FLIP ROBO

FAKE NEWS DETECTOR

Submitted by:

MD RASHIDUNNABI

1. Introduction

In our daily routine, we receive news in a variety of ways, but it may be difficult to determine which sources are legitimate and which are not.

How much confidence do you have in the news that you read on the Internet?

Almost all of the news that we consume is not true. A person's ideas or thoughts might alter when they consume false news that they believe to be genuine.

When all news is false, how can we detect the difference?

Text-based news will be the emphasis of this article, as we attempt to develop a model that will allow us to determine whether or not a specific piece of news is true or not.

Let's define a few terms first.

2. Terminologies

2.1 Fake News

Essentially sensationalized reporting, counterfeit news is most often distributed through web-based media and other online media.

When it comes to political planning, this is a common practice for promoting or forcing particular types of thinking or falsely promoting items.

They may contain fabricated and misrepresented cases, or they may be virtualized by a computer program.

2.2 Tfidf Vectorizer

TF (Term Frequency): As a result of the document's high frequency of terms, the term frequency is given. It implies that this word appears the most in this part compared to other terms if you receive the highest values in this area. A very good match is shown when you receive the message "Word is parts of speech word".

IDF (Inverse Document Frequency): Many terms occur in a single text, but are also found in other documents that are unrelated. Indicator of term importance (IDF) is a measure of a term's importance in the corpus.

Word collection A matrix containing TF-IDF characteristics will be created from documents using TfidfVectorizer.

3. Project

A machine learning model is required to properly classify news as authentic or false.

We will apply 'TfidfVectorizer' in our news data, which we will acquire from internet media, to detect false and true news.

Initializing the classifier, transforming and fitting the model will follow after the first step. Our model will be evaluated using a corresponding performance matrix. Once we've calculated the performance matrices, we'll be able to observe how well our model performs in real-world scenarios.

A step-by-step explanation will be provided on how to utilize these technologies in the field.

3.1 Data Analysis

Here I will explain the dataset.

There are 6 columns in the dataset provided to you. The description of each of the column is given below:

"id": Unique id of each news article

"headline": It is the title of the news.

"news": It contains the full text of the news article

"Unnamed:0": It is a serial number

"written_by": It represents the author of the news article

"label": It tells whether the news is fake (1) or not fake (0).

3.2 Libraries

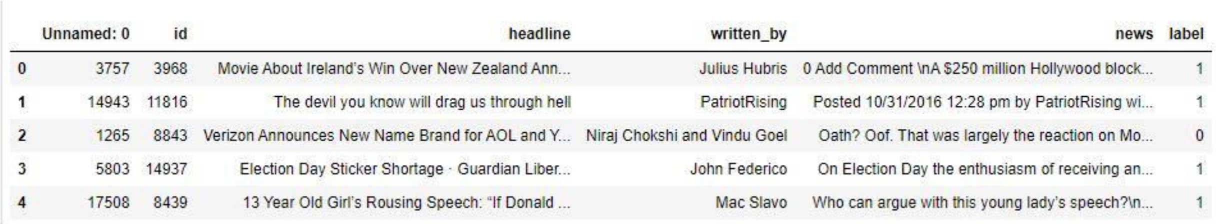
The very basic data science libraries are sklearn, pandas, NumPy e.t.c and some specific libraries such as transformers.

```
import pandas as pd


import numpy as np
import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfTransformer

from sklearn import feature_extraction, linear_model, model_selection, preprocessing

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

from sklearn.pipeline import Pipeline
```

3.3 Read dataset from CSV File

```
df=pd.read_csv('train_news.csv')

df.head()
```

output:-



	Unnamed: 0	id	headline	written_by	news	label
0	3757	3968	Movie About Ireland's Win Over New Zealand Ann...	Julius Hubris	0 Add Comment \nA $250 million Hollywood block...	1
1	14943	11816	The devil you know will drag us through hell	PatriotRising	Posted 10/31/2016 12:28 pm by PatriotRising wi...	1
2	1265	8843	Verizon Announces New Name Brand for AOL and Y...	Niraj Chokshi and Vindu Goel	Oath? Oof. That was largely the reaction on Mo...	0
3	5803	14937	Election Day Sticker Shortage · Guardian Liber...	John Federico	On Election Day the enthusiasm of receiving an...	1
4	17508	8439	13 Year Old Girl's Rousing Speech: "If Donald ...	Mac Slavo	Who can argue with this young lady's speech?\n...	1

Before moving on, we must determine whether our dataset contains a null value.

```
df.isnull().sum()
```

There is null value in this dataset. That's why I have replaced the null values.

```
df = df.fillna(' ')
```

3.4 Data Preprocessing

When it comes to data processing, we'll focus on the text column of this data, which contains the actual news. This text column will be modified to extract more information in order to improve the predictability of the model as a whole. We will use a package called nltk to extract information from the text column.

Removing Stopwords, Tokenization, and Lemmatization are 'nltk' library capabilities that will be used in this example. These three examples will allow us to examine each of these functions one at a time. It is my hope that after reading this, you would have a better grasp of how to extract information from text columns.

3.4.1 Removing Stopwords:-

The terms that are commonly employed to connect words in any language or to indicate the tense of sentences are known as conjunctions. Because these words do not contribute much significance to the context of a phrase, we can still comprehend the context even after deleting the stopwords.

3.4.2 Tokenization:-

When text is tokenized, it's broken down into smaller bits called tokens.

A token is a representation of a word, special character, or number in NLP.

