FLIP ROBO

PROJECT REPORT ON:

“FAKE NEWS DETECTION”

SUBMITTED BY

Safiya Firdose Khan

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my mentors from Data Trained academy and Flip Robo Technologies Bangalore for letting me work on this project. Their suggestions and directions have helped me in the completion of this project successfully. All the required information & the dataset are provided by Flip Robo Technologies.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

Contents:

1.INTRODUCTION

1.1 Business Problem Framing:

News media has become a channel to pass on the information of what's happening in the world to the people living. Often people perceive whatever conveyed in the news to be true. There were circumstances where even the news channels acknowledged that their news is not true as they wrote. But some news has a significant impact not only on the people or government but also on the economy. One news can shift the curves up and down depending on the emotions of people and political situation. It is important to identify the fake news from the real true news. The problem has been taken over and resolved with the help of Natural Language Processing tools which help us identify fake or true news based on historical data. The news is now in safe hands!

1.2 Conceptual Background of the Domain Problem

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. The sensationalism of not-so-accurate eye-catching and intriguing headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of information broadcast. On social networking websites, the reach and effects of information spread are however significantly amplified and occur at such a fast pace, that distorted, inaccurate, or false information acquires a tremendous potential to cause real impacts, within minutes, for millions of users.

1.3 Review of Literature

Fake news is not a new concept. Before the era of digital technology, it was spread through mainly yellow journalism with a focus on sensational news such as crime, gossip, disasters and satirical news. With the widespread dissemination of information via digital media platforms, it is of utmost importance for individuals and societies to be able to judge the credibility of it. Fake news is not a recent concept, but it is a commonly occurring phenomenon in current times. The consequence of fake news can range from being merely annoying to influencing and misleading societies or even nations. A variety of approaches exist to identify fake news.

1.4 Motivation for the Problem Undertaken

The widespread problem of fake news is very difficult to tackle in today's digital world where there are thousands of information sharing platforms through which fake news or misinformation may propagate. It has become a greater issue because of the advancements in AI which brings along artificial bots that may be used to create and spread fake news. The situation is dire because many people believe anything they read on the internet and the ones who are amateur or are new to the digital technology may be easily fooled. A similar problem is fraud that may happen due to spam or malicious emails and messages. So, it is compelling enough to acknowledge this problem take on this challenge to control the rates of crime, political unrest, grief, and thwart the attempts of spreading fake news. Text, or natural language, is one form that is difficult to process simply because of various linguistic features and styles like sarcasm, metaphors, etc. Moreover, there are thousands of spoken languages and every language has its grammar, script and syntax. Natural language processing is a branch of artificial intelligence and it encompasses techniques that can utilize text, create models and produce predictions. This work aims to create a system or model that can use the data of past news reports and predict the chances of a news report being fake or not.

2. Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem

1) Cleaned Data by removing irrelevant features

2) Pre-processing of text using NLP processing

3) Used Word Counts

4) Used Character Counts

5) Used Tf-Idf Vectorizer

6) Split data into train and test

7) Built Model

8) Hyper parameter tuning

2.2 Data Sources and their formats

The data-set is in csv format: fake_news.csv and true_news.csv.

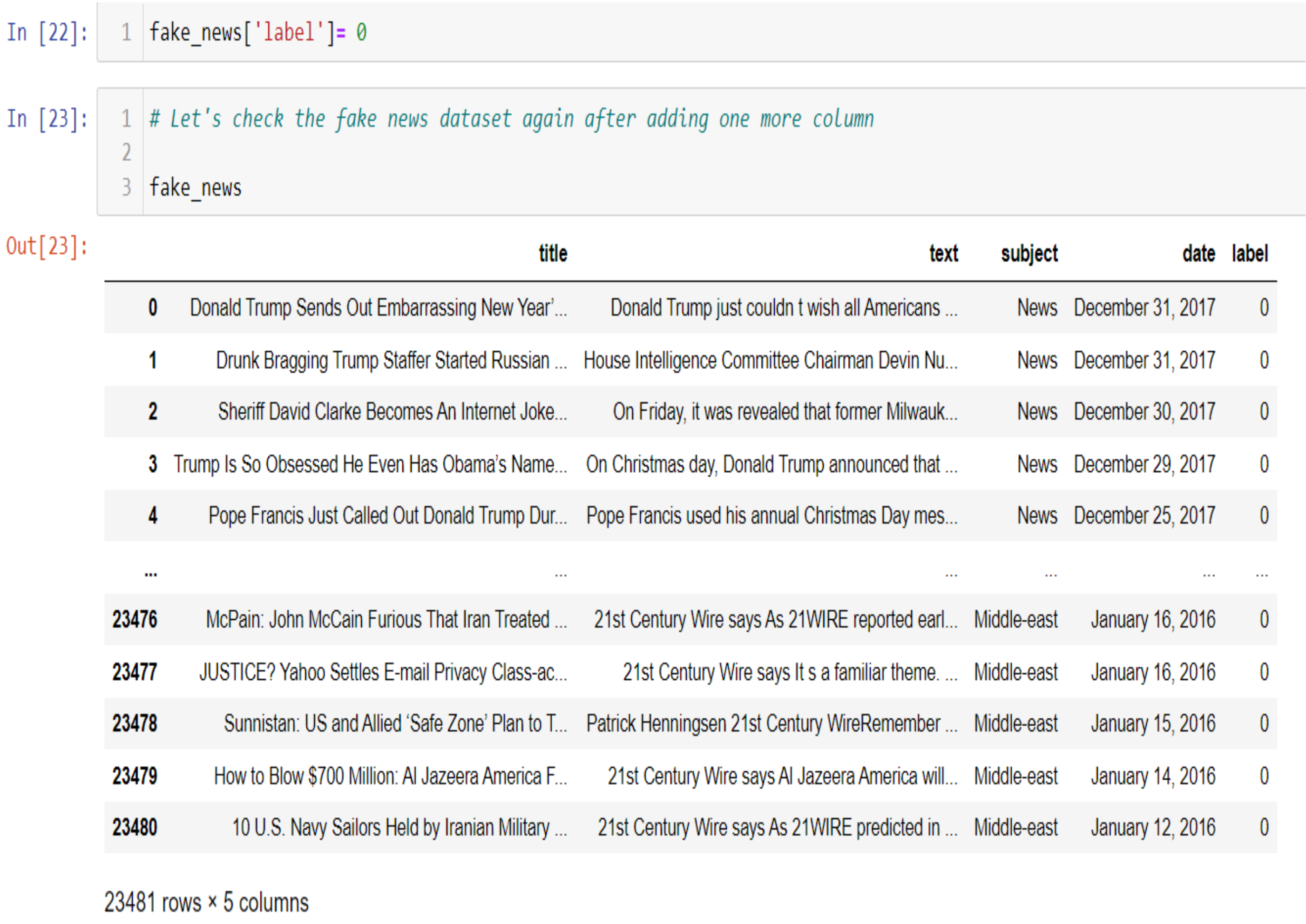
Features of this dataset are:

- title
- text (containing news)
- subject
- date

2.3 Data Preprocessing Done

- ✓ As a first step, I have imported required libraries and I have imported the two datasets, one for fake news and one for true news, which are in csv format.

- ✓ As there is no label column present, I have inserted the label column zero for fake news and one for true news.



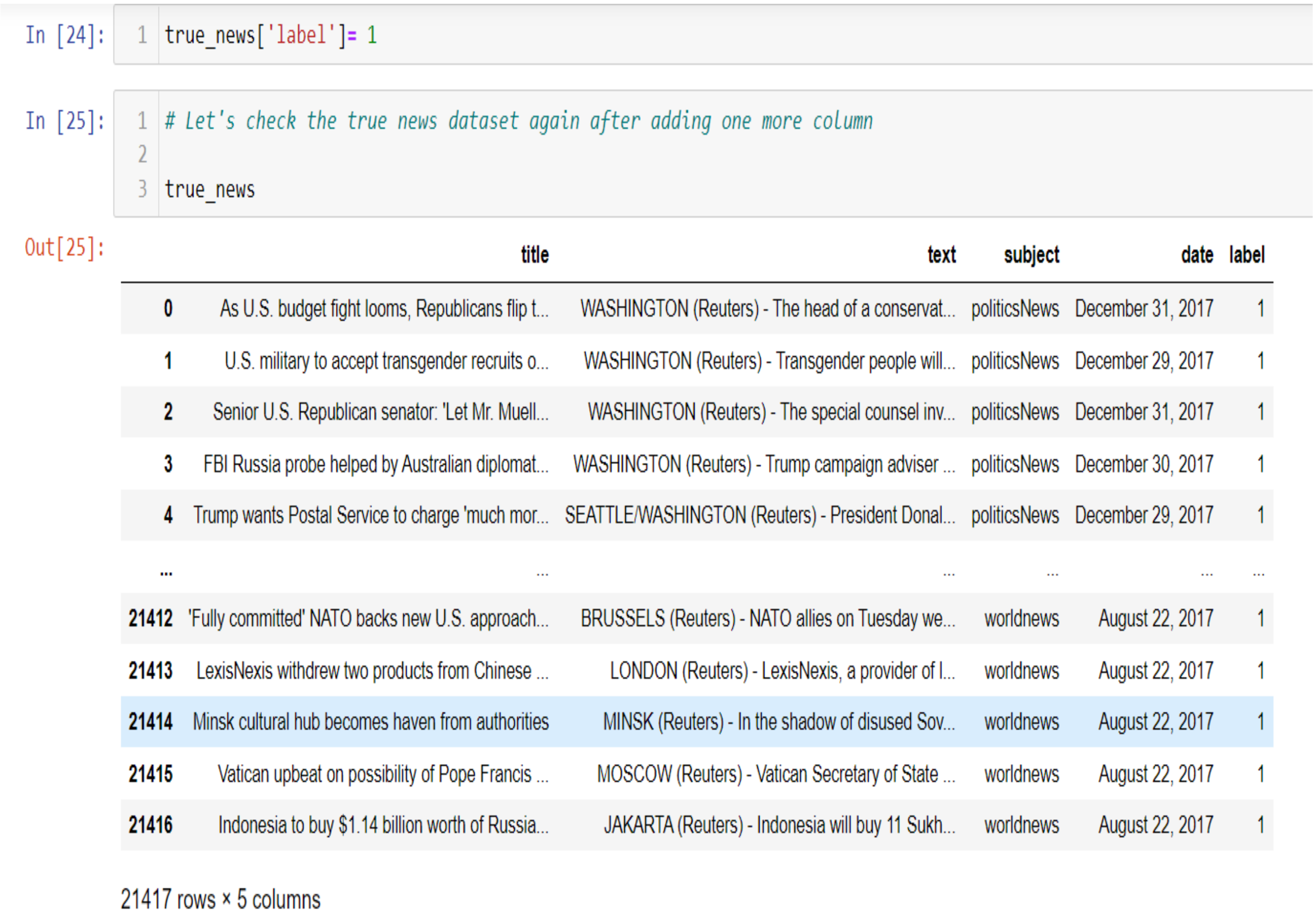
In [22]:
```
fake_news['label']= 0
```

In [23]:
```
# Let's check the fake news dataset again after adding one more column

fake_news
```

Out[23]:

	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
...	...	...	...	...	...
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	0
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016	0
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	0
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	0
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	0

23481 rows × 5 columns



In [24]:

```python
true_news['label']= 1
```

In [25]:

```python
# Let's check the true news dataset again after adding one more column

true_news
```

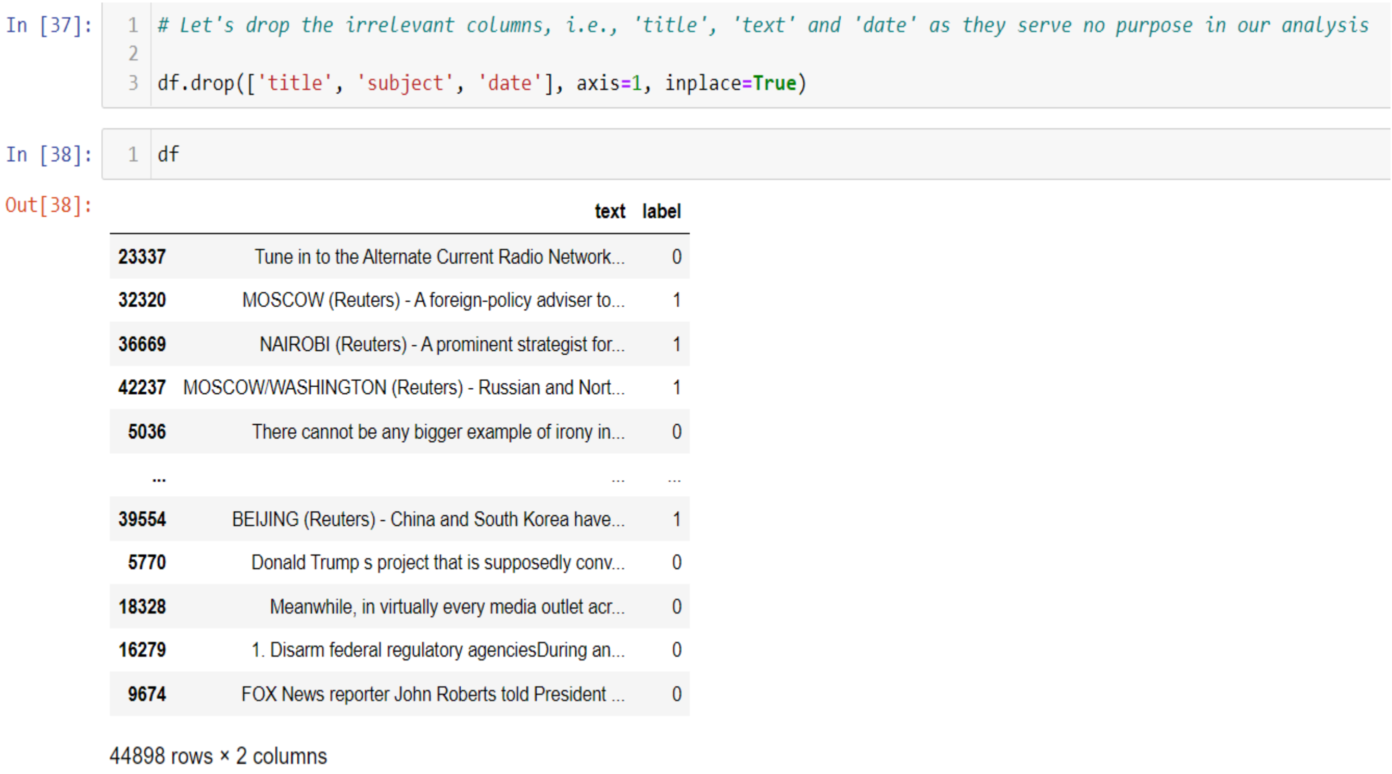
Out[25]:

	title	text	subject	date	label
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1
...	...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

21417 rows × 5 columns

- ✓ Then, I performed the statistical analysis on the dataset, like checking shape, nunique, info, etc.
- ✓ Then, I checked the null values in both the datasets, and found out that there are no null values present in both the datasets.

✓ I have dropped the irrelevant columns, as they serve no purpose in our analysis.



