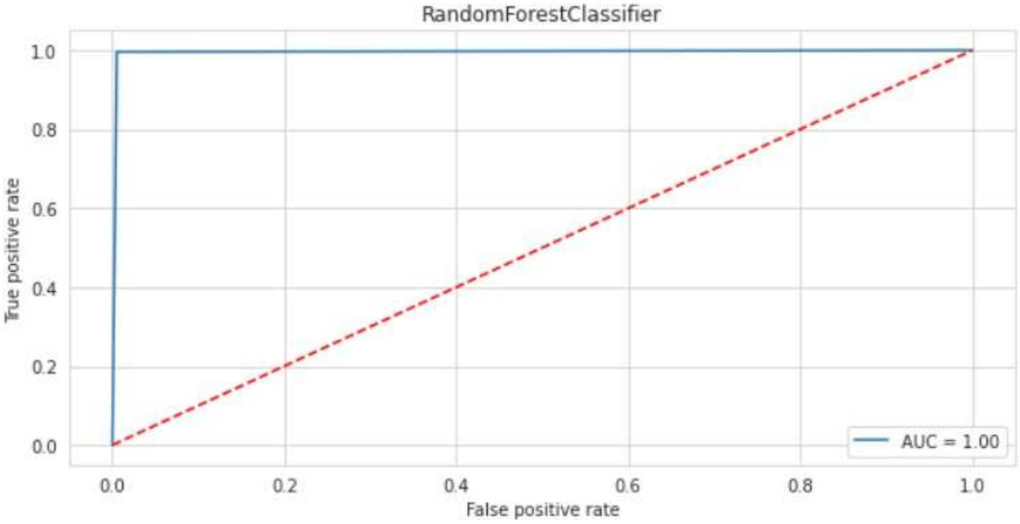


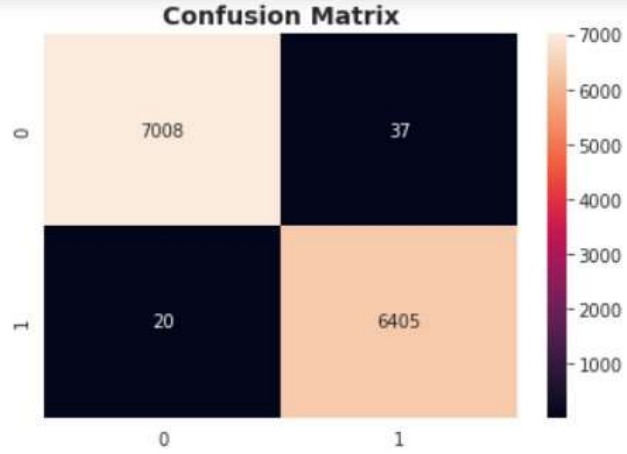
```
AUC_ROC curve:
```



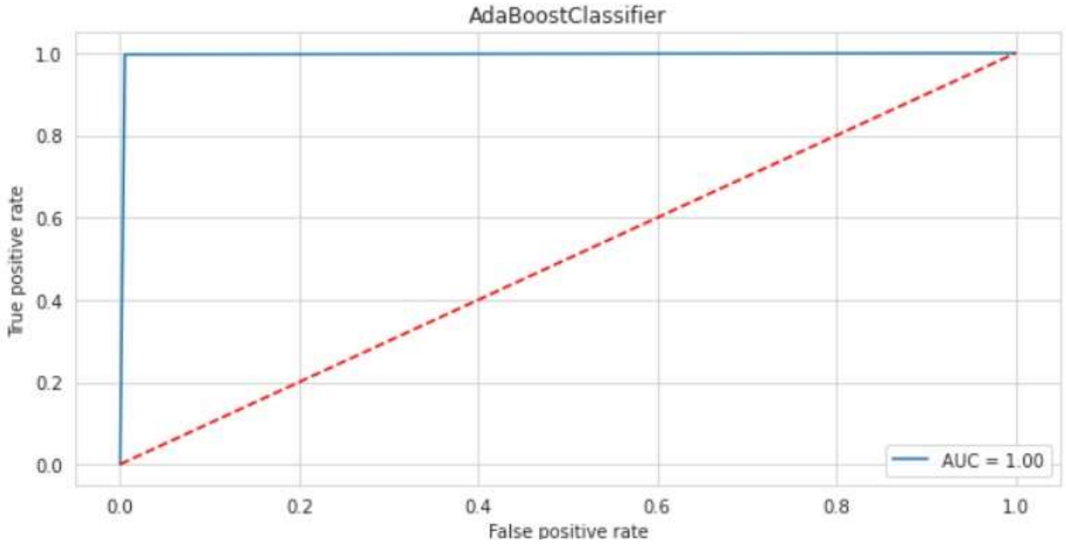


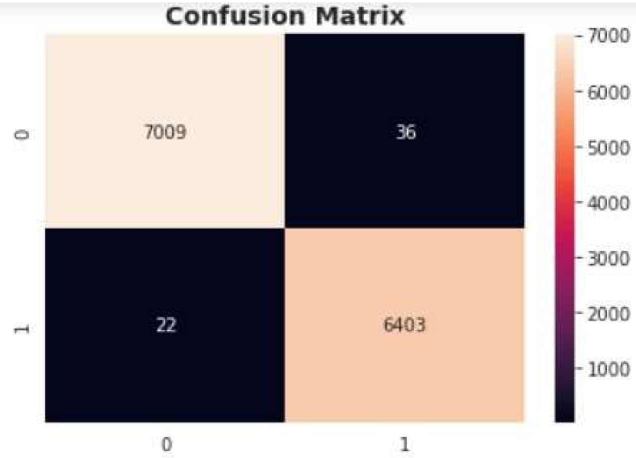
```
AUC_ROC curve:
```



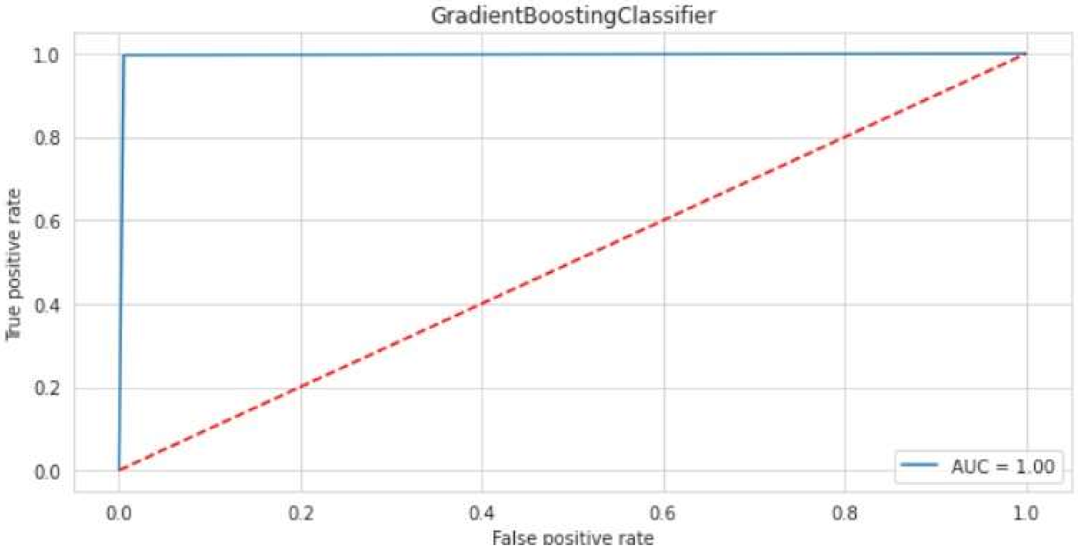


```
AUC_ROC curve:
```

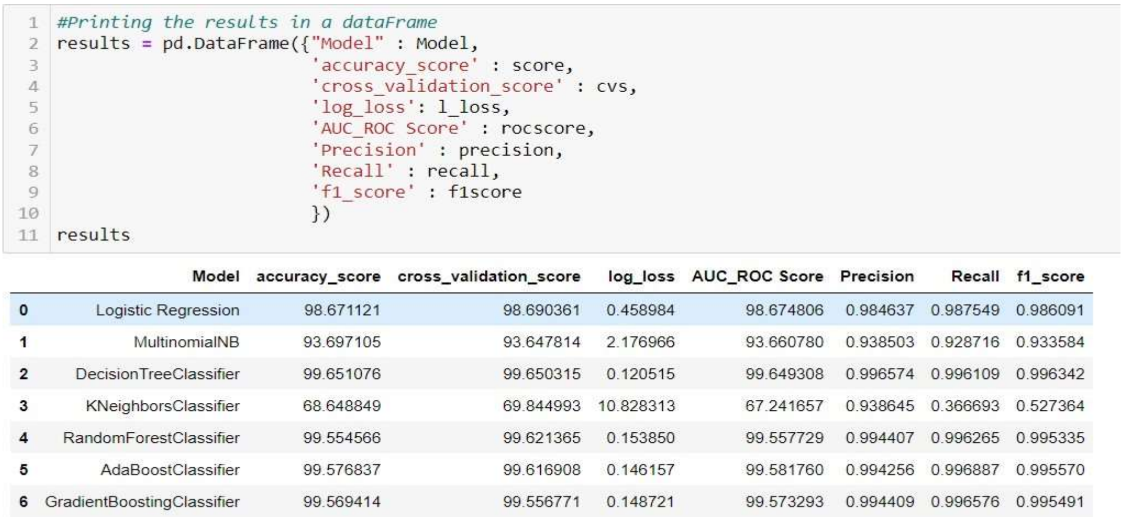




```
AUC_ROC curve:
```



Interpretation of the Results

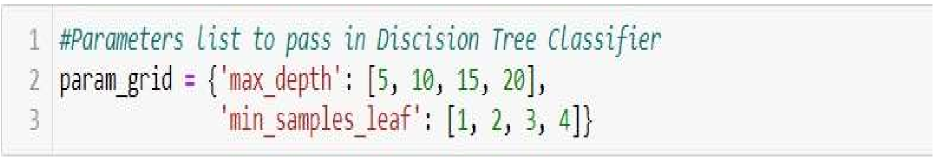


```
#Printing the results in a dataFrame
results = pd.DataFrame({"Model" : Model,
                        'accuracy_score' : score,
                        'cross_validation_score' : cvs,
                        'log_loss': l_loss,
                        'AUC_ROC Score' : rocscore,
                        'Precision' : precision,
                        'Recall' : recall,
                        'f1_score' : f1score
                        })
results
```