When a piece of code is tokenized, it's broken down into smaller parts called "tokens."

```
from nltk.tokenize import word_tokenize
```

```
# Function to plot the confusion matrix (code from https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html)

from sklearn import metrics

import itertools



def plot_confusion_matrix(cm, classes,

                          normalize=False,

                          title='Confusion matrix',

                          cmap=plt.cm.Blues):



    plt.imshow(cm, interpolation='nearest', cmap=cmap)

    plt.title(title)

    plt.colorbar()

    tick_marks = np.arange(len(classes))

    plt.xticks(tick_marks, classes, rotation=45)

    plt.yticks(tick_marks, classes)
```

```
if normalize:
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    print("Normalized confusion matrix")
else:
    print('Confusion matrix, without normalization')


thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, cm[i, j],
             horizontalalignment="center",
             color="white" if cm[i, j] > thresh else "black")


plt.tight_layout()
plt.ylabel('True label')
```

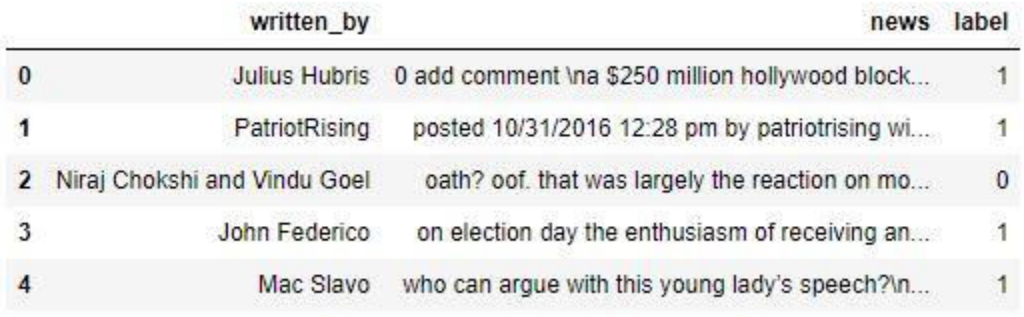
```
    plt.xlabel('Predicted label')
```

3.5. Data cleaning and preparation :-

1. **Convert to lowercase**

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


df.head()
```

Output :

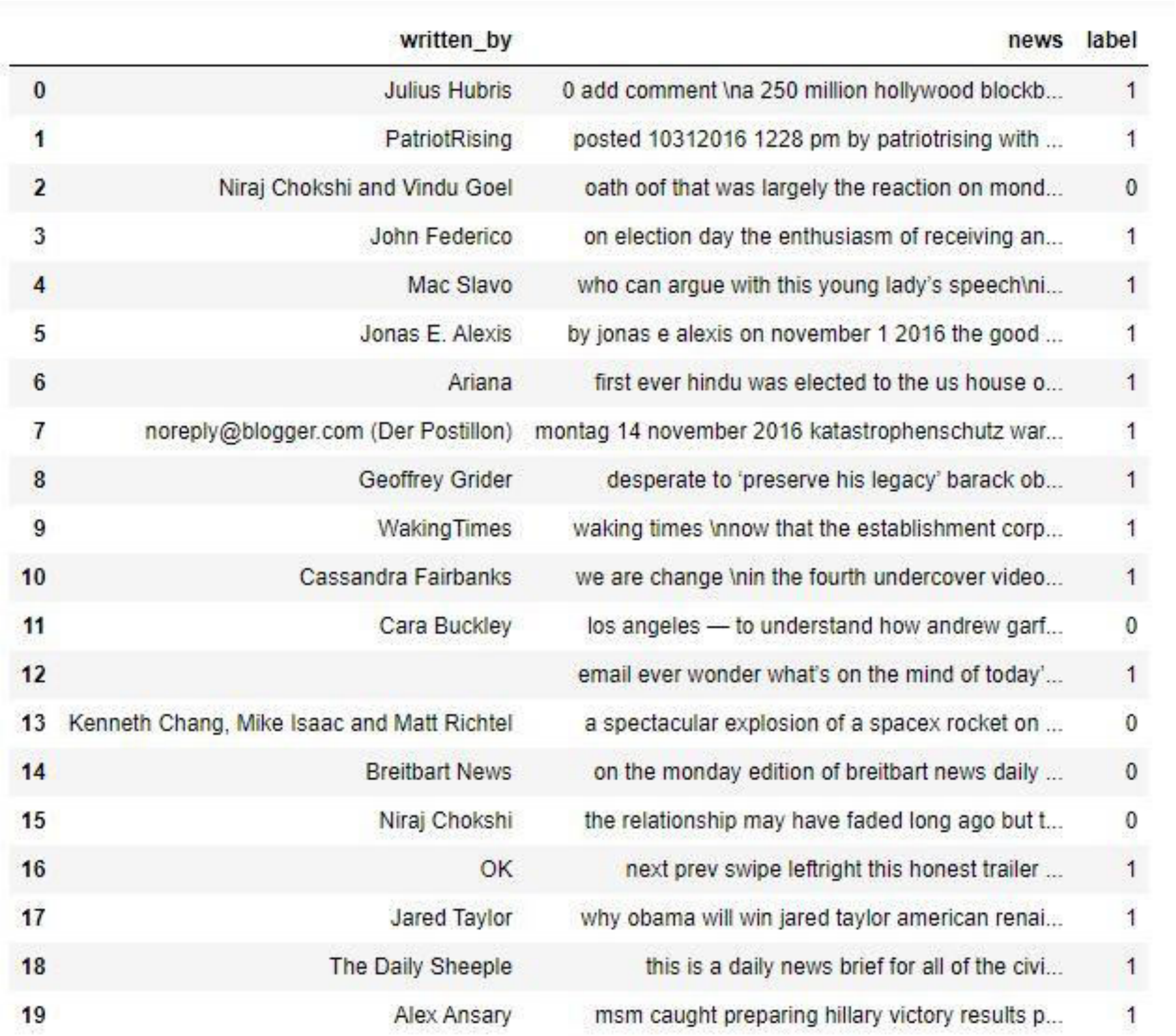


	written_by	news	label
0	Julius Hubris	0 add comment \na $250 million hollywood block...	1
1	PatriotRising	posted 10/31/2016 12:28 pm by patriotrising wi...	1
2	Niraj Chokshi and Vindu Goel	oath? oof. that was largely the reaction on mo...	0
3	John Federico	on election day the enthusiasm of receiving an...	1
4	Mac Slavo	who can argue with this young lady's speech?\n...	1

2. **Remove punctuation**

```
import string




def punctuation_removal(news):


    all_list = [char for char in news if char not in string.punctuation]


    clean_str = ''.join(all_list)
    return clean_str




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

Output :