In [37]:

```
# Let's drop the irrelevant columns, i.e., 'title', 'text' and 'date' as they serve no purpose in our analysis

df.drop(['title', 'subject', 'date'], axis=1, inplace=True)
```

In [38]:

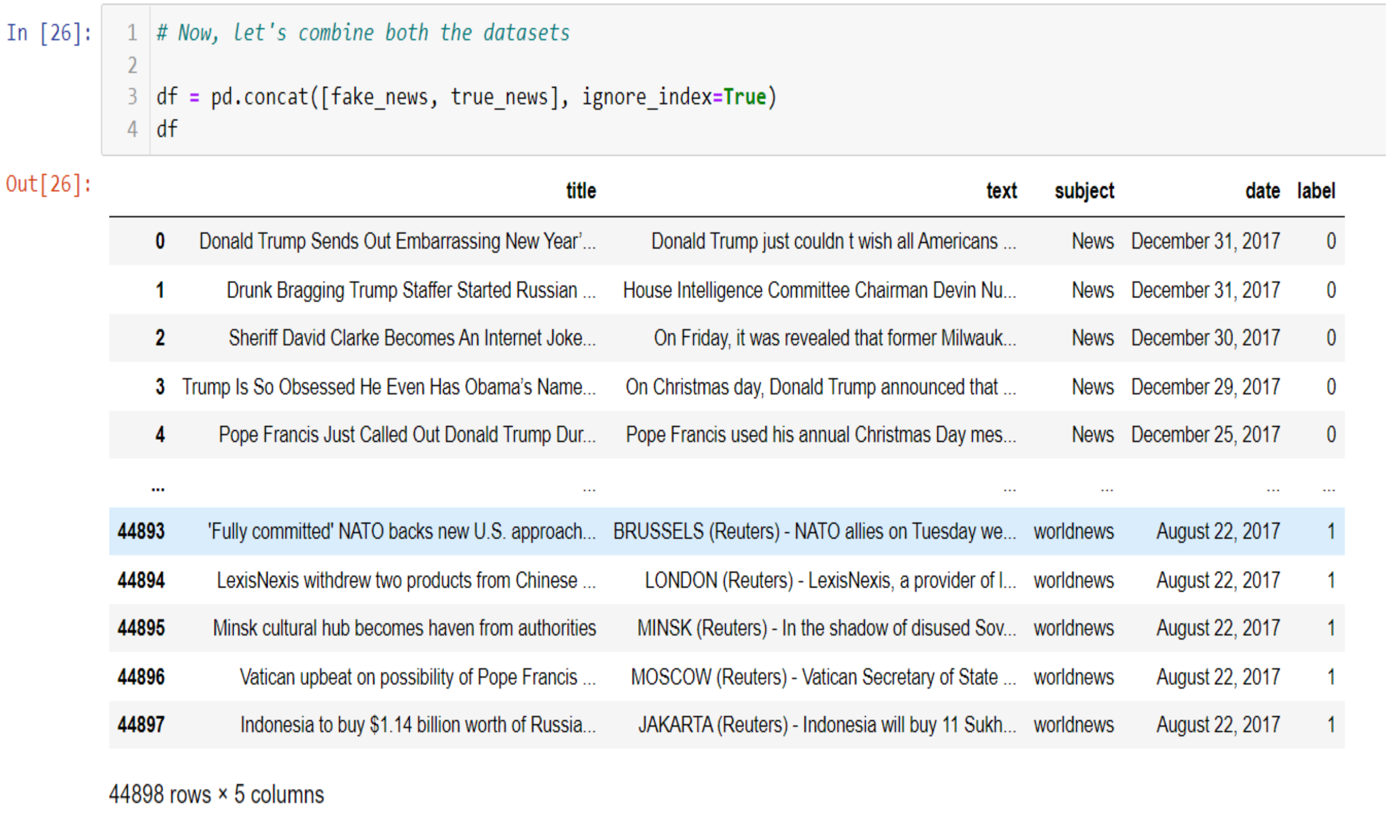
```
df
```

Out[38]:

	text	label
23337	Tune in to the Alternate Current Radio Network...	0
32320	MOSCOW (Reuters) - A foreign-policy adviser to...	1
36669	NAIROBI (Reuters) - A prominent strategist for...	1
42237	MOSCOW/WASHINGTON (Reuters) - Russian and Nort...	1
5036	There cannot be any bigger example of irony in...	0
...	...	...
39554	BEIJING (Reuters) - China and South Korea have...	1
5770	Donald Trump s project that is supposedly conv...	0
18328	Meanwhile, in virtually every media outlet acr...	0
16279	1. Disarm federal regulatory agenciesDuring an...	0
9674	FOX News reporter John Roberts told President ...	0

44898 rows × 2 columns

- ✓ Then, I have combined both the datasets using pandas built-in function.



In [26]:

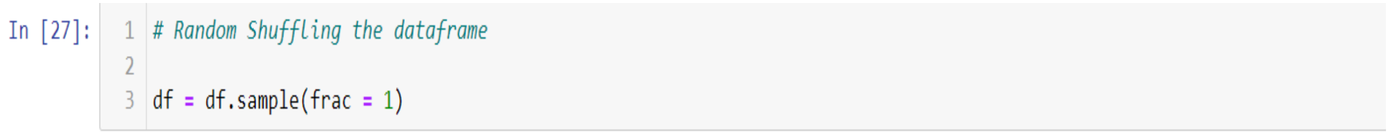
```
# Now, let's combine both the datasets

df = pd.concat([fake_news, true_news], ignore_index=True)
df
```

Out[26]:

	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
...	...	...	...	...	...
44893	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
44894	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
44895	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
44896	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
44897	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

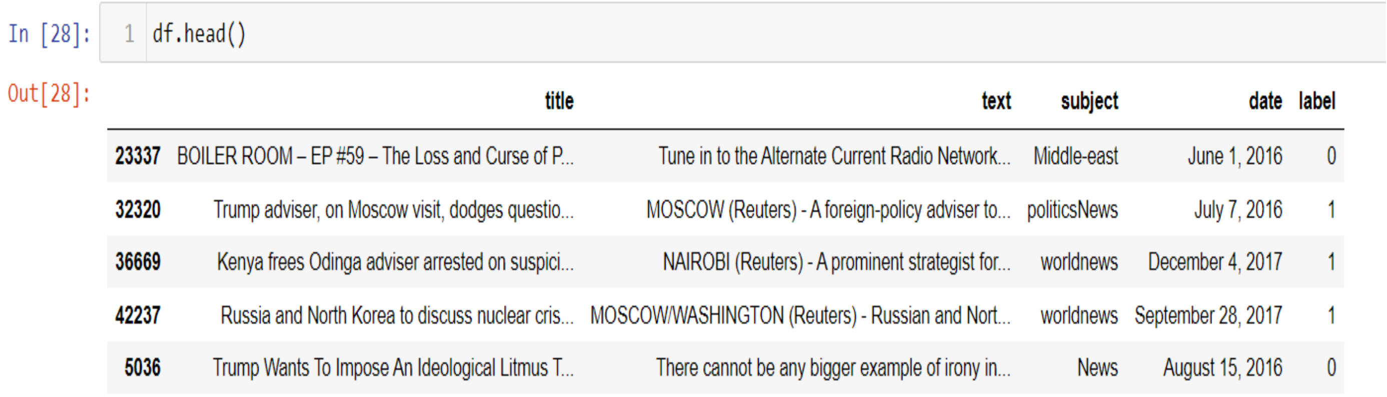
44898 rows × 5 columns

- ✓ Then, I have randomly shuffled the dataframe.



In [27]:

```
# Random Shuffling the dataframe

df = df.sample(frac = 1)
```



In [28]:

```
df.head()
```

Out[28]:

	title	text	subject	date	label
23337	BOILER ROOM – EP #59 – The Loss and Curse of P...	Tune in to the Alternate Current Radio Network...	Middle-east	June 1, 2016	0
32320	Trump adviser, on Moscow visit, dodges questio...	MOSCOW (Reuters) - A foreign-policy adviser to...	politicsNews	July 7, 2016	1
36669	Kenya frees Odinga adviser arrested on suspici...	NAIROBI (Reuters) - A prominent strategist for...	worldnews	December 4, 2017	1
42237	Russia and North Korea to discuss nuclear cris...	MOSCOW/WASHINGTON (Reuters) - Russian and Nort...	worldnews	September 28, 2017	1
5036	Trump Wants To Impose An Ideological Litmus T...	There cannot be any bigger example of irony in...	News	August 15, 2016	0

- ✓ Cleaning the raw data involves the deletion of words or special characters that do not add meaning to the text.

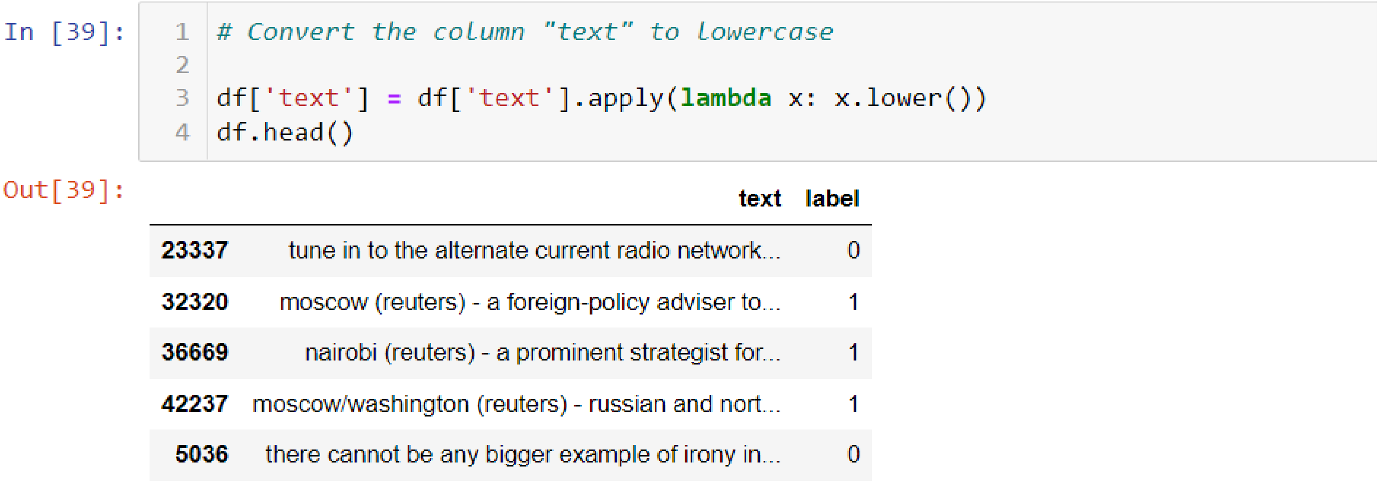
Important cleaning steps are as follows:

1. Lowering case

2. Handling of special characters

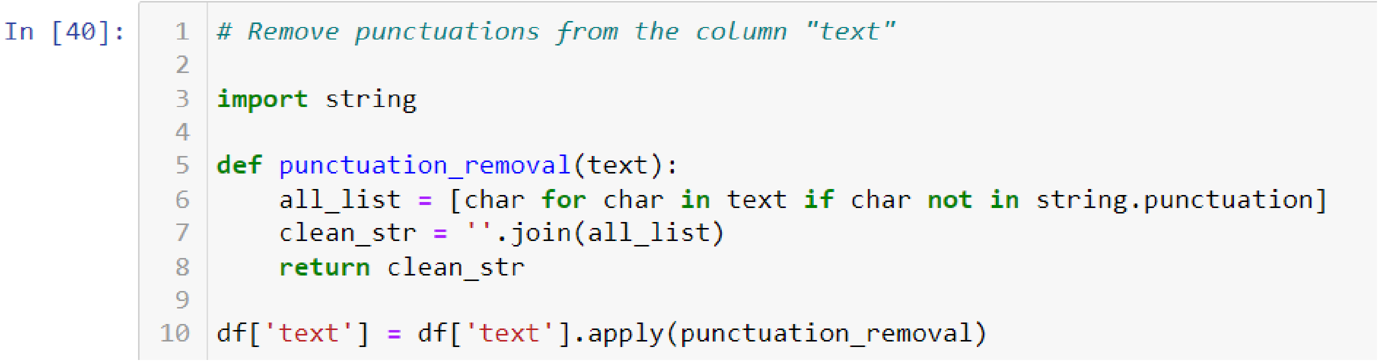
3. Removal of stopwords

4. Converting words to the most suitable base form by using lemmatization.

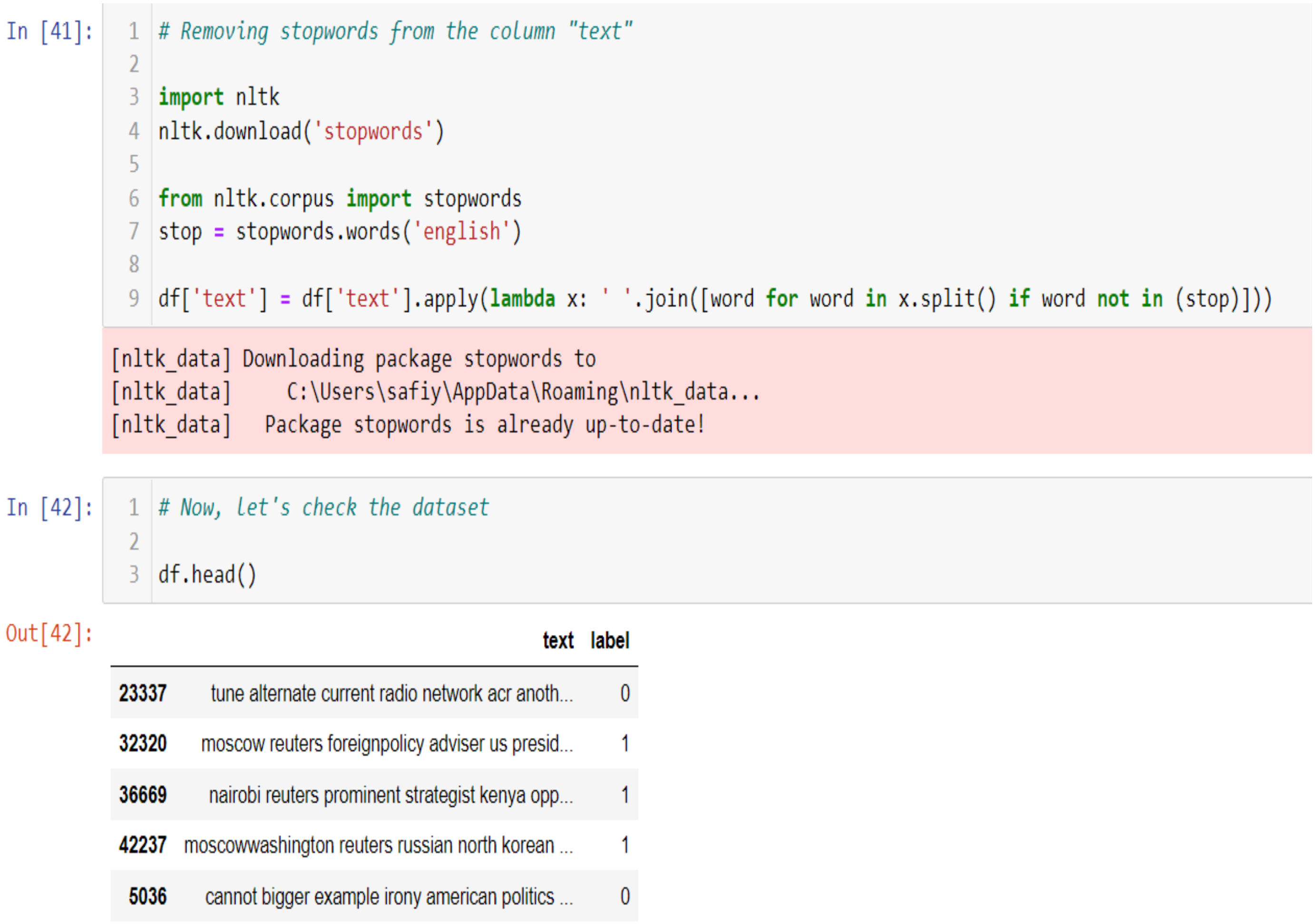


```
In [39]:
# Convert the column "text" to lowercase

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

