```
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
```

```
Fake = pd.read_csv("D:\Capstone Project\Fake news detection\Fake.csv")
true = pd.read_csv("D:\Capstone Project\Fake news detection\True.csv")
```

```
C:\Users\Admin\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns
(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57
,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83
,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,1
07,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126
,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,1
46,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165
,166,167,168,169,170,171) have mixed types.Specify dtype option on import or
set low_memory=False. exec(code_obj, self.user_global_ns, self.user_ns)
```

```
Fake.shape
```

```
(23502, 172)
```

```
true.shape
```

```
(21417, 4)
```

```
Fake['label'] = 0
true['label'] = 1
```

```
Fake.shape
```

```
(23502, 173)
```

```
Fake.columns
```

```
Index(['title', 'text', 'subject', 'date', 'Unnamed: 4', 'Unnamed: 5',
'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9', ... 'Unnamed: 163',
'Unnamed: 164', 'Unnamed: 165', 'Unnamed: 166', 'Unnamed: 167', 'Unnamed:
168', 'Unnamed: 169', 'Unnamed: 170', 'Unnamed: 171', 'label'],
dtype='object', length=173)
```

```
Fake = Fake.loc[:, ~Fake.columns.str.contains('^Unnamed')]
```

```
Fake.columns
```
```
Index(['title', 'text', 'subject', 'date', 'label'], dtype='object')
```

```
Fake.drop(["date","title"],axis=1,inplace=True)
```

```
Fake.head()
```
```
                                                text     subject      label
0      Donald Trump just couldn t wish all Americans ... News   0
1      House Intelligence Committee Chairman Devin Nu... News   0
2      On Friday, it was revealed that former Milwauk... News   0
3      On Christmas day, Donald Trump announced that ... News   0
4      Pope Francis used his annual Christmas Day mes... News   0
```

```
true.shape
```
```
(21417, 5)
```

```
true.columns
```
```
Index(['title', 'text', 'subject', 'date', 'label'], dtype='object')
```

```
true.drop(["date","title"],axis=1,inplace=True)
```

```
true.head()
```
```
                                     text      subject      label
0      WASHINGTON (Reuters) - The head of a conservat... politicsNews 1
1      WASHINGTON (Reuters) - Transgender people will... politicsNews 1
2      WASHINGTON (Reuters) - The special counsel inv... politicsNews 1
3      WASHINGTON (Reuters) - Trump campaign adviser ... politicsNews 1
4      SEATTLE/WASHINGTON (Reuters) - President Donal... politicsNews 1
```

```
News = pd.concat([Fake, true]).reset_index(drop = True)
News.shape
```
```
(44919, 3)
```

```
from sklearn.utils import shuffle
```

```
News = shuffle(News)
News= News.reset_index(drop=True)
```

```
News.head()
```

	text	subject	label
0	BRUSSELS (Reuters) - The European Union s exec...	worldnews	1
1	JERUSALEM (Reuters) - Israel has ordered the d...	worldnews	1
2	A tweet caught my eye after exploring all of t...	politics	0
3	KABUL (Reuters) - The human rights group Amnes...	worldnews	1
4	ROME (Reuters) - Silvio Berlusconi has suggest...	worldnews	1