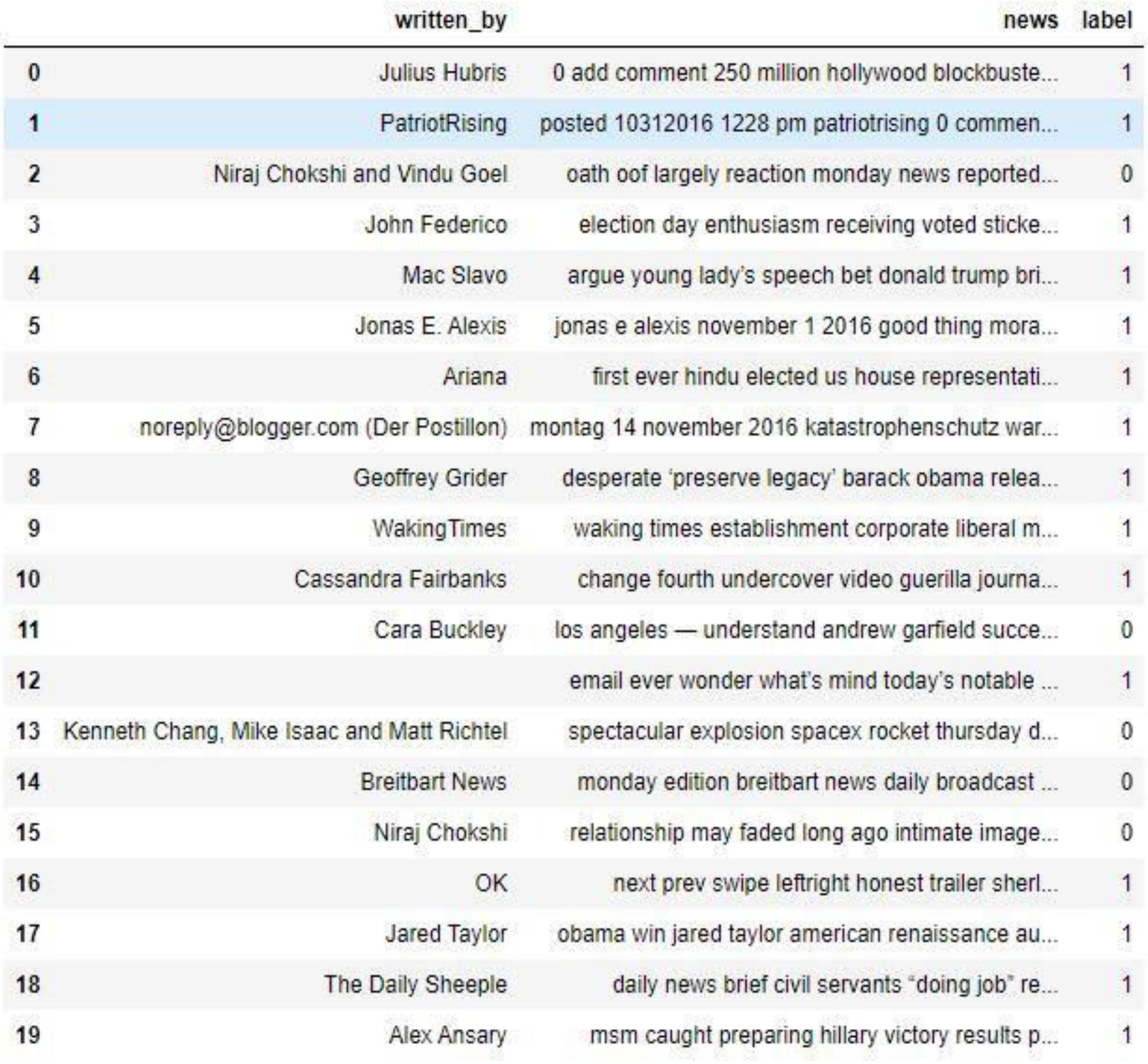


	written_by	news	label
0	Julius Hubris	0 add comment \na 250 million hollywood blockb...	1
1	PatriotRising	posted 10312016 1228 pm by patriotrising with ...	1
2	Niraj Chokshi and Vindu Goel	oath oof that was largely the reaction on mond...	0
3	John Federico	on election day the enthusiasm of receiving an...	1
4	Mac Slavo	who can argue with this young lady’s speech\ni...	1
5	Jonas E. Alexis	by jonas e alexis on november 1 2016 the good ...	1
6	Ariana	first ever hindu was elected to the us house o...	1
7	noreply@blogger.com (Der Postillon)	montag 14 november 2016 katastrophenschutz war...	1
8	Geoffrey Grider	desperate to ‘preserve his legacy’ barack ob...	1
9	WakingTimes	waking times \nnow that the establishment corp...	1
10	Cassandra Fairbanks	we are change \nin the fourth undercover video...	1
11	Cara Buckley	los angeles — to understand how andrew garf...	0
12		email ever wonder what’s on the mind of today’...	1
13	Kenneth Chang, Mike Isaac and Matt Richtel	a spectacular explosion of a spacex rocket on ...	0
14	Breitbart News	on the monday edition of breitbart news daily ...	0
15	Niraj Chokshi	the relationship may have faded long ago but t...	0
16	OK	next prev swipe leftright this honest trailer ...	1
17	Jared Taylor	why obama will win jared taylor american renai...	1
18	The Daily Sheeple	this is a daily news brief for all of the civi...	1
19	Alex Ansary	msm caught preparing hillary victory results p...	1