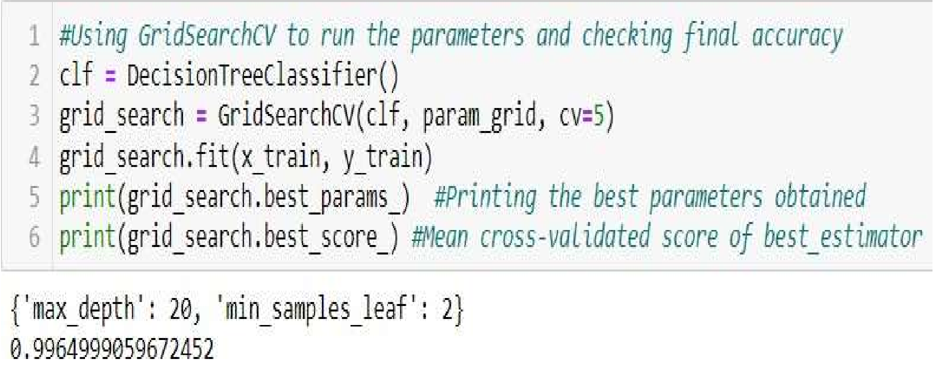
	Model	accuracy_score	cross_validation_score	log_loss	AUC_ROC Score	Precision	Recall	f1_score
0	Logistic Regression	98.671121	98.690361	0.458984	98.674806	0.984637	0.987549	0.986091
1	MultinomialNB	93.697105	93.647814	2.176966	93.660780	0.938503	0.928716	0.933584
2	DecisionTreeClassifier	99.651076	99.650315	0.120515	99.649308	0.996574	0.996109	0.996342
3	KNeighborsClassifier	68.648849	69.844993	10.828313	67.241657	0.938645	0.366693	0.527364
4	RandomForestClassifier	99.554566	99.621365	0.153850	99.557729	0.994407	0.996265	0.995335
5	AdaBoostClassifier	99.576837	99.616908	0.146157	99.581760	0.994256	0.996887	0.995570
6	GradientBoostingClassifier	99.569414	99.556771	0.148721	99.573293	0.994409	0.996576	0.995491

After running the algorithms and according to the scores of performance metrics and other scores, we can see that Decision Tree Classifier algorithms are performing well. Now, we will perform Hyperparameter Tuning to find out the best parameters and try to increase the scores .

Hyper-parameter Tuning

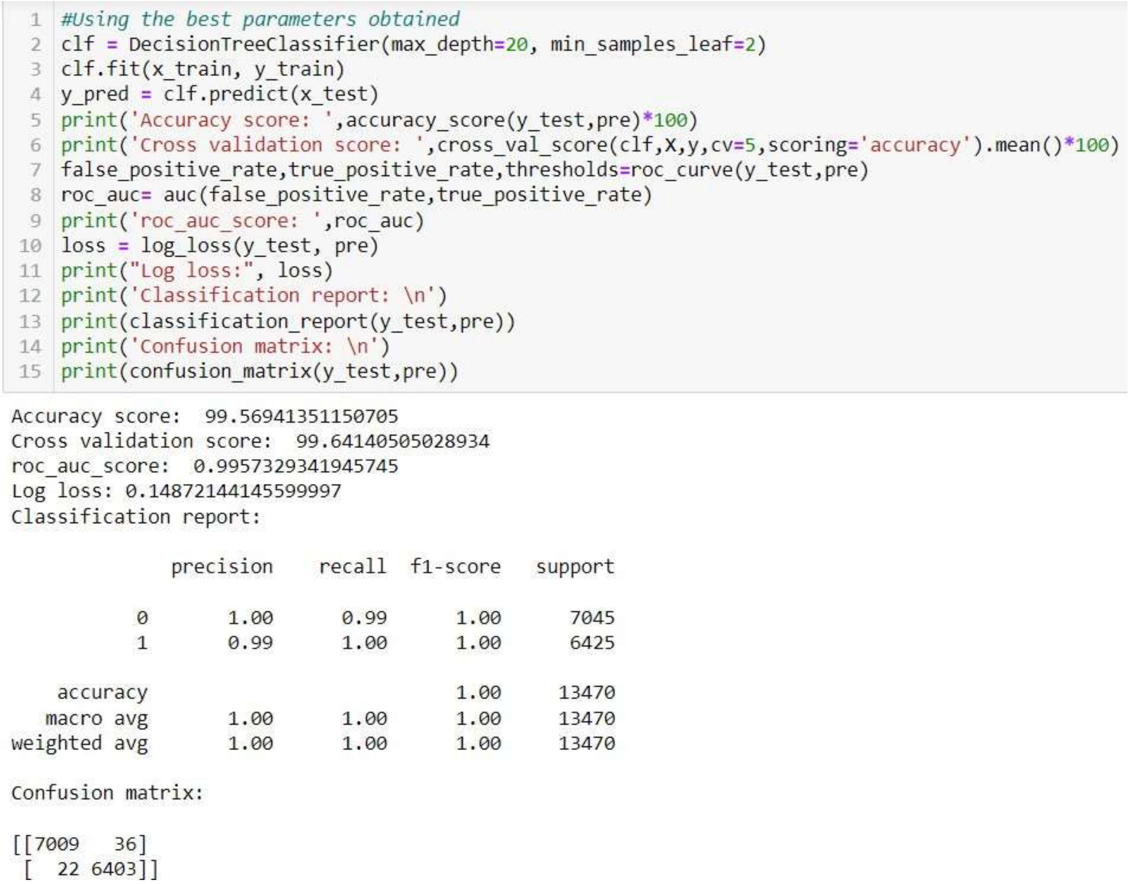


```
#Parameters list to pass in Discision Tree Classifier
param_grid = {'max_depth': [5, 10, 15, 20],
              'min_samples_leaf': [1, 2, 3, 4]}
```



```
#Using GridSearchCV to run the parameters and checking final accuracy
clf = DecisionTreeClassifier()
grid_search = GridSearchCV(clf, param_grid, cv=5)