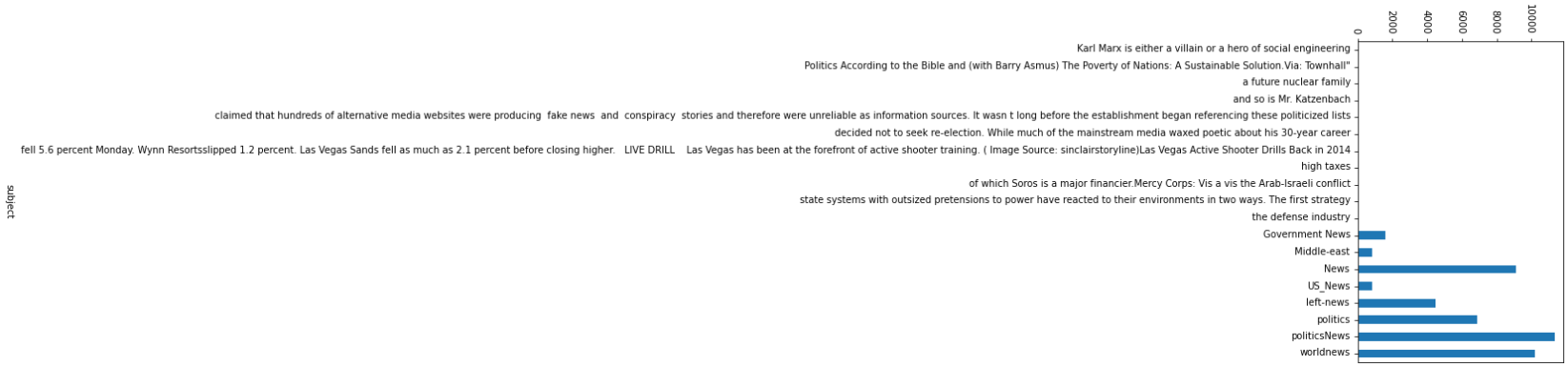
```
News['label'].value_counts()
```

```
0    23502
1    21417
Name: label, dtype: int64
```

```
import re
import string
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

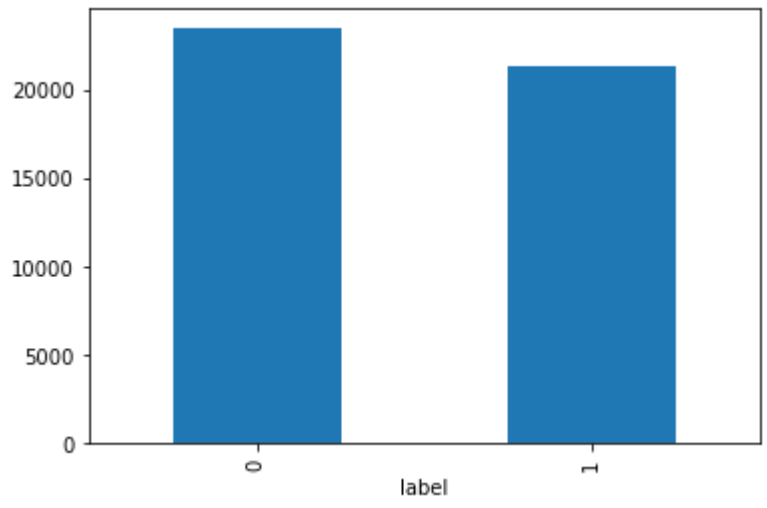
```
News["text"] = News["text"].apply(wordopt)
```

```
print(News.groupby(['subject'])['text'].count())
News.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()
```


0
2000
4000
6000
8000
10000
subject
Karl Marx is either a villain or a hero of social engineering
Politics According to the Bible and (with Barry Asmus) The Poverty of Nations: A Sustainable Solution.Via: Townhall"
a future nuclear family
and so is Mr. Katzenbach
claimed that hundreds of alternative media websites were producing fake news and conspiracy stories and therefore were unreliable as information sources. It wasn t long before the establishment began referencing these politicized lists
decided not to seek re-election. While much of the mainstream media waxed poetic about his 30-year career
fell 5.6 percent Monday. Wynn Resortsslipped 1.2 percent. Las Vegas Sands fell as much as 2.1 percent before closing higher. LIVE DRILL Las Vegas has been at the forefront of active shooter training. (Image Source: sinclairstoryline)Las Vegas Active Shooter Drills Back in 2014
high taxes
of which Soros is a major financier.Mercy Corps: Vis a vis the Arab-Israeli conflict
state systems with outsized pretensions to power have reacted to their environments in two ways. The first strategy
the defense industry
Government News
Middle-east
News
US_News
left-news
politics
politicsNews
worldnews

```
# How many fake and real articles?
print(News.groupby(['label'])['text'].count())
News.groupby(['label'])['text'].count().plot(kind="bar")
plt.show()
```

```
label
0    23502
1    21417
Name: text, dtype: int64
```



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

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

```
[nltk_data] Error loading stopwords: <urlopen error [SSL:
[nltk_data]     CERTIFICATE_VERIFY_FAILED] certificate verify failed:
[nltk_data]     self signed certificate in certificate chain
[nltk_data]     (_ssl.c:1129)>
```