3. Removing stopwords

```
import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

stop = stopwords.words('english')



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

Output :



	written_by	news	label
0	Julius Hubris	0 add comment 250 million hollywood blockbuste...	1
1	PatriotRising	posted 10312016 1228 pm patriotrising 0 commen...	1
2	Niraj Chokshi and Vindu Goel	oath oof largely reaction monday news reported...	0
3	John Federico	election day enthusiasm receiving voted sticke...	1
4	Mac Slavo	argue young lady's speech bet donald trump bri...	1
5	Jonas E. Alexis	jonas e alexis november 1 2016 good thing mora...	1
6	Ariana	first ever hindu elected us house representati...	1
7	noreply@blogger.com (Der Postillon)	montag 14 november 2016 katastrophenschutz war...	1
8	Geoffrey Grider	desperate 'preserve legacy' barack obama relea...	1
9	WakingTimes	waking times establishment corporate liberal m...	1
10	Cassandra Fairbanks	change fourth undercover video guerilla journa...	1
11	Cara Buckley	los angeles — understand andrew garfield succe...	0
12		email ever wonder what's mind today's notable ...	1
13	Kenneth Chang, Mike Isaac and Matt Richtel	spectacular explosion spacex rocket thursday d...	0
14	Breitbart News	monday edition breitbart news daily broadcast ...	0
15	Niraj Chokshi	relationship may faded long ago intimate image...	0
16	OK	next prev swipe leftright honest trailer sherl...	1
17	Jared Taylor	obama win jared taylor american renaissance au...	1
18	The Daily Sheeple	daily news brief civil servants "doing job" re...	1
19	Alex Ansary	msm caught preparing hillary victory results p...	1

3.5 CONVERTING LABELS:-

The dataset has a Label column whose datatype is Numeric Category. So, no need to convert it.

3.6. VECTORIZATION

Vectorization translates words or phrases from vocabulary to a matching vector of actual numbers to identify word predictions, word similarities/semantics.

As a result, computer programs need that papers be converted into a numerical representation. A good example of this is 'Bag of Words'.

From the start, Scikit-Learn offers trustworthy vectorizer objects.

```
pipe = Pipeline([('vect', CountVectorizer()),

                 ('tfidf', TfidfTransformer()),

                 ('model', LogisticRegression())])
```

By utilizing CountVectorizer, we are able to compute the word counts for 'Tfidftransformer' before calculating the IDF values and the Tf-IDF scores. "Tfidfvectorizer" allows us to do all three processes at once, saving us time.

The code above will supply you with a matrix that represents your text, which you can then manipulate. Compressed Sparse Row format will be used for the sparse matrix.

Among the most popular vectorizers are the following:

Count Vectorizer: Calculates a token's weight by counting how many times a token appears in a document.
Hash Vectorizer: To save as much memory as possible, this one has been intended to be as little as feasible. It does this by using a hashing technique to encode them as numerical indices, rather than textual values. After vectorization, the names of the features cannot be recovered.

TF-IDF Vectorizer: In other words, the weight assigned to each token is based on both its frequency in a text and its frequency in the corpus as a whole. Here's more information on it.

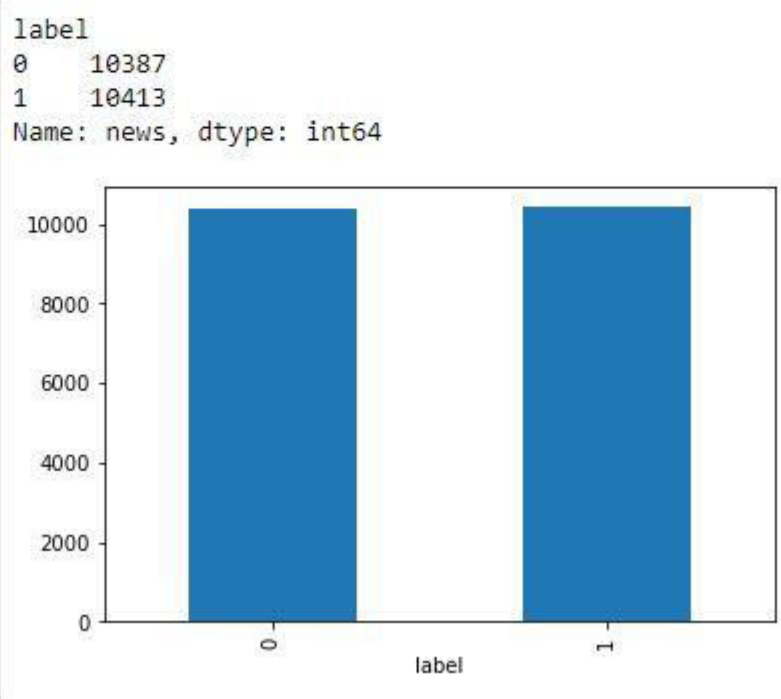
Visualization :

1. How many fake and real articles?

```
print(df.groupby(['label'])['news'].count())