grid_search.fit(x_train, y_train)
print(grid_search.best_params_)  #Printing the best parameters obtained
print(grid_search.best_score_) #Mean cross-validated score of best_estimator
```

```
{'max_depth': 20, 'min_samples_leaf': 2}
0.9964999059672452
```



```python
#Using the best parameters obtained
clf = DecisionTreeClassifier(max_depth=20, min_samples_leaf=2)
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)
print('Accuracy score: ',accuracy_score(y_test,pre)*100)
print('Cross validation score: ',cross_val_score(clf,X,y,cv=5,scoring='accuracy').mean()*100)
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)
loss = log_loss(y_test, pre)
print("Log loss:", loss)
print('Classification report: \n')
print(classification_report(y_test,pre))
print('Confusion matrix: \n')
print(confusion_matrix(y_test,pre))
```

```
Accuracy score:  99.56941351150705
Cross validation score:  99.64140505028934
roc_auc_score:  0.9957329341945745
Log loss: 0.14872144145599997
Classification report: 

              precision    recall  f1-score   support

           0       1.00      0.99      1.00      7045
           1       0.99      1.00      1.00      6425

    accuracy                           1.00     13470
   macro avg       1.00      1.00      1.00     13470
weighted avg       1.00      1.00      1.00     13470

Confusion matrix: 

[[7009   36]
 [  22 6403]]
```