```
News.head(10)
```

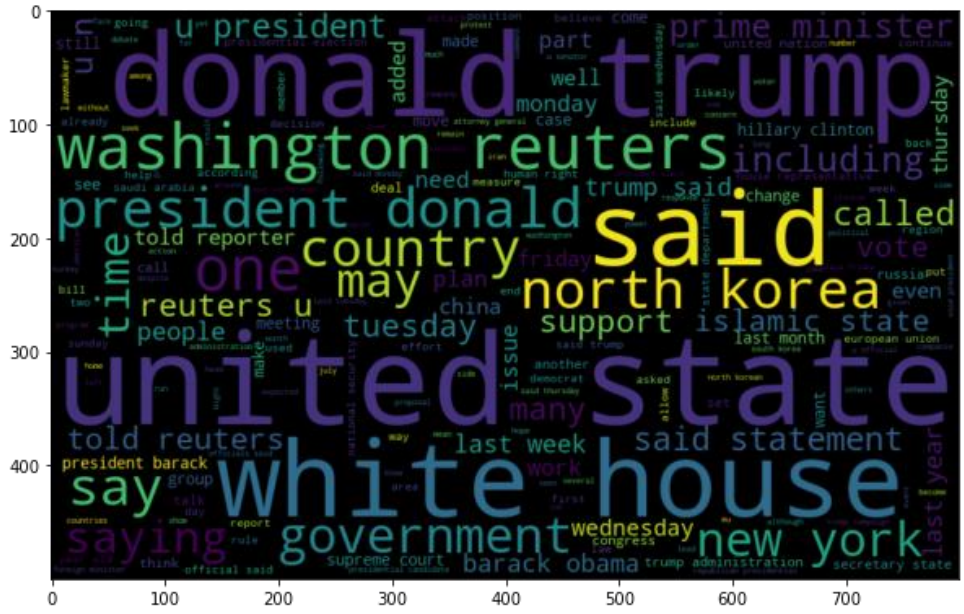
	text	subject	label
0	union executive offe...	brussels reuters european	worldnews 1
1	ordered deportation t...	jerusalem reuters israel	worldnews 1
2	hate violence poste...	tweet caught eye exploring	politics 0
3	group amnesty inter...	kabul reuters human rights	worldnews 1
4	berlusconi suggested carab...	rome reuters silvio	worldnews 1
5	network acr anoth...	tune alternate current radio	Middle-east 0
6	whether u representat...	detroit reuters question	politicsNews 1
7	admonished trump saying ...	many times press left	politics 0
8	reuters often fiercely ...	simi valley california	politicsNews 1
9	director james comey in...	doubt left whether fbi	News 0

```
#pip install WordCloud
```

```
from wordcloud import WordCloud
plt.figure(figsize=(10,7))
```

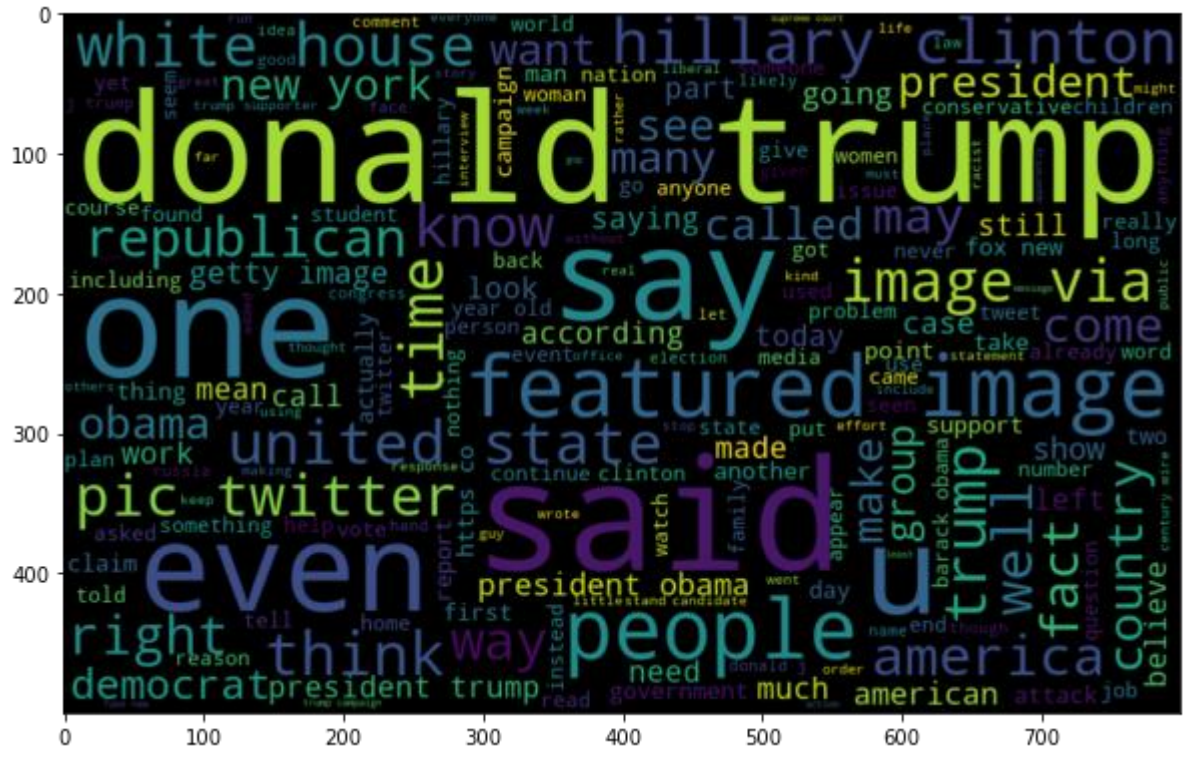
```
wc=WordCloud(width=800,height=500,max_font_size=110).generate("
".join(News[News.label == 1].text))
plt.imshow(wc,interpolation='bilinear')
```