df.groupby(['label'])['news'].count().plot(kind="bar")
```

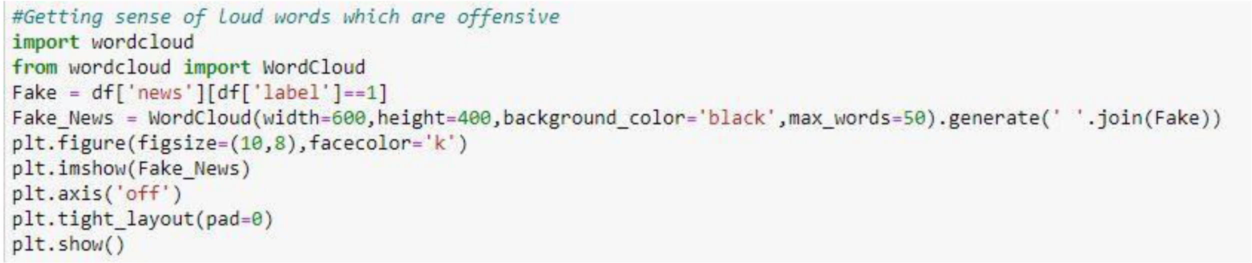
```
plt.show()
```

Output :



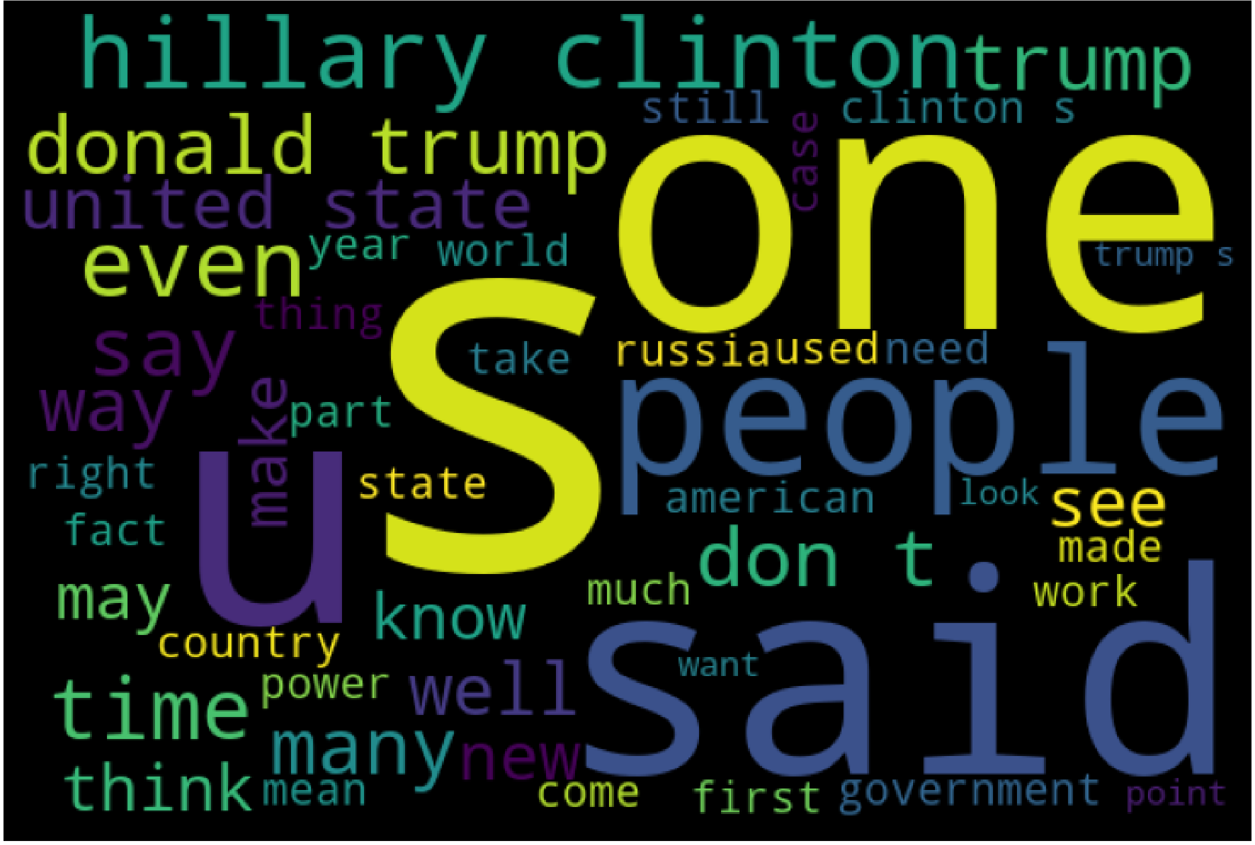
```
label
0    10387
1    10413
Name: news, dtype: int64
```

2. Getting sense of Fake words which are offensive.

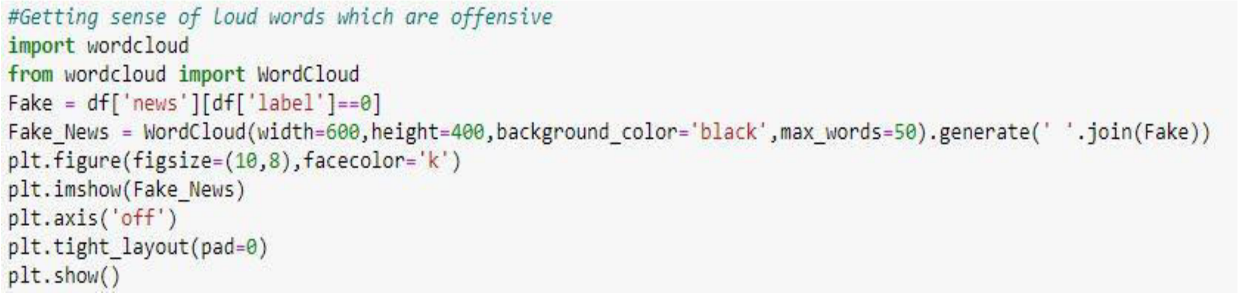


```
#Getting sense of loud words which are offensive
import wordcloud
from wordcloud import WordCloud
Fake = df['news'][df['label']==1]
Fake_News = WordCloud(width=600,height=400,background_color='black',max_words=50).generate(' '.join(Fake))
plt.figure(figsize=(10,8),facecolor='k')
plt.imshow(Fake_News)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

Output :