```
Out[39]:
                                                     text  label
23337        tune in to the alternate current radio network...      0
32320      moscow (reuters) - a foreign-policy adviser to...      1
36669          nairobi (reuters) - a prominent strategist for...      1
42237  moscow/washington (reuters) - russian and nort...      1
 5036     there cannot be any bigger example of irony in...      0
```



```
In [40]:
# Remove punctuations from the column "text"

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

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



In [41]:

```
# Removing stopwords from the column "text"

import nltk
nltk.download('stopwords')

from nltk.corpus import stopwords
stop = stopwords.words('english')

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

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\safiy\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

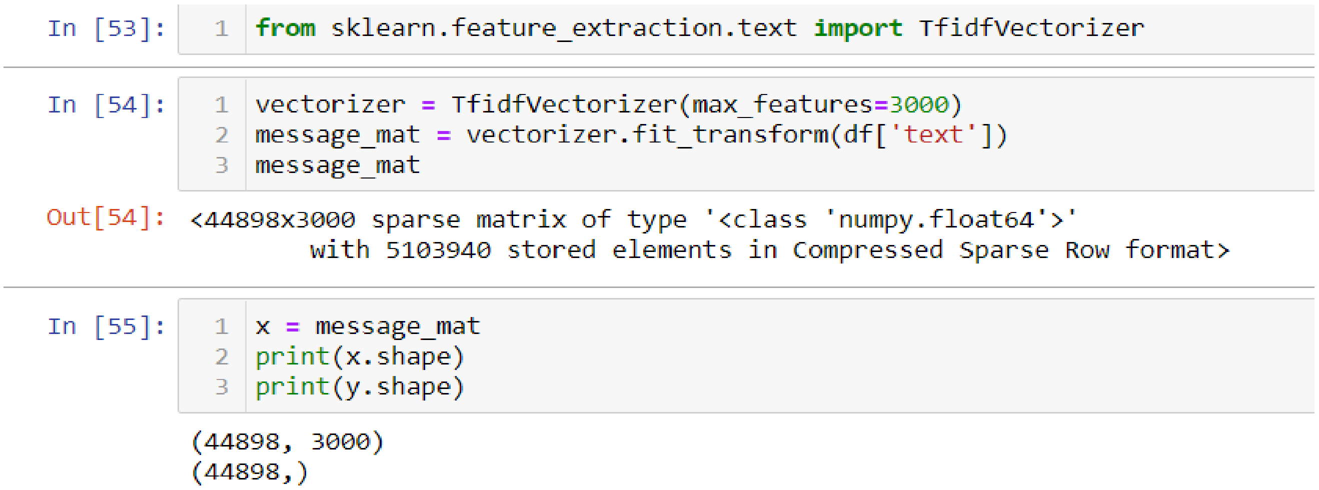
In [42]:

```
# Now, let's check the dataset

df.head()
```

Out[42]:

	text	label
23337	tune alternate current radio network acr anoth...	0
32320	moscow reuters foreignpolicy adviser us presid...	1
36669	nairobi reuters prominent strategist kenya opp...	1
42237	moscowwashington reuters russian north korean ...	1
5036	cannot bigger example irony american politics ...	0

- ✓ Then, I have converted text to vectors using TfidfVectorizer.



In [53]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [54]:

```
vectorizer = TfidfVectorizer(max_features=3000)
message_mat = vectorizer.fit_transform(df['text'])
message_mat
```

Out[54]:

```
<44898x3000 sparse matrix of type '<class 'numpy.float64'>'
        with 5103940 stored elements in Compressed Sparse Row format>
```

In [55]:

```
x = message_mat
print(x.shape)
print(y.shape)
```

```
(44898, 3000)
(44898,)
```

2.4 Data Inputs- Logic- Output Relationships

For this data's input and output logic, we will analyze the column "text", and study its relationship with the column "label".

2.5 Hardware and Software Requirements and Tools Used

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software.

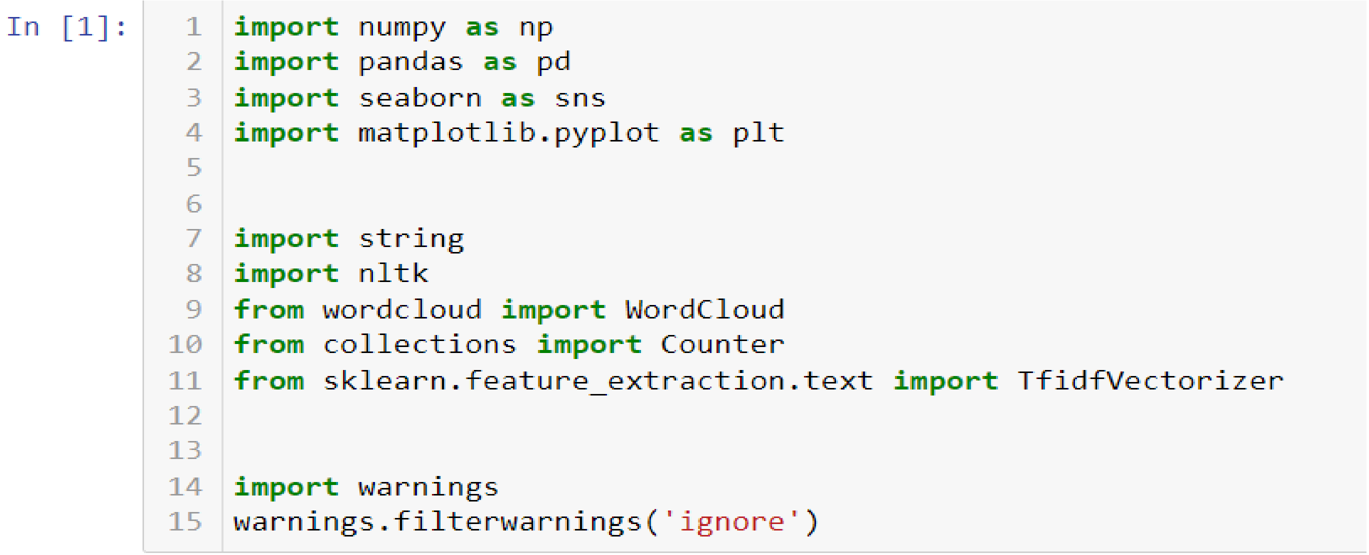
Hardware required: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1. Jupyter Notebook (Anaconda 3) – Python 3.7.6
2. Microsoft Excel 2010

Libraries required :-



```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


import string
import nltk
from wordcloud import WordCloud
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer


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

✓ **Pandas:** Pandas is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.

✓ **Seaborn:** Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

✓ **Matplotlib:** matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

✓ **Scikit Learn:** This is the most important library for Machine Learning since it contains various Machine Learning Algorithms which are used in this project. Scikit Learn also contains Preprocessing library which is used in data preprocessing. Apart from this, it contains a very useful joblib library for serialization purpose using which the final model has been saved in this project.

- ✓ **NLTK:** Natural language took kit is one of the most used libraries for building NLP projects.

3. Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)

Understanding the problem is the first crucial step in solving any problem. From the given dataset, it can be concluded that it is a binary classification problem. I have built the model by using 10 different classification algorithms, and finally saved the best model.

3.2 Testing of Identified Approaches (Algorithms)

Since “label” is my target variable and it is discrete in nature, so this particular problem is a classification problem. I have used 10 different classification algorithms to build my model. By looking at the accuracy score and Cross Validation score, I found that Light Gradient Boosting Machine Classifier is the best model. Below is the list of classification algorithms, that I have used in my project.

- ✓ Logistic Regression
- ✓ Random Forest Classifier
- ✓ Decision Tree Classifier
- ✓ Linear Support Vector Machine Classifier
- ✓ XGB Classifier
- ✓ Gradient Boosting Classifier
- ✓ Stochastic Gradient Descent Classifier
- ✓ Bernoulli Naive Bayes Classifier
- ✓ Multinomial Naive Bayes Classifier
- ✓ Light Gradient Boosting Machine Classifier

3.3 Key Metrics for success in solving problem under consideration

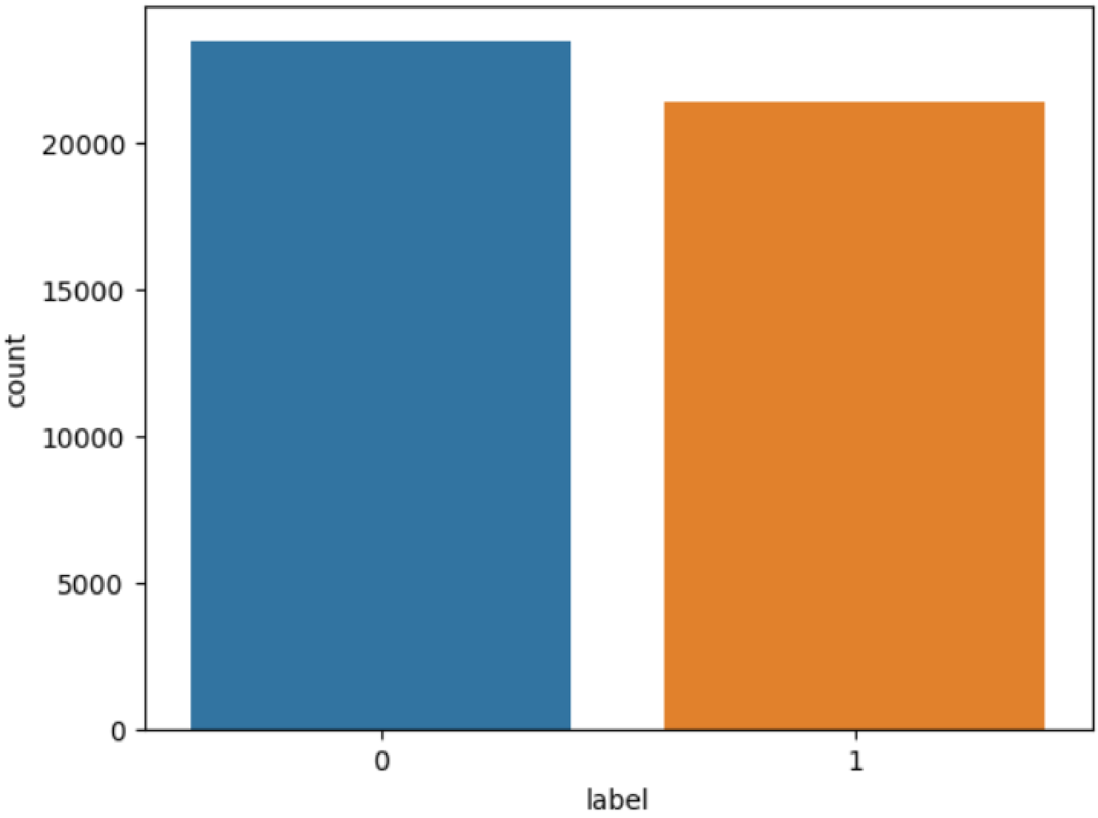
Accuracy Score, ROC AUC score, Confusion Matrix, Classification Report and Cross Validation score are used for evaluating the model.

3.4 Visualizations

```
In [43]:
print(df['label'].value_counts())
sns.countplot(df['label'])
```

```
0    23481
1    21417
Name: label, dtype: int64
```

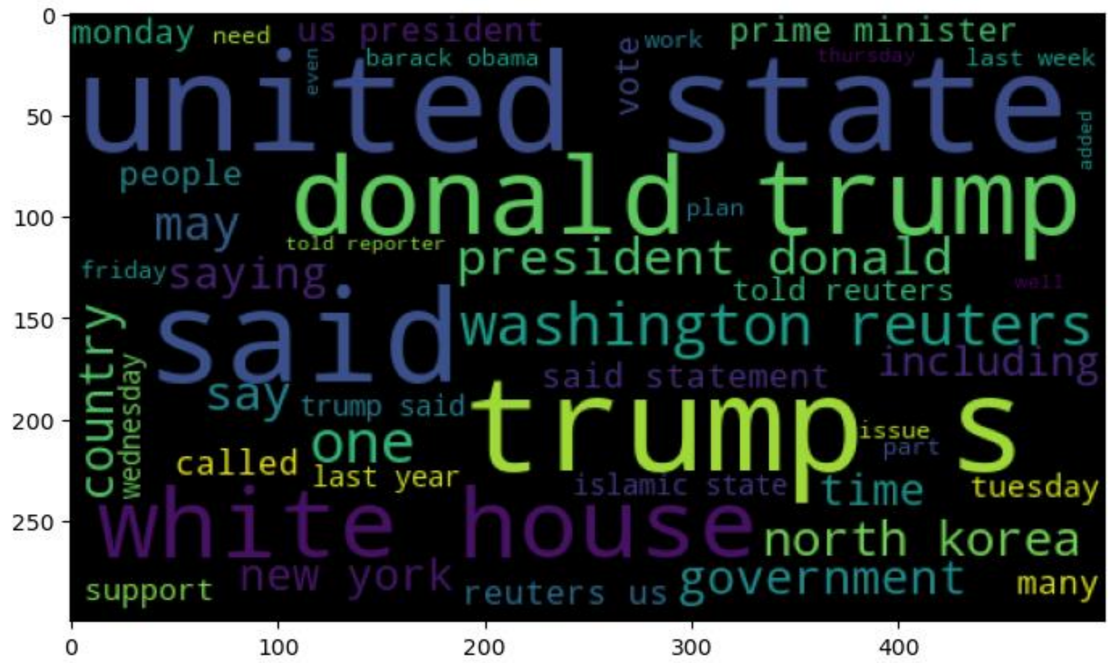
```
Out[43]: <AxesSubplot:xlabel='label', ylabel='count'>
```



```
In [44]:  from wordcloud import WordCloud
```