```
<matplotlib.image.AxesImage at 0x23b36c12f70>
```



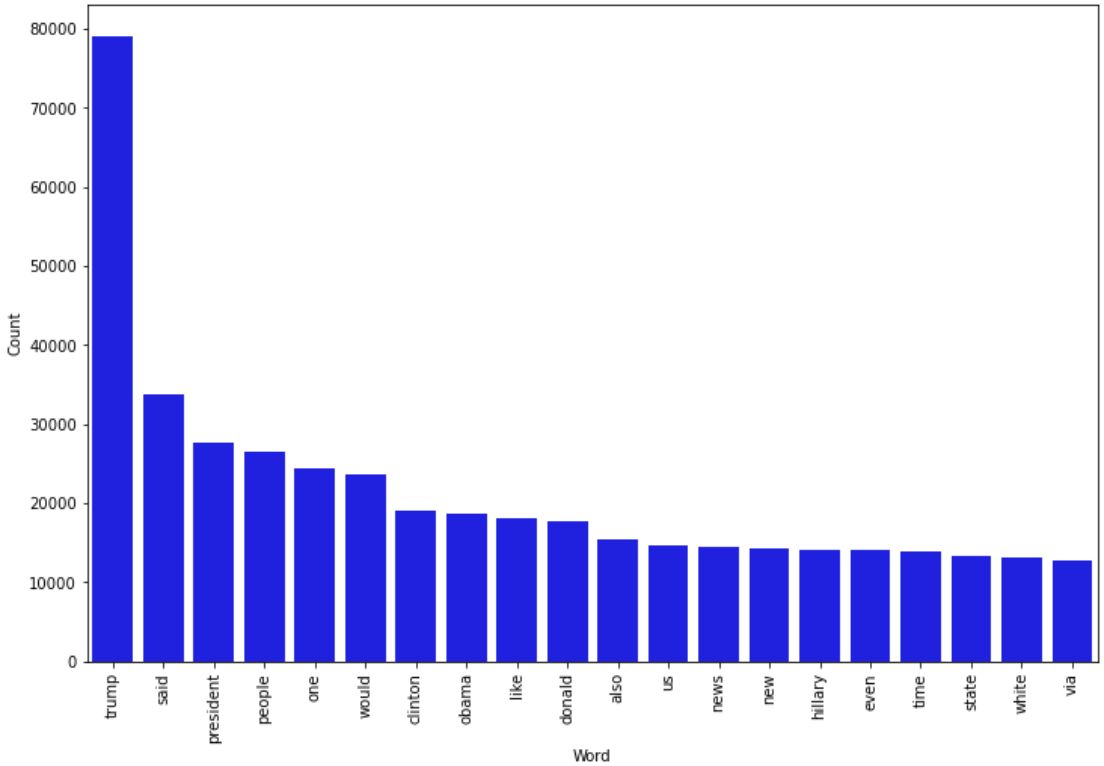

```
plt.figure(figsize=(10,7))
wc=WordCloud(width=800,height=500,max_font_size=110).generate("
".join(News[News.label == 0].text))
plt.imshow(wc,interpolation='bilinear')
```