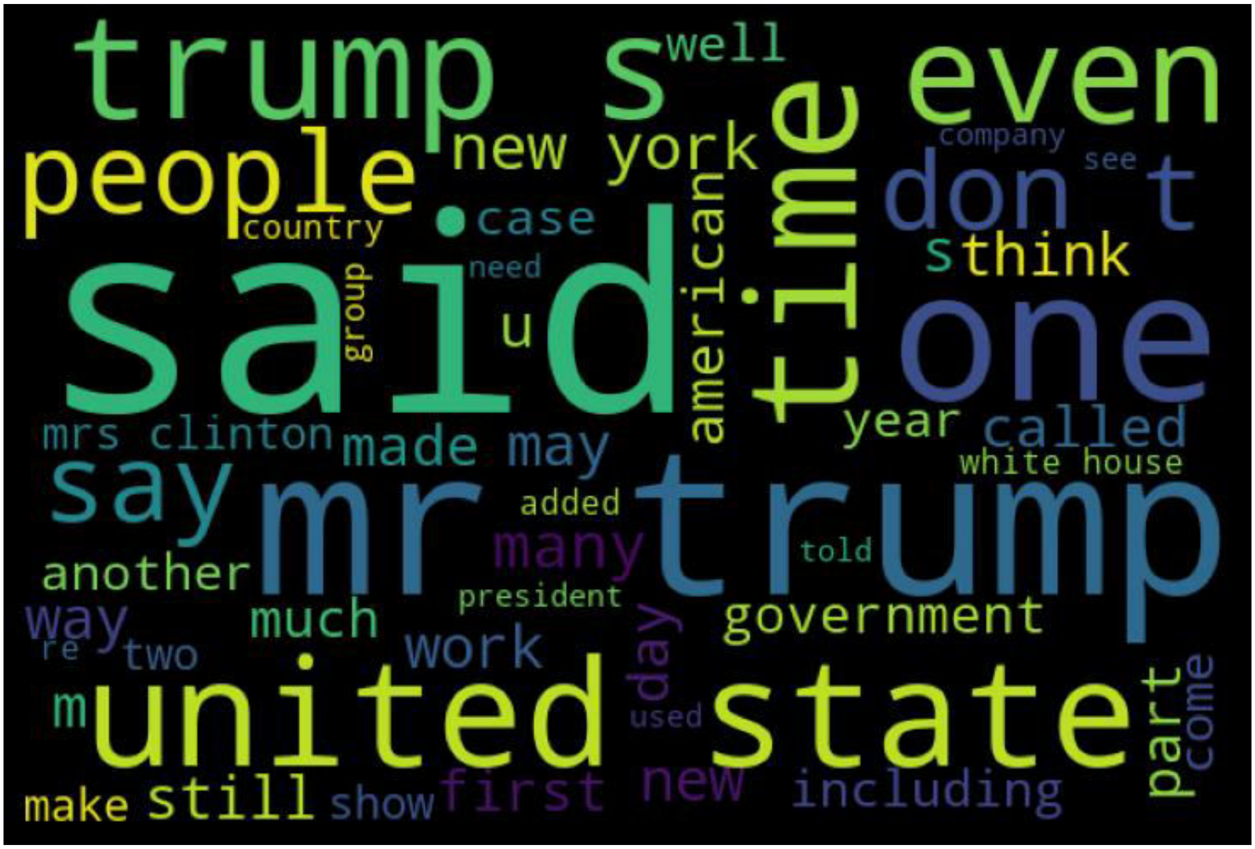


3. Getting sense of True words which are offensive.



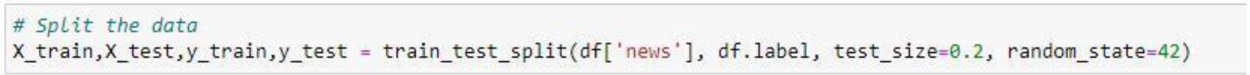
```
#Getting sense of loud words which are offensive
import wordcloud
from wordcloud import WordCloud
Fake = df['news'][df['label']==0]
Fake_News = WordCloud(width=600,height=400,background_color='black',max_words=50).generate(' '.join(Fake))
plt.figure(figsize=(10,8),facecolor='k')
plt.imshow(Fake_News)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

Output :



3.7. MODELING

Our data is then separated into test and training segments.



```
# Split the data
X_train,X_test,y_train,y_test = train_test_split(df['news'], df.label, test_size=0.2, random_state=42)
```

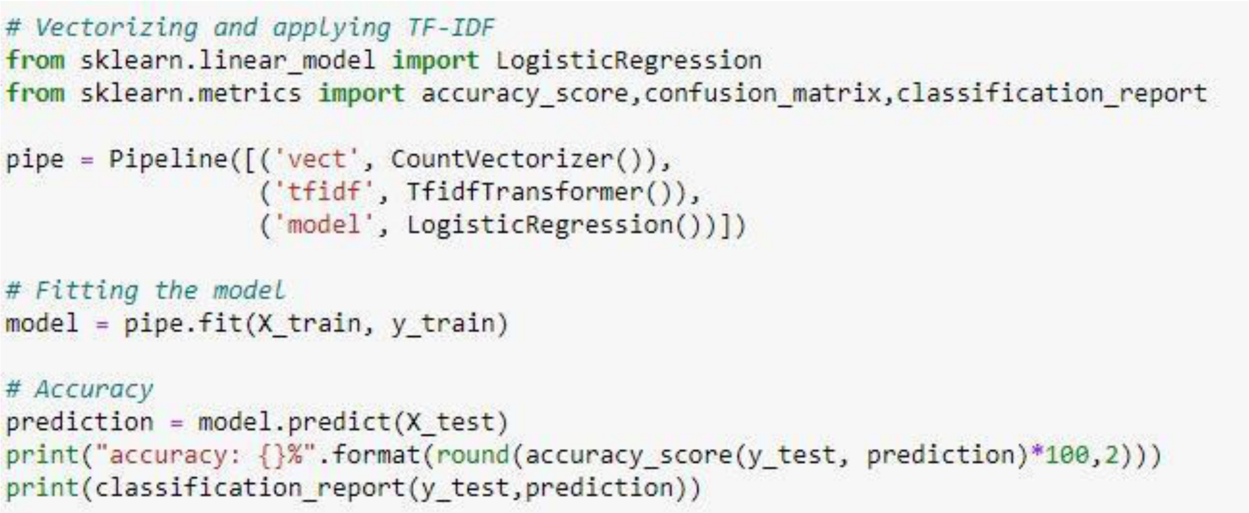
To analyze the data, I created four machine learning models (ML models).

Logistic Regression, Decision Tree, and Random Forest Classifier.

To do this, we used sklearn's accuracy score() and classification_report function to forecast on the test set.

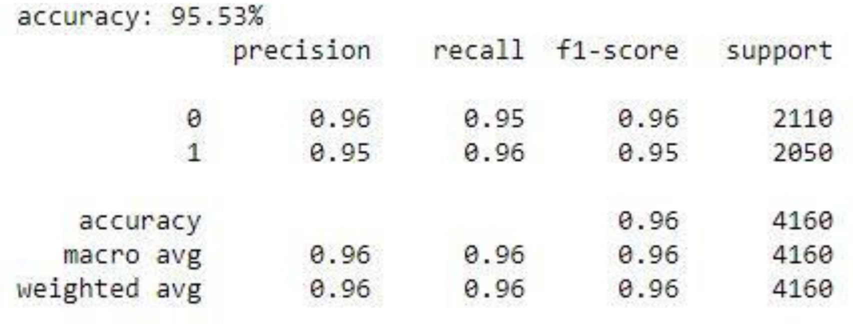
3.7.1. Logistic Regression

#LOGISTIC REGRESSION

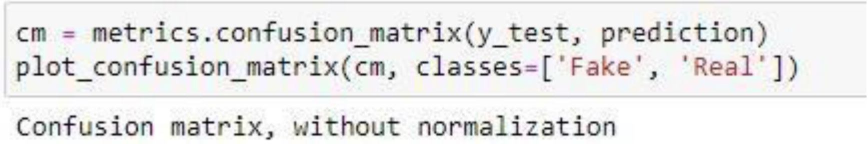