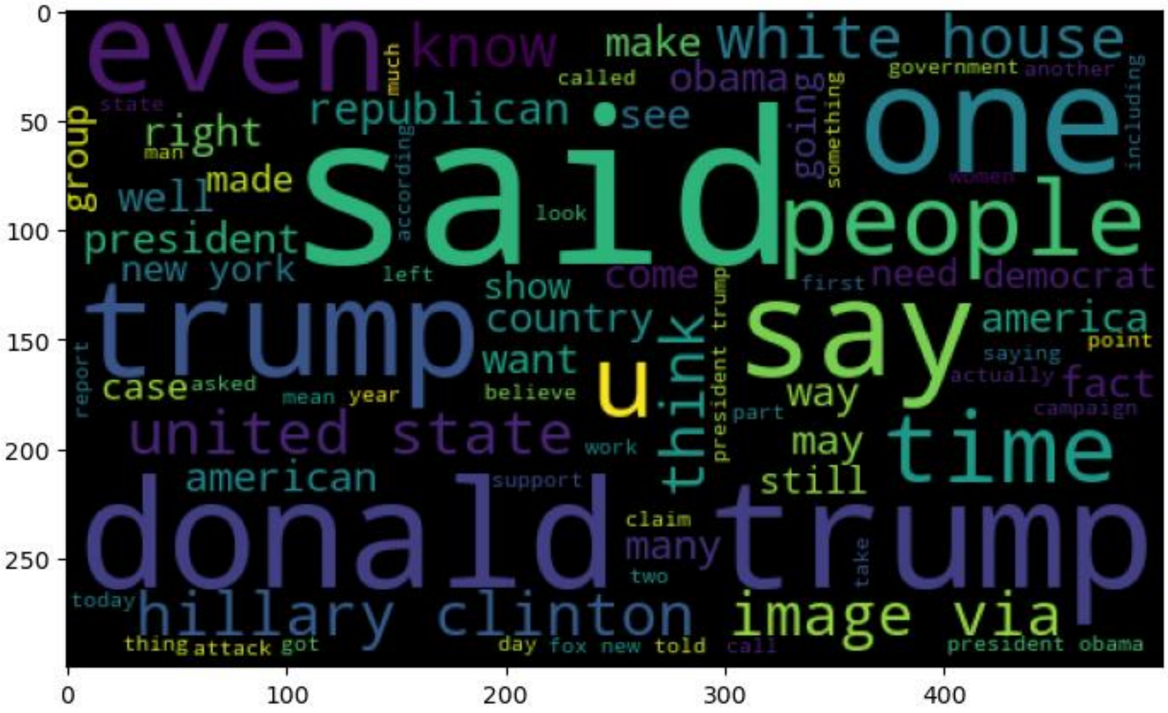
```
In [45]:  wc = WordCloud(width = 500, height = 300, min_font_size= 10, background_color= 'black')
```

```
In [46]:  # Generating Word Cloud for True News

           true_wordcloud = wc.generate(df[df['label']==1]['text'].str.cat(sep = " "))
           plt.figure(figsize=(10,5))
           plt.imshow(true_wordcloud)
           plt.show()
```



In [47]:

```python
# Generating Word Cloud for Fake News

fake_wordcloud = wc.generate(df[df['label']==0]['text'].str.cat(sep = " "))
plt.figure(figsize=(10,5))
plt.imshow(fake_wordcloud)
plt.show()
```

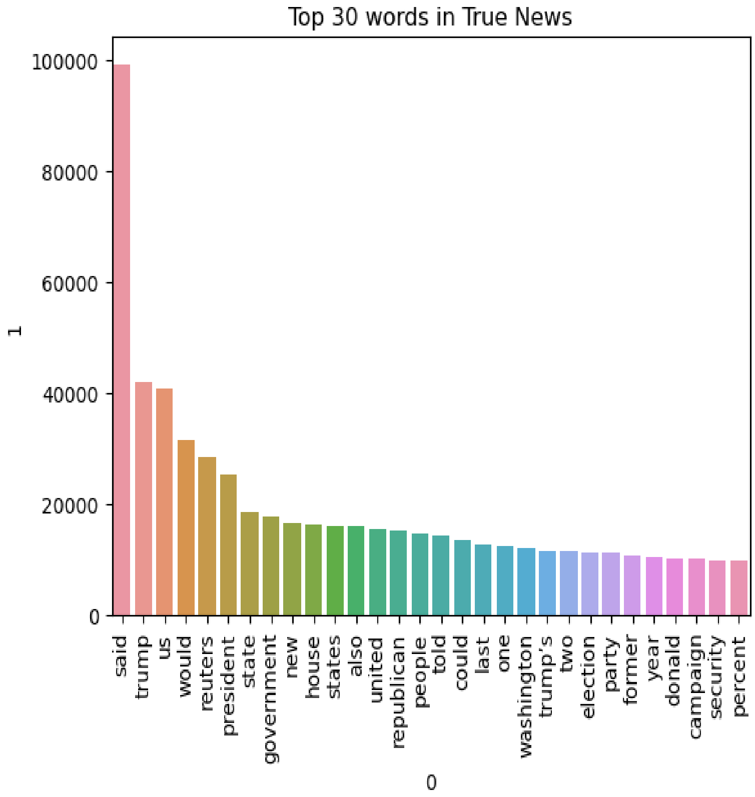


In [48]:

```
true_corpus = []

for msg in df[df['label']==1]['text'].tolist():
    for word in msg.split():
        true_corpus.append(word)
```

In [49]:

```
from collections import Counter

sns.barplot(pd.DataFrame(Counter(true_corpus).most_common(30))[0] , pd.DataFrame(Counter(true_corpus).most_common(30))[1])

plt.title("Top 30 words in True News")
plt.xticks(rotation = "vertical")
plt.show()
```

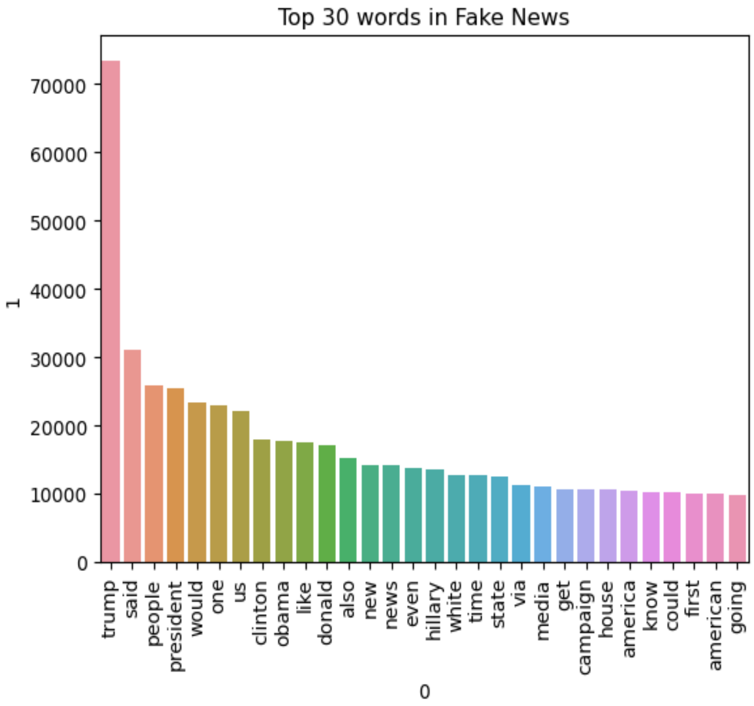


In [50]:

```python
fake_corpus = []

for msg in df[df['label']==0]['text'].tolist():
    for word in msg.split():
        fake_corpus.append(word)
```

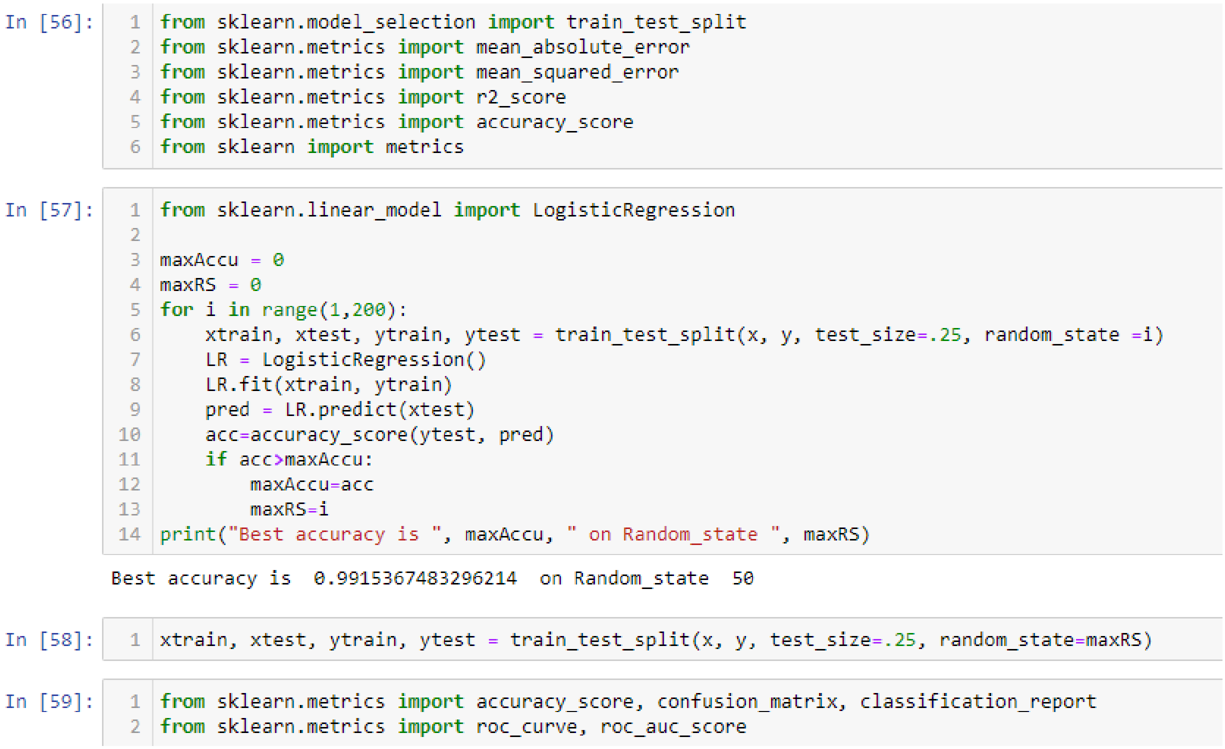
In [51]:

```python
sns.barplot(pd.DataFrame(Counter(fake_corpus).most_common(30))[0] , pd.DataFrame(Counter(fake_corpus).most_common(30))[1])

plt.title("Top 30 words in Fake News")
plt.xticks(rotation = "vertical")
plt.show()
```



3.5 Run and Evaluate selected models

1. Finding best random state



In [56]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score
from sklearn import metrics
```

In [57]:

```
from sklearn.linear_model import LogisticRegression

maxAccu = 0
maxRS = 0
for i in range(1,200):
    xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=.25, random_state =i)
    LR = LogisticRegression()
    LR.fit(xtrain, ytrain)
    pred = LR.predict(xtest)
    acc=accuracy_score(ytest, pred)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print("Best accuracy is ", maxAccu, " on Random_state ", maxRS)
```

```
Best accuracy is  0.9915367483296214  on Random_state  50
```

In [58]:

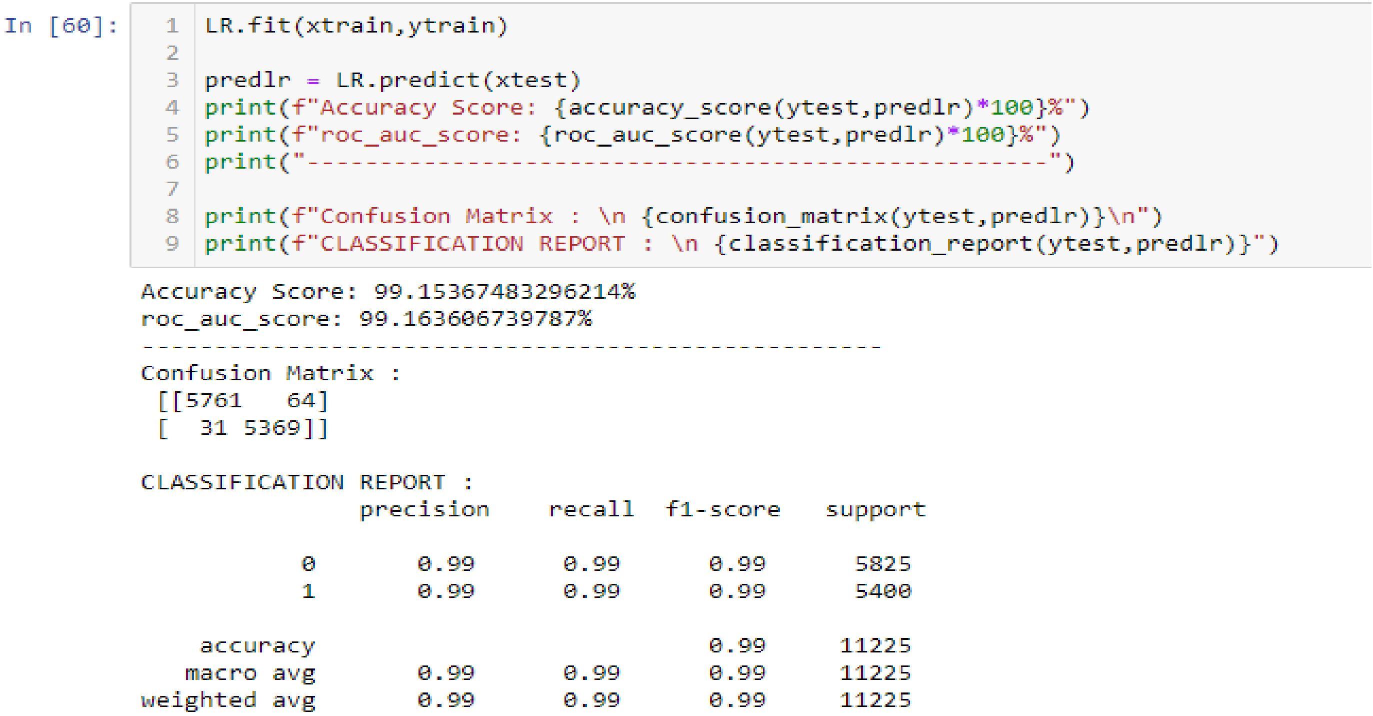
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=.25, random_state=maxRS)
```

In [59]:

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import roc_curve, roc_auc_score
```

2. Model Building

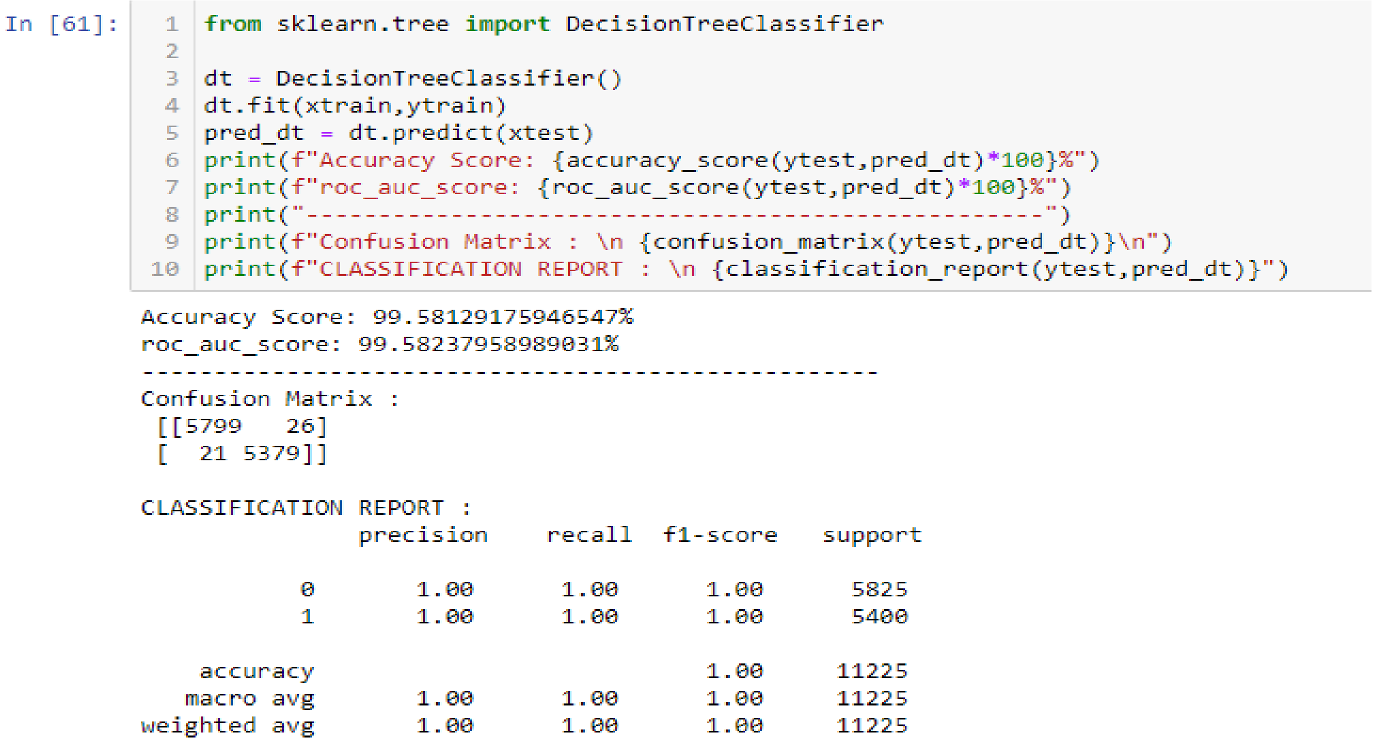
1) Logistic Regression:



```
In [60]:
LR.fit(xtrain,ytrain)

predlr = LR.predict(xtest)
print(f"Accuracy Score: {accuracy_score(ytest,predlr)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,predlr)*100}%")
print("----------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,predlr)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,predlr)}")
```

```
Accuracy Score: 99.15367483296214%
roc_auc_score: 99.163606739787%
---------------------------------------------------
Confusion Matrix :
 [[5761   64]
 [  31 5369]]

CLASSIFICATION REPORT :
               precision    recall  f1-score   support

           0       0.99      0.99      0.99      5825
           1       0.99      0.99      0.99      5400

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225
```

2) Decision Tree Classifier:

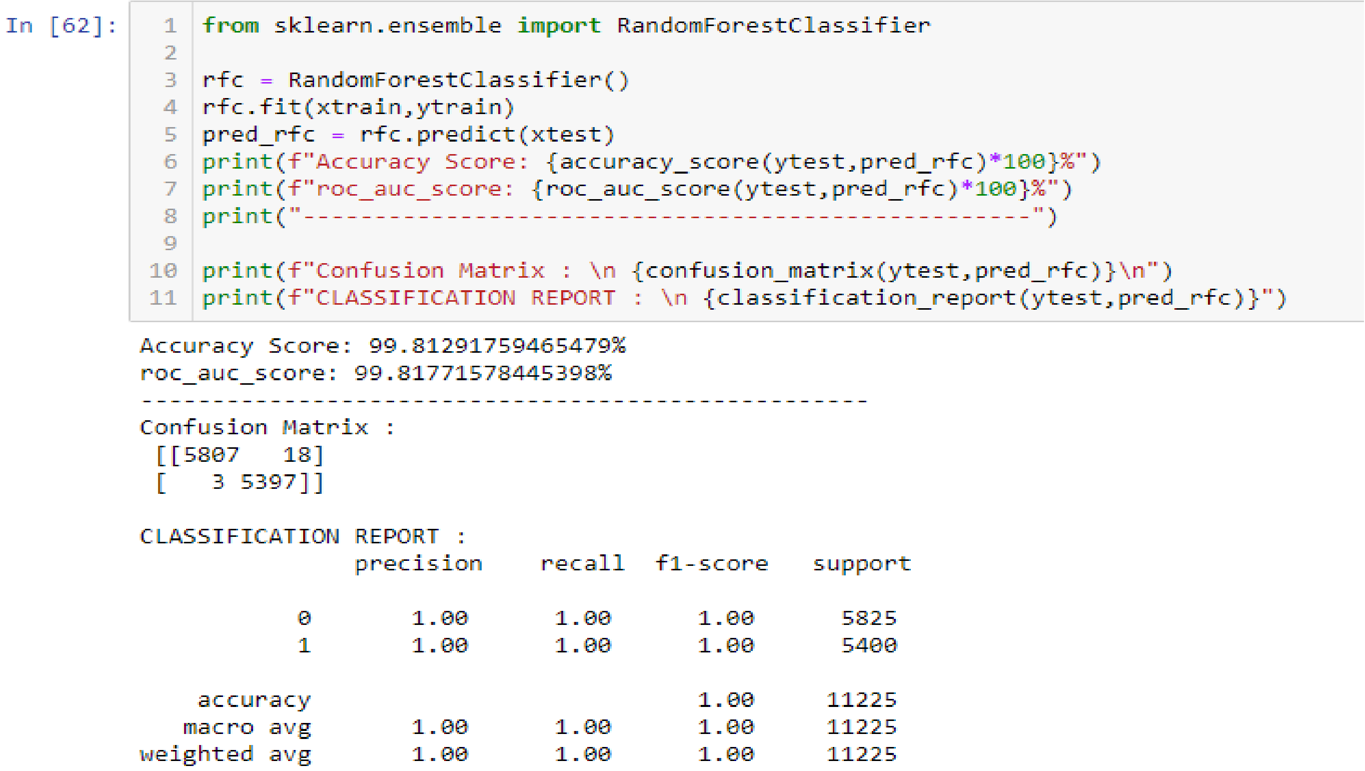


In [61]:

```python
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()
dt.fit(xtrain,ytrain)
pred_dt = dt.predict(xtest)
print(f"Accuracy Score: {accuracy_score(ytest,pred_dt)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_dt)*100}%")
print("---------------------------------------------------")
print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_dt)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_dt)}")
```

```
Accuracy Score: 99.58129175946547%
roc_auc_score: 99.58237958989031%
---------------------------------------------------
Confusion Matrix : 
 [[5799   26]
 [  21 5379]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

3) Random Forest Classifier:

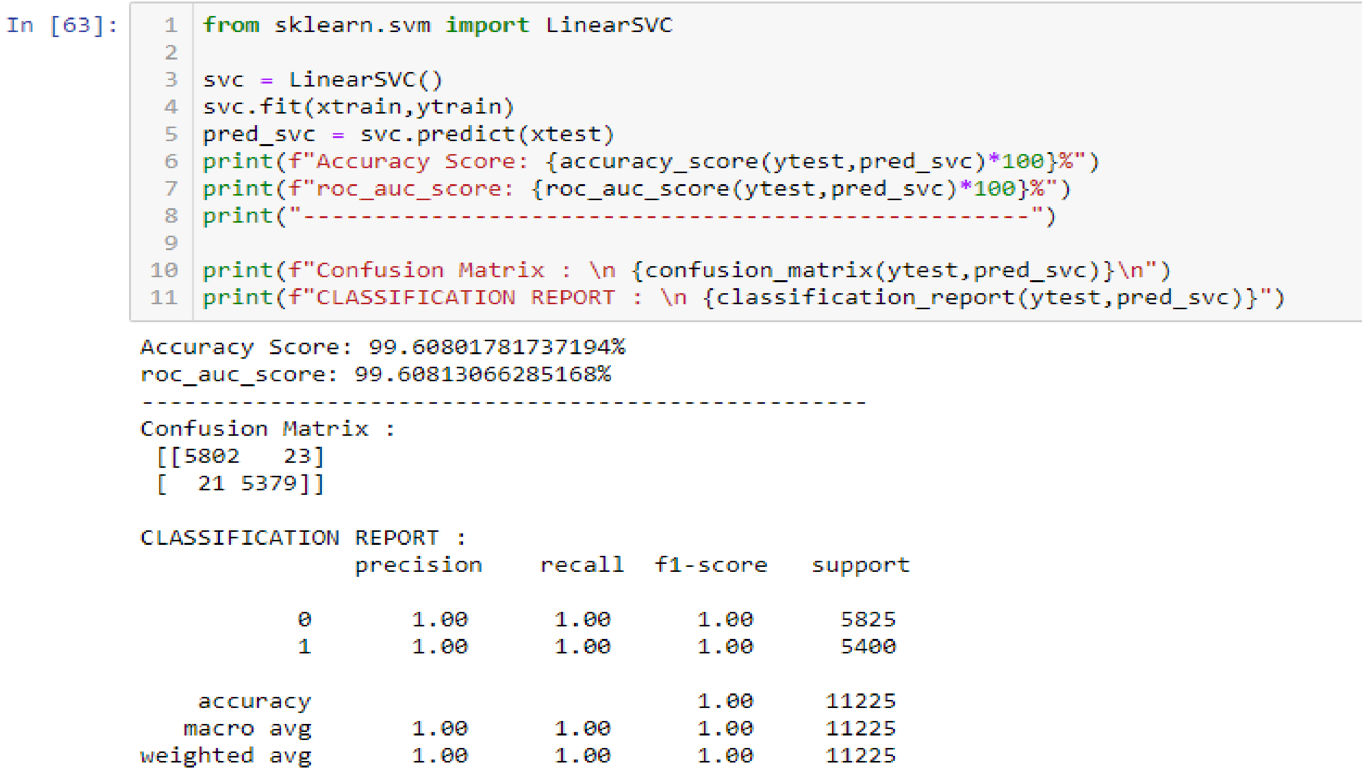


In [62]:

```python
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc.fit(xtrain,ytrain)
pred_rfc = rfc.predict(xtest)
print(f"Accuracy Score: {accuracy_score(ytest,pred_rfc)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_rfc)*100}%")
print("---------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_rfc)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_rfc)}")
```

```
Accuracy Score: 99.81291759465479%
roc_auc_score: 99.81771578445398%
---------------------------------------------------
Confusion Matrix : 
 [[5807   18]
 [   3 5397]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

4) Linear Support Vector Machine Classifier:

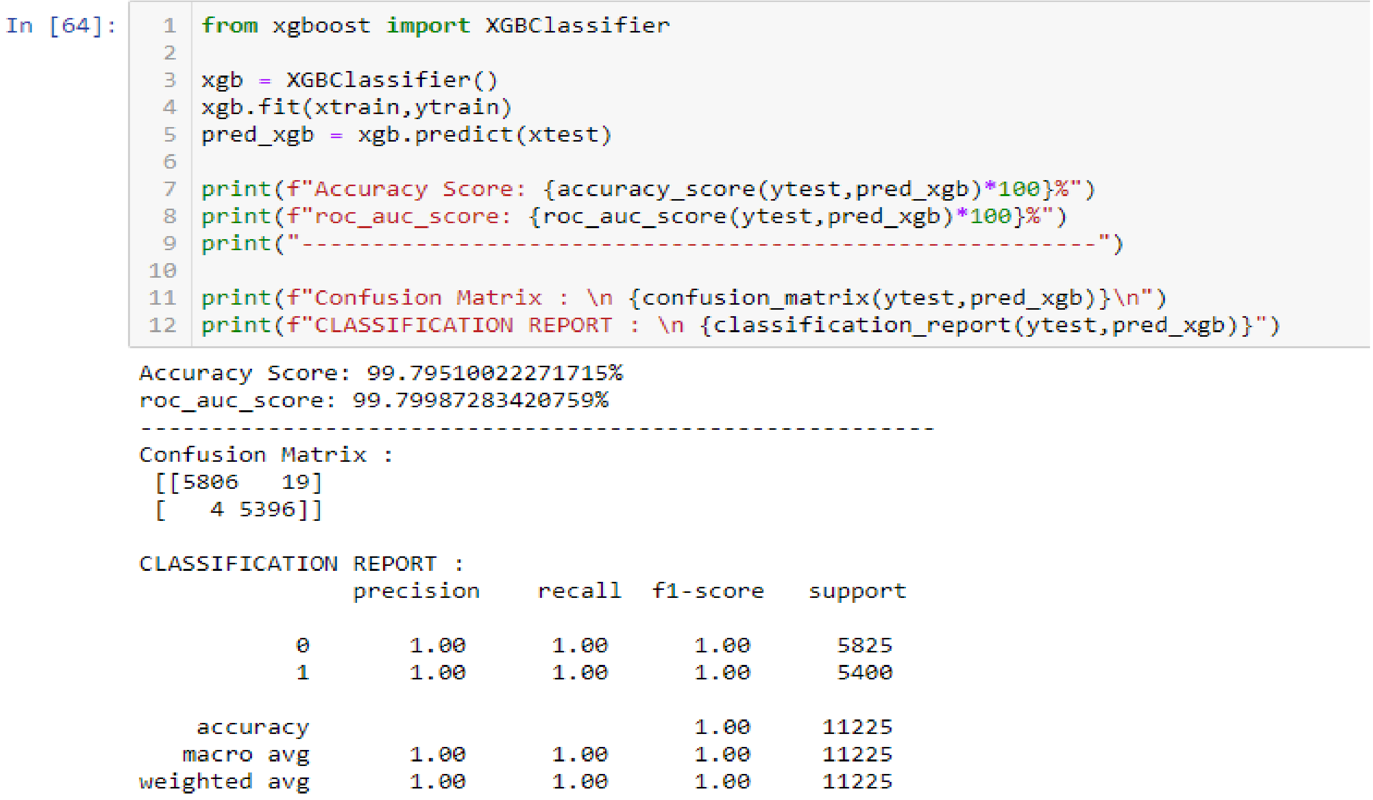


In [63]:

```python
from sklearn.svm import LinearSVC

svc = LinearSVC()
svc.fit(xtrain,ytrain)
pred_svc = svc.predict(xtest)
print(f"Accuracy Score: {accuracy_score(ytest,pred_svc)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_svc)*100}%")
print("---------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_svc)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_svc)}")
```

```
Accuracy Score: 99.60801781737194%
roc_auc_score: 99.60813066285168%
---------------------------------------------------
Confusion Matrix : 
 [[5802   23]
 [  21 5379]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

5) XGB Classifier:

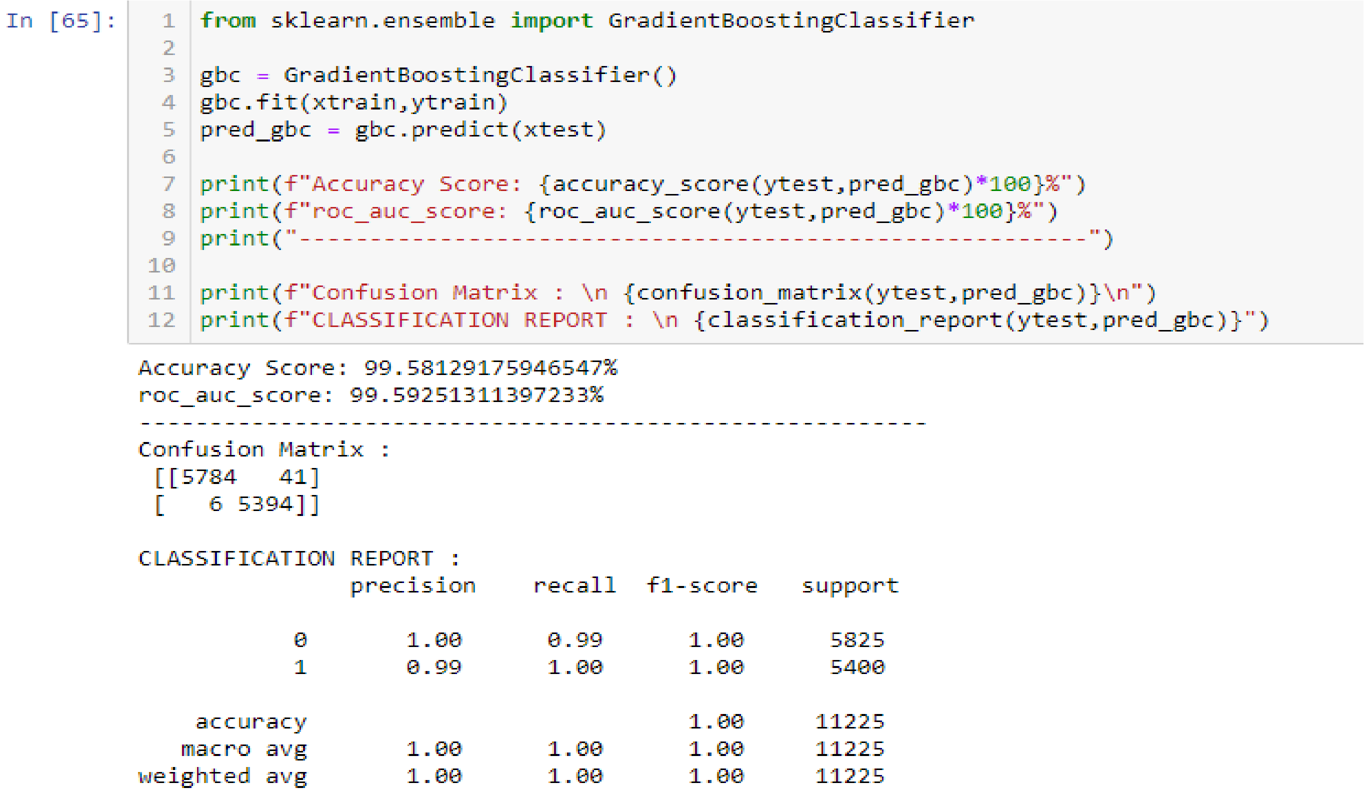


In [64]:

```
from xgboost import XGBClassifier

xgb = XGBClassifier()
xgb.fit(xtrain,ytrain)
pred_xgb = xgb.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_xgb)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_xgb)*100}%")
print("--------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_xgb)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_xgb)}")
```

```
Accuracy Score: 99.79510022271715%
roc_auc_score: 99.79987283420759%
--------------------------------------------------------
Confusion Matrix : 
 [[5806   19]
 [   4 5396]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

6) Gradient Boosting Classifier:

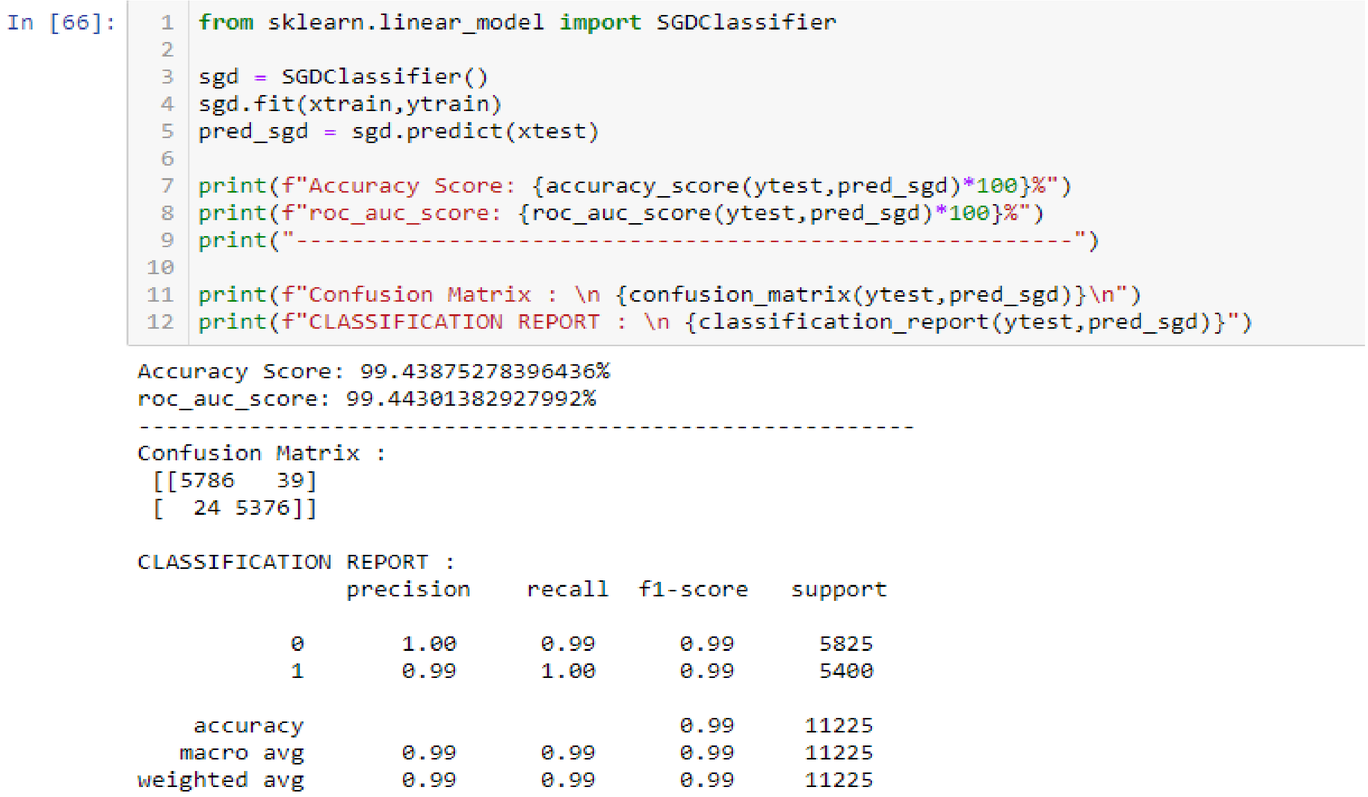


In [65]:

```python
from sklearn.ensemble import GradientBoostingClassifier

gbc = GradientBoostingClassifier()
gbc.fit(xtrain,ytrain)
pred_gbc = gbc.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_gbc)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_gbc)*100}%")
print("---------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_gbc)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_gbc)}")
```

```
Accuracy Score: 99.58129175946547%
roc_auc_score: 99.59251311397233%
--------------------------------------------------------
Confusion Matrix :
 [[5784   41]
 [   6 5394]]

CLASSIFICATION REPORT :
               precision    recall  f1-score   support

           0       1.00      0.99      1.00      5825
           1       0.99      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

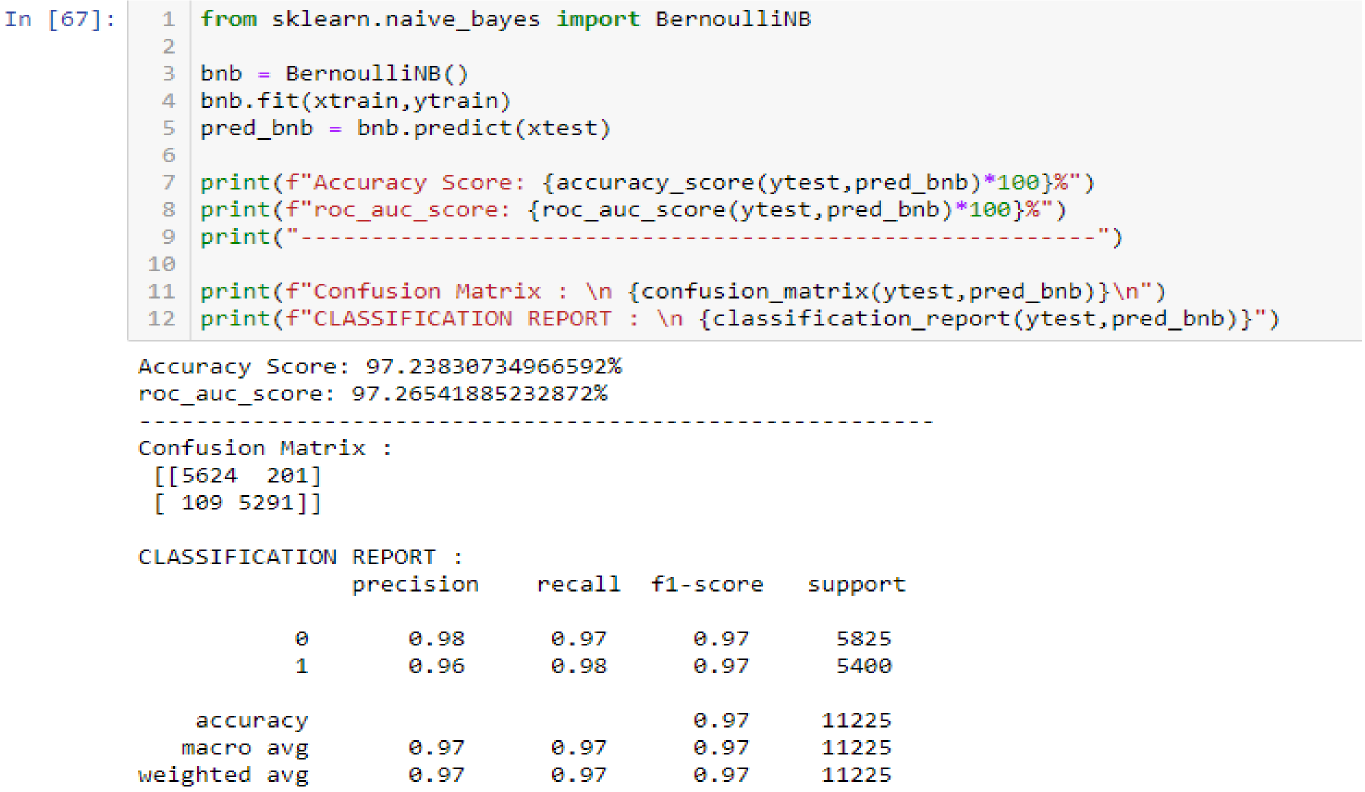
7) Stochastic Gradient Descent Classifier:



```
In [66]:
from sklearn.linear_model import SGDClassifier

sgd = SGDClassifier()
sgd.fit(xtrain,ytrain)
pred_sgd = sgd.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_sgd)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_sgd)*100}%")
print("--------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_sgd)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_sgd)}")
```

```
Accuracy Score: 99.43875278396436%
roc_auc_score: 99.44301382927992%
--------------------------------------------------------
Confusion Matrix :
 [[5786   39]
 [  24 5376]]

CLASSIFICATION REPORT :
               precision    recall  f1-score   support

           0       1.00      0.99      0.99      5825
           1       0.99      1.00      0.99      5400

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225
```

8) Bernoulli Naive Bayes Classifier:

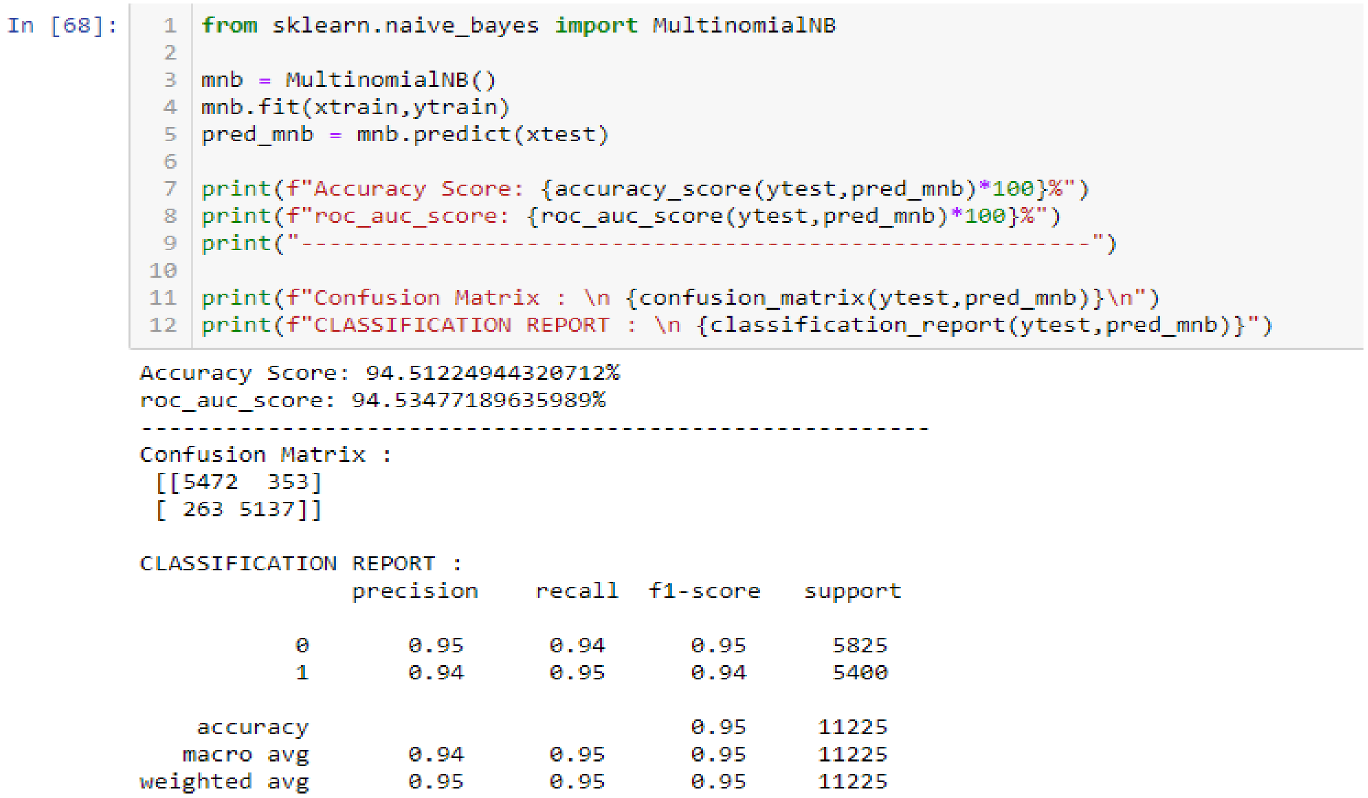


In [67]:

```python
from sklearn.naive_bayes import BernoulliNB

bnb = BernoulliNB()
bnb.fit(xtrain,ytrain)
pred_bnb = bnb.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_bnb)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_bnb)*100}%")
print("--------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_bnb)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_bnb)}")
```

```
Accuracy Score: 97.23830734966592%
roc_auc_score: 97.26541885232872%
--------------------------------------------------------
Confusion Matrix : 
 [[5624  201]
 [ 109 5291]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       0.98      0.97      0.97      5825
           1       0.96      0.98      0.97      5400

    accuracy                           0.97     11225
   macro avg       0.97      0.97      0.97     11225
weighted avg       0.97      0.97      0.97     11225
```

9) Multinomial Naive Bayes Classifier:

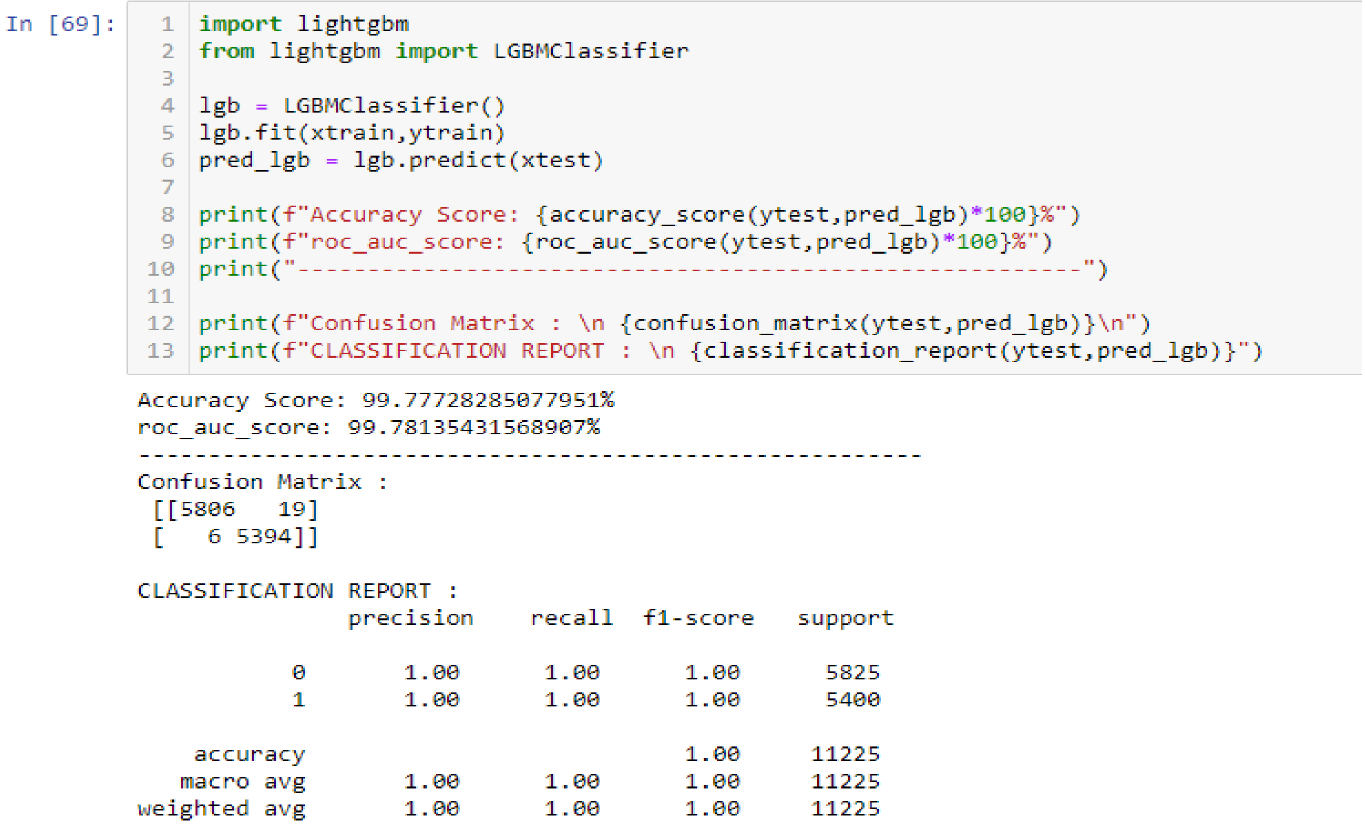


In [68]:

```
from sklearn.naive_bayes import MultinomialNB

mnb = MultinomialNB()
mnb.fit(xtrain,ytrain)
pred_mnb = mnb.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_mnb)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_mnb)*100}%")
print("---------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_mnb)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_mnb)}")
```

```
Accuracy Score: 94.51224944320712%
roc_auc_score: 94.53477189635989%
--------------------------------------------------------
Confusion Matrix :
 [[5472  353]
 [ 263 5137]]

CLASSIFICATION REPORT :
               precision    recall  f1-score   support

           0       0.95      0.94      0.95      5825
           1       0.94      0.95      0.94      5400

    accuracy                           0.95     11225
   macro avg       0.94      0.95      0.95     11225
weighted avg       0.95      0.95      0.95     11225
```

10) Light Gradient Boosting Machine Classifier:

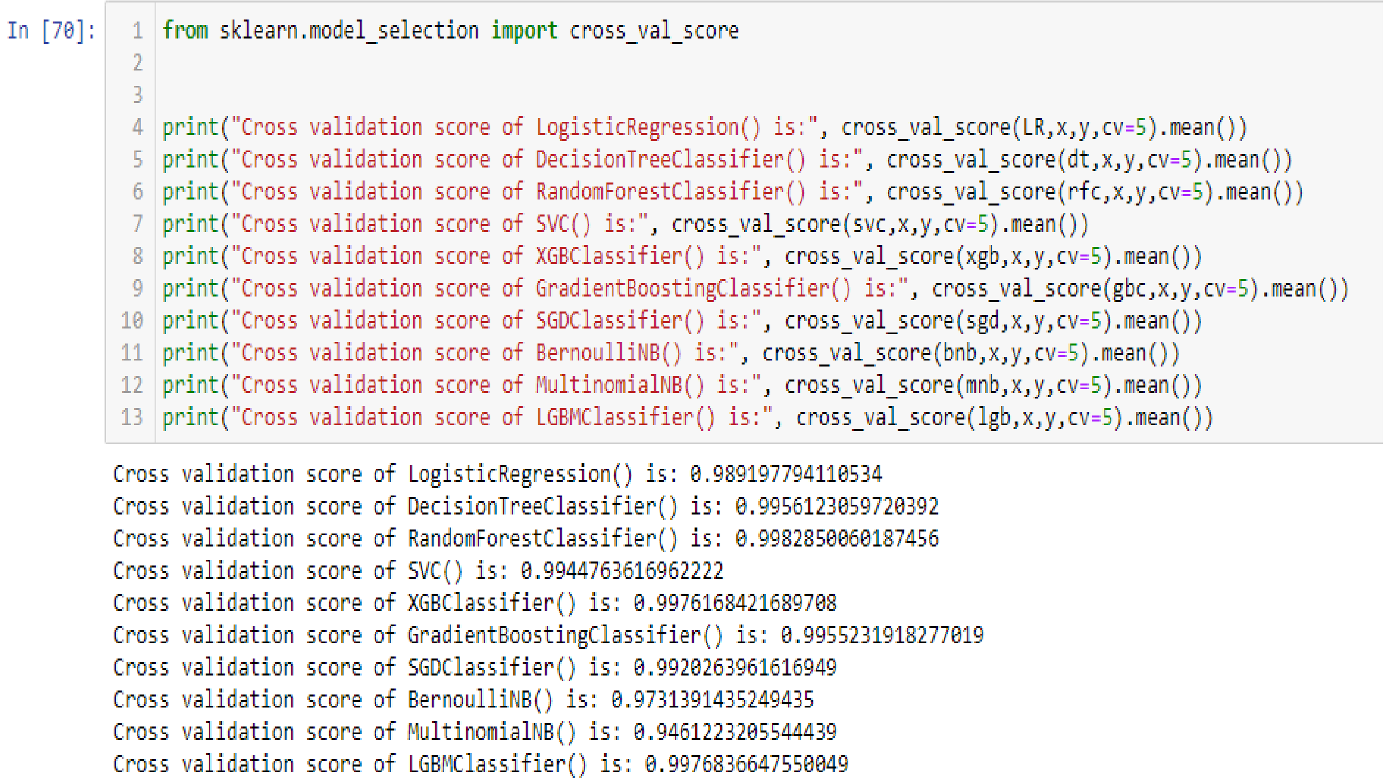


In [69]:

```python
import lightgbm
from lightgbm import LGBMClassifier

lgb = LGBMClassifier()
lgb.fit(xtrain,ytrain)
pred_lgb = lgb.predict(xtest)

print(f"Accuracy Score: {accuracy_score(ytest,pred_lgb)*100}%")
print(f"roc_auc_score: {roc_auc_score(ytest,pred_lgb)*100}%")
print("--------------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred_lgb)}\n")
print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred_lgb)}")
```

```
Accuracy Score: 99.77728285077951%
roc_auc_score: 99.78135431568907%
--------------------------------------------------------
Confusion Matrix : 
 [[5806   19]
 [   6 5394]]

CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

Cross Validation



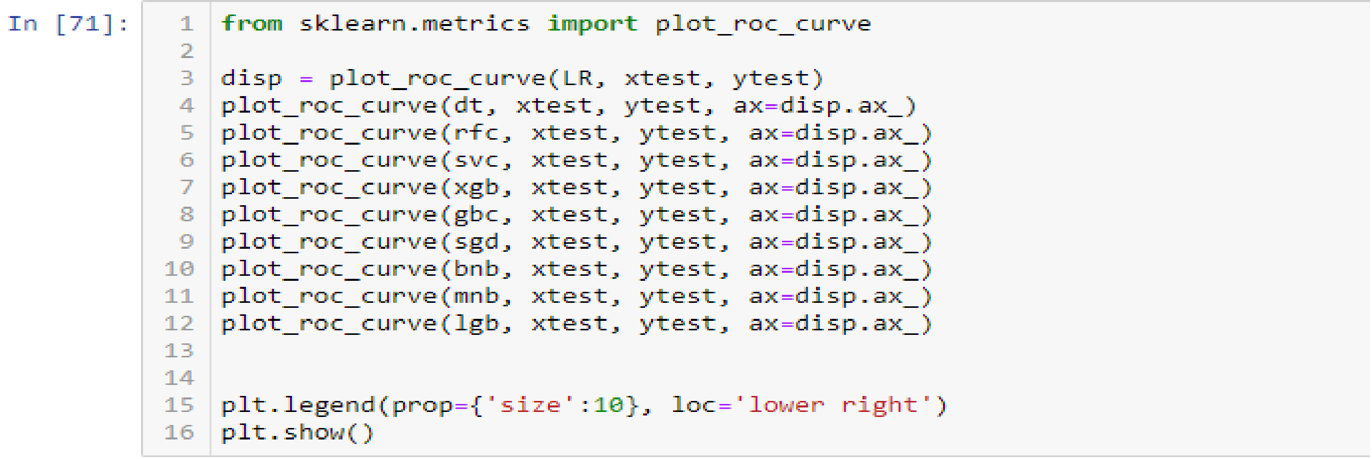
In [70]:

```
from sklearn.model_selection import cross_val_score


print("Cross validation score of LogisticRegression() is:", cross_val_score(LR,x,y,cv=5).mean())
print("Cross validation score of DecisionTreeClassifier() is:", cross_val_score(dt,x,y,cv=5).mean())
print("Cross validation score of RandomForestClassifier() is:", cross_val_score(rfc,x,y,cv=5).mean())
print("Cross validation score of SVC() is:", cross_val_score(svc,x,y,cv=5).mean())
print("Cross validation score of XGBClassifier() is:", cross_val_score(xgb,x,y,cv=5).mean())
print("Cross validation score of GradientBoostingClassifier() is:", cross_val_score(gbc,x,y,cv=5).mean())
print("Cross validation score of SGDClassifier() is:", cross_val_score(sgd,x,y,cv=5).mean())
print("Cross validation score of BernoulliNB() is:", cross_val_score(bnb,x,y,cv=5).mean())
print("Cross validation score of MultinomialNB() is:", cross_val_score(mnb,x,y,cv=5).mean())
print("Cross validation score of LGBMClassifier() is:", cross_val_score(lgb,x,y,cv=5).mean())
```

```
Cross validation score of LogisticRegression() is: 0.989197794110534
Cross validation score of DecisionTreeClassifier() is: 0.9956123059720392
Cross validation score of RandomForestClassifier() is: 0.9982850060187456
Cross validation score of SVC() is: 0.9944763616962222
Cross validation score of XGBClassifier() is: 0.9976168421689708
Cross validation score of GradientBoostingClassifier() is: 0.9955231918277019
Cross validation score of SGDClassifier() is: 0.9920263961616949
Cross validation score of BernoulliNB() is: 0.9731391435249435
Cross validation score of MultinomialNB() is: 0.9461223205544439
Cross validation score of LGBMClassifier() is: 0.9976836647550049
```

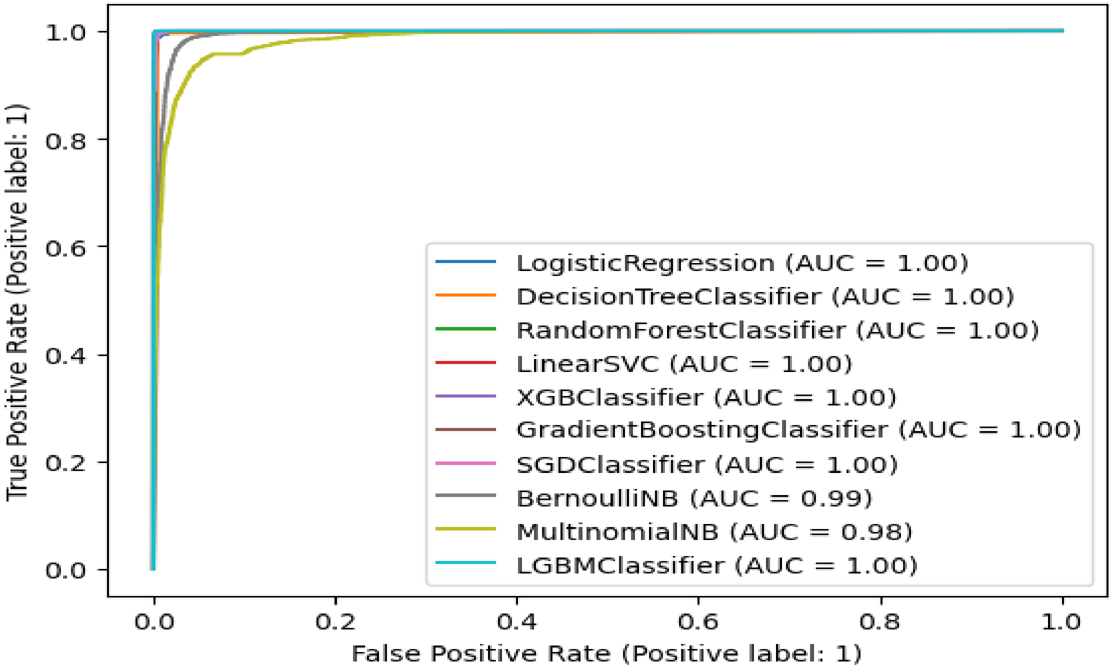
As we can see, Light Gradient Boosting Machine Classifier is the best model.

Plotting ROC-AUC curves



In [71]:

```python
from sklearn.metrics import plot_roc_curve

disp = plot_roc_curve(LR, xtest, ytest)
plot_roc_curve(dt, xtest, ytest, ax=disp.ax_)
plot_roc_curve(rfc, xtest, ytest, ax=disp.ax_)
plot_roc_curve(svc, xtest, ytest, ax=disp.ax_)
plot_roc_curve(xgb, xtest, ytest, ax=disp.ax_)
plot_roc_curve(gbc, xtest, ytest, ax=disp.ax_)
plot_roc_curve(sgd, xtest, ytest, ax=disp.ax_)
plot_roc_curve(bnb, xtest, ytest, ax=disp.ax_)
plot_roc_curve(mnb, xtest, ytest, ax=disp.ax_)
plot_roc_curve(lgb, xtest, ytest, ax=disp.ax_)


plt.legend(prop={'size':10}, loc='lower right')
plt.show()
```



Hyperparameter Tuning

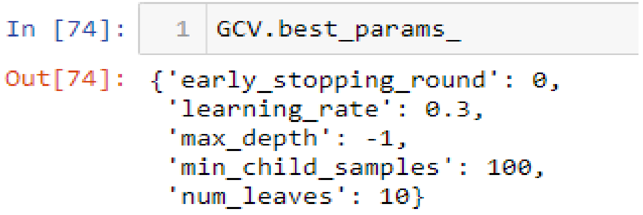
In [72]:

```
# Now, let's perform Hyperparameter Tuning for Light Gradient Boosting Machine Classifier

from sklearn.model_selection import GridSearchCV


parameters = {'num_leaves': (10, 20, 30),
              'min_child_samples': (100, 200, 300),
              'max_depth': (-1, 0, 1),
              'learning_rate': (0.1, 0.2, 0.3),
              'early_stopping_round': (0, 1)}


GCV = GridSearchCV(LGBMClassifier(), parameters, cv=5)

GCV.fit(xtrain, ytrain)
```

Out[72]:

```
GridSearchCV(cv=5, estimator=LGBMClassifier(),
             param_grid={'early_stopping_round': (0, 1),
                         'learning_rate': (0.1, 0.2, 0.3),
                         'max_depth': (-1, 0, 1),
                         'min_child_samples': (100, 200, 300),
                         'num_leaves': (10, 20, 30)})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

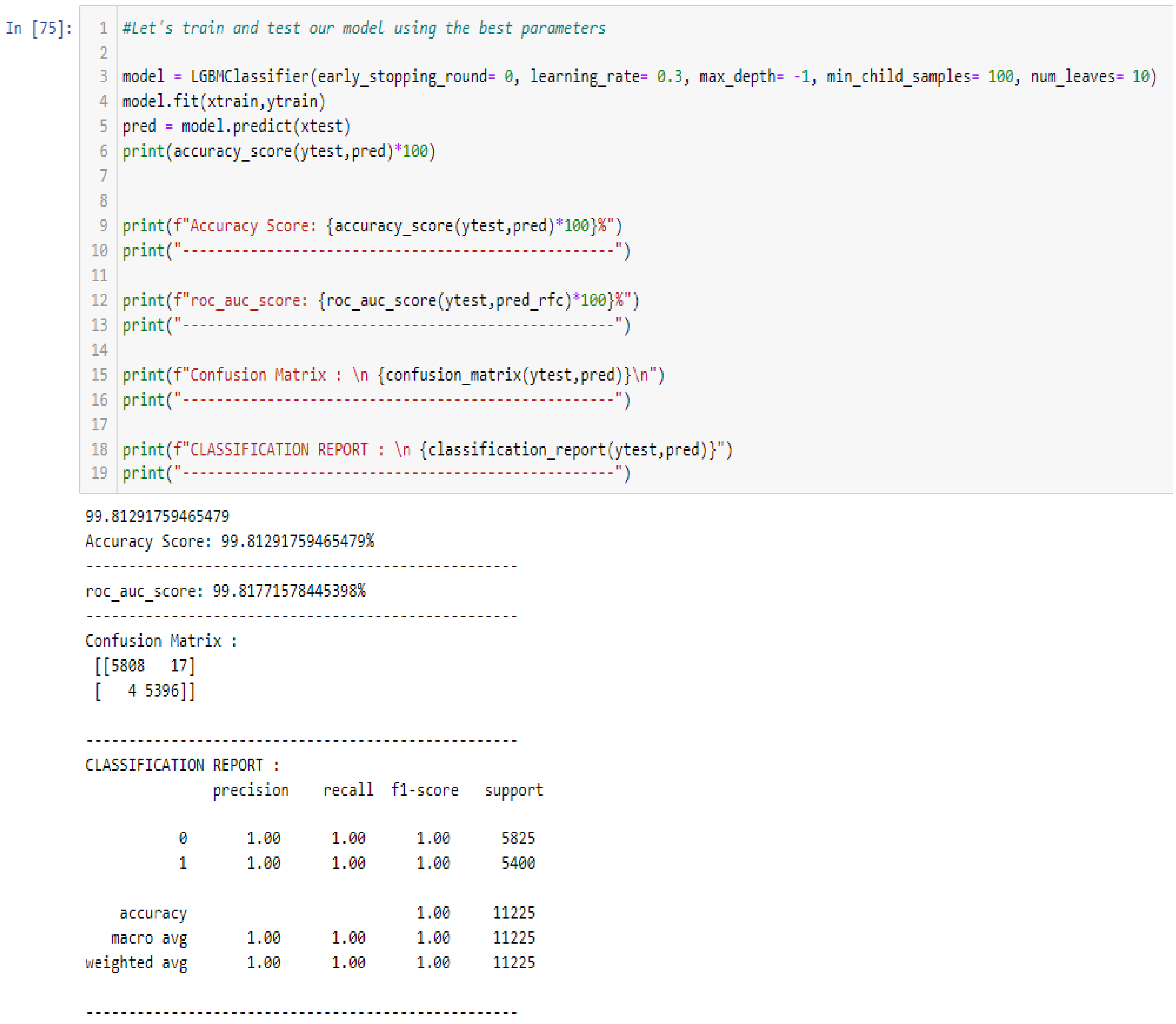


In [74]:

```
GCV.best_params_
```

Out[74]:

```
{'early_stopping_round': 0,
 'learning_rate': 0.3,
 'max_depth': -1,
 'min_child_samples': 100,
 'num_leaves': 10}
```



```
In [75]:
#Let's train and test our model using the best parameters

model = LGBMClassifier(early_stopping_round= 0, learning_rate= 0.3, max_depth= -1, min_child_samples= 100, num_leaves= 10)
model.fit(xtrain,ytrain)
pred = model.predict(xtest)
print(accuracy_score(ytest,pred)*100)


print(f"Accuracy Score: {accuracy_score(ytest,pred)*100}%")
print("--------------------------------------------------")

print(f"roc_auc_score: {roc_auc_score(ytest,pred_rfc)*100}%")
print("--------------------------------------------------")

print(f"Confusion Matrix : \n {confusion_matrix(ytest,pred)}\n")
print("--------------------------------------------------")

print(f"CLASSIFICATION REPORT : \n {classification_report(ytest,pred)}")
print("--------------------------------------------------")
```

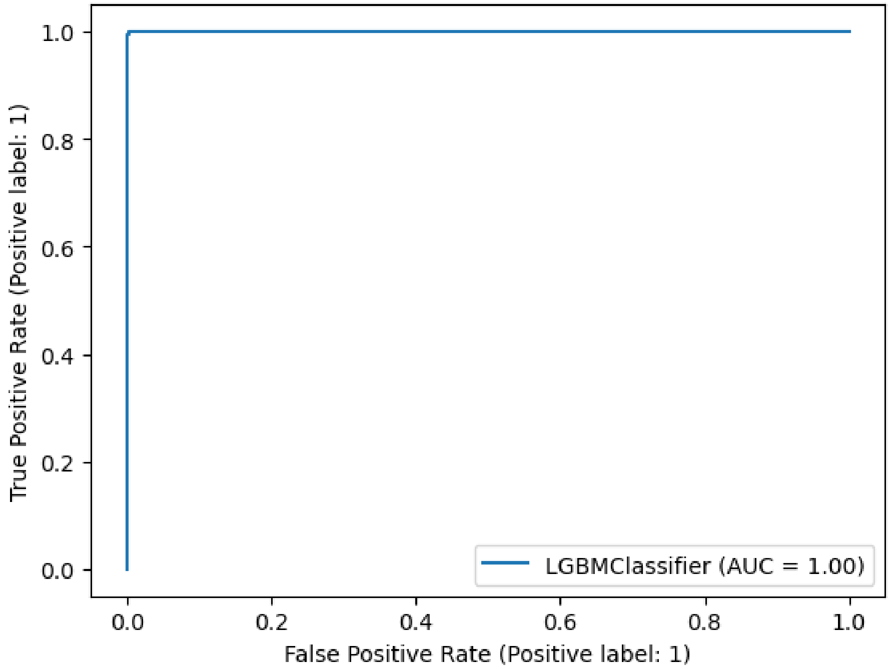
```
99.81291759465479
Accuracy Score: 99.81291759465479%
--------------------------------------------------
roc_auc_score: 99.81771578445398%
--------------------------------------------------
Confusion Matrix : 
 [[5808   17]
 [   4 5396]]

--------------------------------------------------
CLASSIFICATION REPORT : 
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      5825
           1       1.00      1.00      1.00      5400

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225

--------------------------------------------------
```

After Hyperparameter Tuning, we got an accuracy score of 99.81%.

ROC curve for final model

In [76]:

```
#Lets check our model performance after hyperparameter tuning

plot_roc_curve(model, xtest, ytest)
plt.show()
```



Saving the model

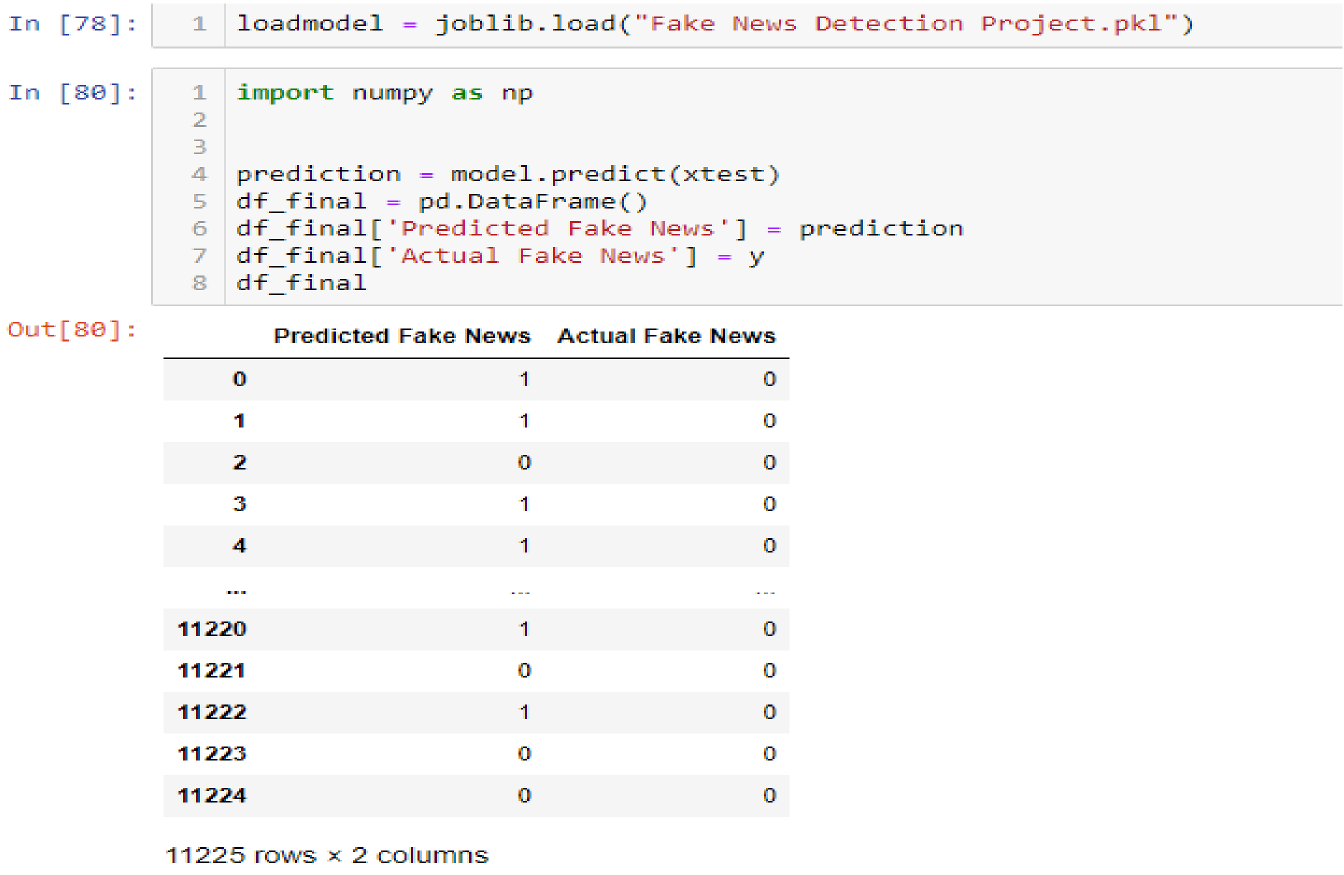
In [77]:

```
import joblib

joblib.dump(model,"Fake News Detection Project.pkl")
```

Out[77]:

```
['Fake News Detection Project.pkl']
```

Loading the model

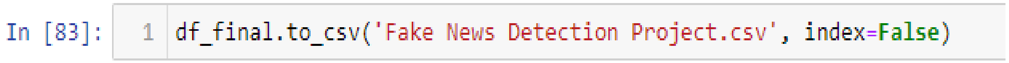


```
In [78]:  loadmodel = joblib.load("Fake News Detection Project.pkl")
```

```
In [80]:  import numpy as np


          prediction = model.predict(xtest)
          df_final = pd.DataFrame()
          df_final['Predicted Fake News'] = prediction
          df_final['Actual Fake News'] = y
          df_final
```

```
Out[80]:
              Predicted Fake News   Actual Fake News
          0                     1                  0
          1                     1                  0
          2                     0                  0
          3                     1                  0
          4                     1                  0
        ...                   ...                ...
      11220                     1                  0
      11221                     0                  0
      11222                     1                  0
      11223                     0                  0
      11224                     0                  0

     11225 rows × 2 columns
```

Converting the dataframe into CSV format and saving it



```
In [83]:  df_final.to_csv('Fake News Detection Project.csv', index=False)
```

3.6 Interpretation of the Results

- ✓ By applying pre-processing techniques, I have converted the text to lower case, removed Punctations and stop-words. Then, I have converted words to the most suitable base form by using lemmatization.
- ✓ Natural Language Processing and Machine Learning is used in this project.
- ✓ I have used 10 different classification algorithms to build my model. By looking at the accuracy score and Cross Validation score, I found that Light Gradient Boosting Machine Classifier is the best model.

4. Conclusion

4.1 Key Findings and Conclusions of the Study

- ✓ In this project we have detected which news is fake and which news is true.
- ✓ By carrying out different EDA steps, I have analyzed the text.
- ✓ I have checked frequently occurring words in our data, as well as rarely occurring words.
- ✓ After all these steps, I have used 10 different classification algorithms to build my model. By looking at the accuracy score and Cross Validation score, I found that Light Gradient Boosting Machine Classifier is the best model.
- ✓ Then, by performing hyperparameter tuning, I got the best parameters for our final model. And finally, I got improved accuracy score for our final model.

4.2 Learning Outcomes of the Study in respect of Data Science

- ✓ This project has demonstrated the importance of NLP.
- ✓ Through different powerful tools of visualization, we were able to analyse and interpret the huge data and with the help of count plot & word cloud, I am able to see the distribution of fake and true news.
- ✓ Through data cleaning we were able to remove unnecessary columns, values, stop-words and punctuation from our dataset due to which our model would have suffered from overfitting or underfitting.

The few challenges while working on this project were:

- ✓ Using NLP to find punctuations & stop words, it took time in giving the result.
- ✓ The data set took time to run some algorithms & to check the crossvalidation score.

4.3 Limitations of this work and Scope for Future Work

In the future, a web-based GUI can be created for the proposed fake news detection system to classify the news as fake or real on real-time social media platforms such as Facebook, Instagram, Twitter, WhatsApp, etc. Also, the annotated dataset in the sequence of images (with textual content written on them) will be collected and maintained from Facebook and Reddit platforms. The annotated dataset is often used for detecting fake images within the future as no such dataset is out there at the present. The proposed system has the potential to provide an impulse to various emerging applications such as controlling the spread of fake news during elections, terrorism, natural calamities, crimes for the betterment of society. In the future, the efficiency and accuracy of the prototype can be enhanced to a certain level, and also enhance the user interface of the proposed model.