Conclusion

After the completion of this project, we got an insight of how to preprocess the data, analyzing the data and building a model.

First, we imported the 2 datasets True.csv and Fake.csv which had more than 20000 records each.

We did all the required pre-processing steps like checking null values, datatypes check, dropping unnecessary columns, etc.

We did the Exploratory Data Analysis using various plots and recorded the observations.

Using NLP, we pre-processed the comment text and did other steps like: -Removing Punctuations and other special characters -Splitting the comments into individual words -Removing Stop Words -Stemming and Lemmatising -Applying Count Vectoriser -Plotting wordcloud for knowing the weightage of words used

We created many new features like length of words before pre-processing and after pre-processing inorder to know the words cleaned after the necessary steps.

We applied Tf-idf Vectorizer for scaling the data into number vectors and for x feature we combined the written_by, news and headlines together.

Then, we split the data using train_test_split and then we started the model building process by running as many algorithms in a for loop, with difference metrics like cross_val_score, confusion matrix, auc_score, log loss, precision, recall, f1_score, etc.

We found that Decision Tree Classifier was performing well. The next step was to perform hyperparameter tuning technique to these models for finding out the best parameters and trying to improve our scores.

We finalized the model by predicting the outputs, saving the model and storing the results in a csv file.

Problems faced while working in this project:

1- More computational power was required.

2-More missing data were present in the dataset.

3-Loss was more for some algorithms.