```
# Vectorizing and applying TF-IDF
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
print(classification_report(y_test,prediction))
```

Output :

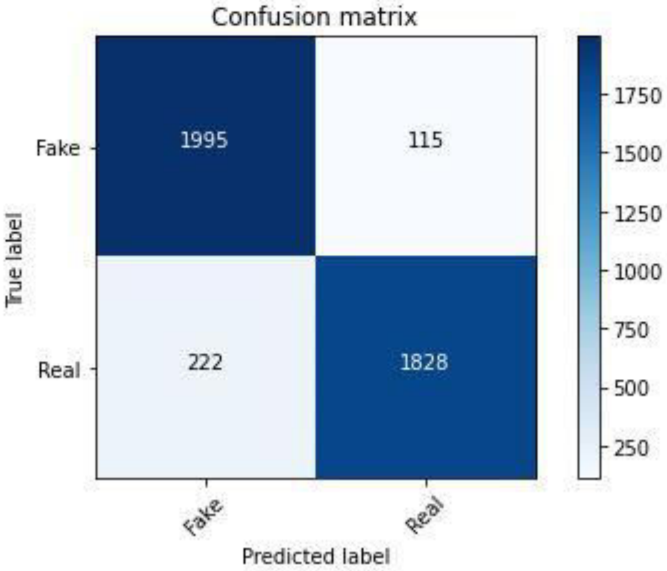


```
accuracy: 95.53%
              precision    recall  f1-score   support

           0       0.96      0.95      0.96      2110
           1       0.95      0.96      0.95      2050

    accuracy                           0.96      4160
   macro avg       0.96      0.96      0.96      4160
weighted avg       0.96      0.96      0.96      4160
```



```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

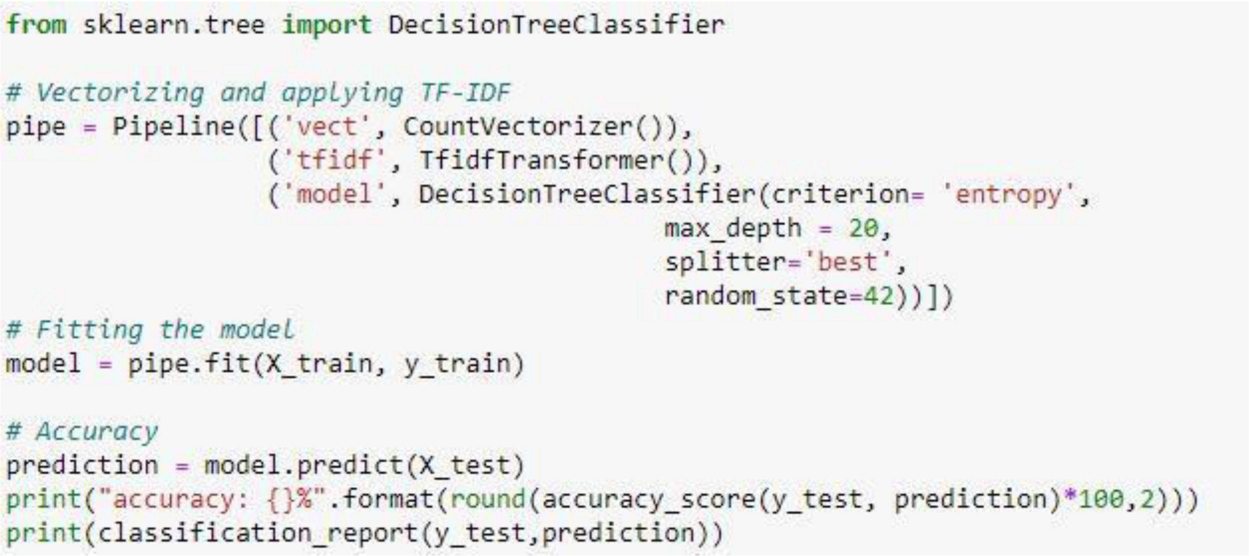
```
Confusion matrix, without normalization
```



Accuracy: 95.53%

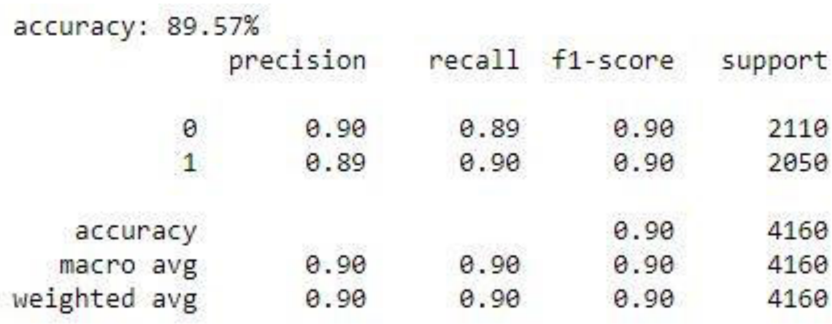
3.7.2. Decision Tree

# DECISION TREE

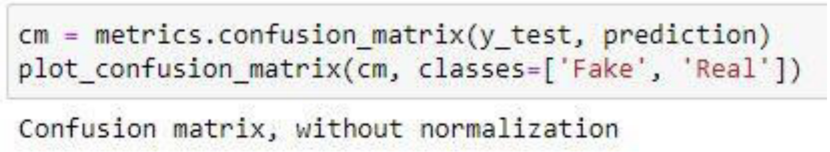


```
from sklearn.tree import DecisionTreeClassifier

# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
print(classification_report(y_test,prediction))
```

Output :

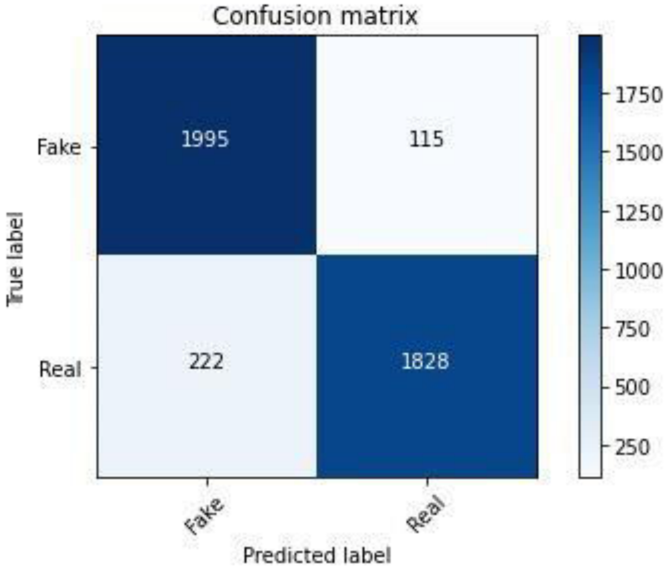


```
accuracy: 89.57%
              precision    recall  f1-score   support

           0       0.90      0.89      0.90      2110
           1       0.89      0.90      0.90      2050

    accuracy                           0.90      4160
   macro avg       0.90      0.90      0.90      4160
weighted avg       0.90      0.90      0.90      4160
```



```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

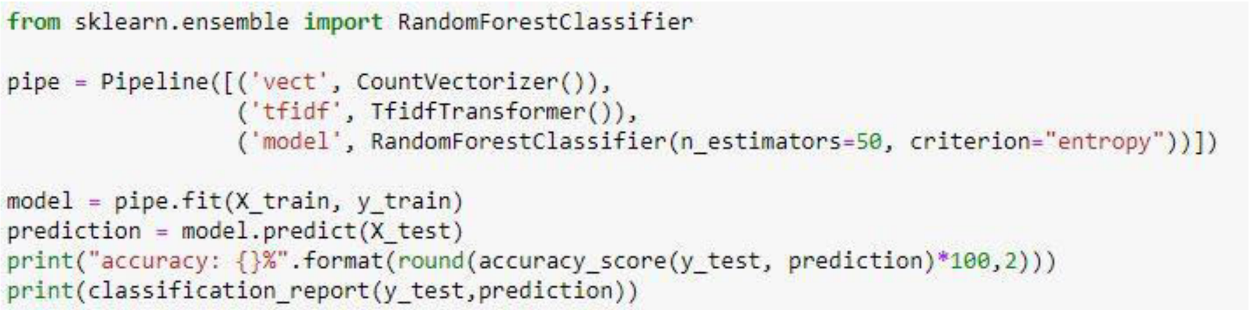
```
Confusion matrix, without normalization
```



Accuracy: 89.57 %

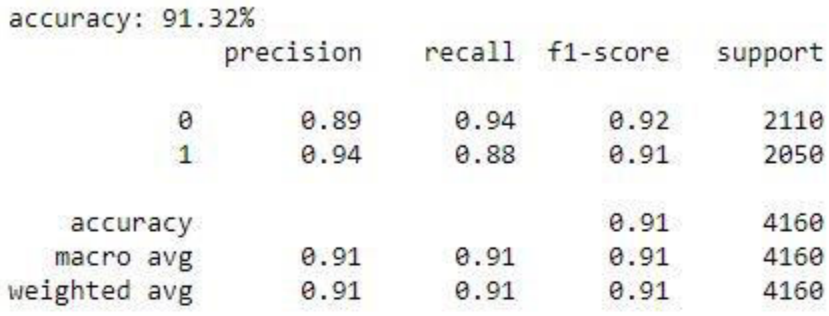
3.7.3. Random Forest Classifier

# Random Forest Classifier

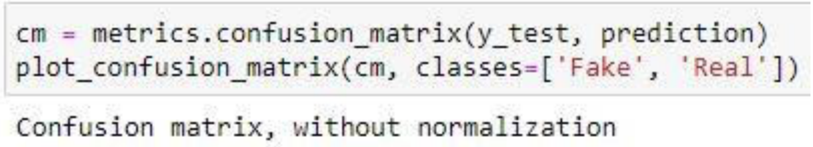


```
from sklearn.ensemble import RandomForestClassifier

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
print(classification_report(y_test,prediction))
```

Output :

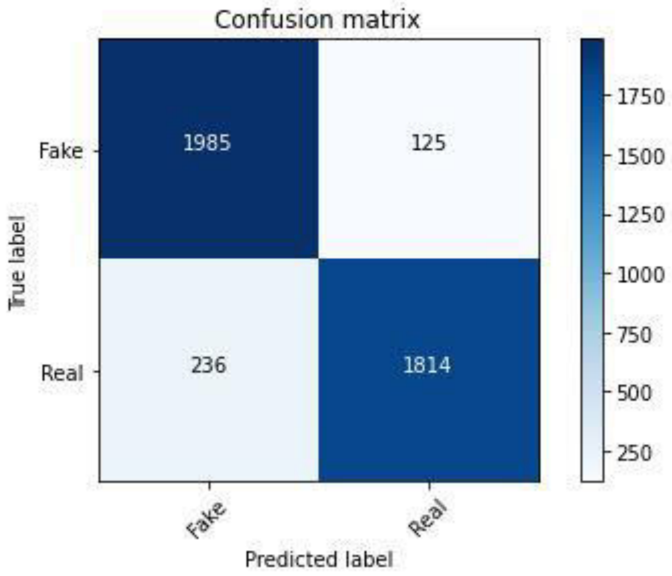


```
accuracy: 91.32%
              precision    recall  f1-score   support

           0       0.89      0.94      0.92      2110
           1       0.94      0.88      0.91      2050

    accuracy                           0.91      4160
   macro avg       0.91      0.91      0.91      4160
weighted avg       0.91      0.91      0.91      4160
```



```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```



Accuracy: 91.32%

4. CONCLUSION

95.53 percent accuracy was achieved through LOGISTIC REGRESSION.

The number of false negatives and genuine positives has also been recorded on a confusion matrix.

Style-based approaches for detecting fake news may be separated from content-based ones, often known as fact-checking. Ungrammaticality, poor grammar and punctuation, a restricted vocabulary and the use of abusive words are commonly cited as indicators of fake news.

This is a scenario where the machine's view must be backed up by explicit and verifiable evidence of the facts verified and the authority by which each fact was decided to be true, now more than ever.

A single data collection effort will not enough, considering the speed at which information is shared in today's linked world and the quantity of articles that are published on a regular basis.