```
<matplotlib.image.AxesImage at 0x23b36c39250>
```

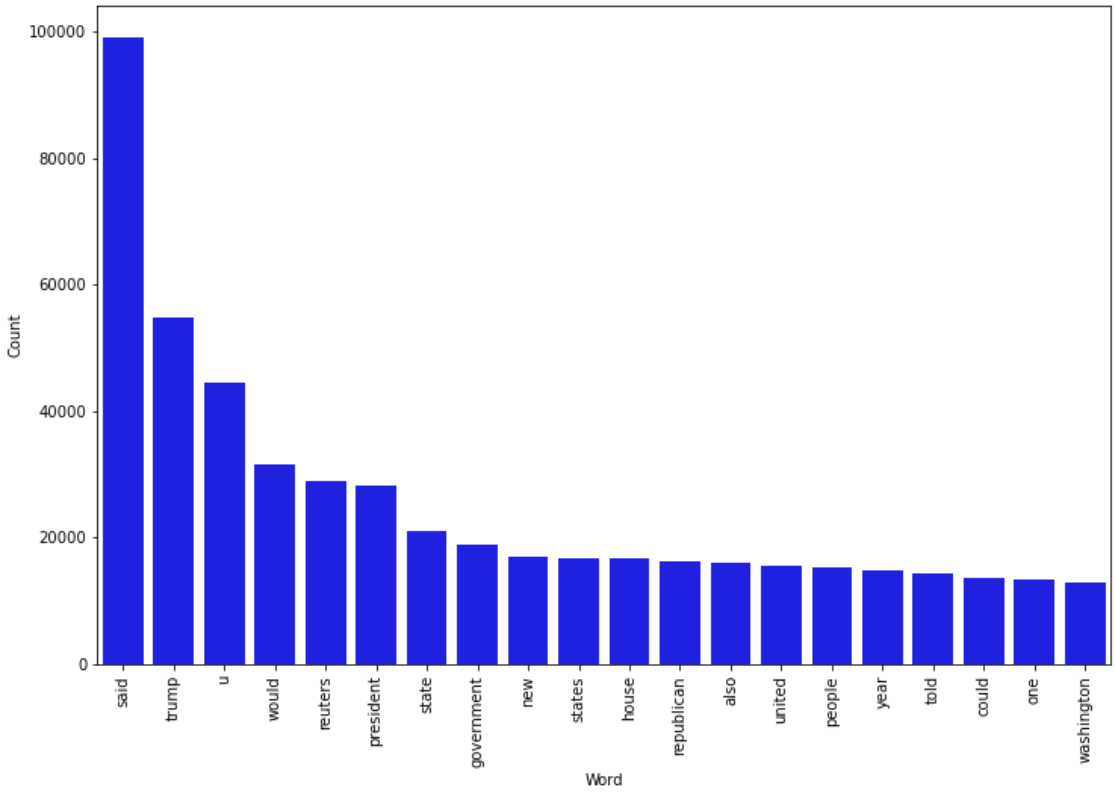


```
# Most frequent words counter (Code adapted from
https://www.kaggle.com/rodolfoluna/fake-news-detector)
from nltk import tokenize

token_space = tokenize.WhitespaceTokenizer()

def counter(text, column_text, quantity):
    all_words = ' '.join([text for text in text[column_text]])
    token_phrase = token_space.tokenize(all_words)
    frequency = nltk.FreqDist(token_phrase)
    df_frequency = pd.DataFrame({"Word": list(frequency.keys()),
                                 "Frequency": list(frequency.values())})
    df_frequency = df_frequency.nlargest(columns = "Frequency", n = quantity)
    plt.figure(figsize=(12,8))
    ax = sns.barplot(data = df_frequency, x = "Word", y = "Frequency", color =
'blue')
    ax.set(ylabel = "Count")
    plt.xticks(rotation='vertical')
    plt.show()
```

```
# Most frequent words in fake news
counter(News[News["label"] == 0], "text", 20)
```




```
# Most frequent words in real news
counter(News[News["label"] == 1], "text", 20)
```




```
x = News["text"]
y = News["label"]
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

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

```
vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

```
list1=[]
```

```
from sklearn.linear_model import LogisticRegression
```

```
LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

```
LogisticRegression()
```

```
pred_lr=LR.predict(xv_test)
```

```
LR.score(xv_test, y_test)
```

0.9878895814781834

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,pred_lr,squared=False)
```
0.11004734672774516

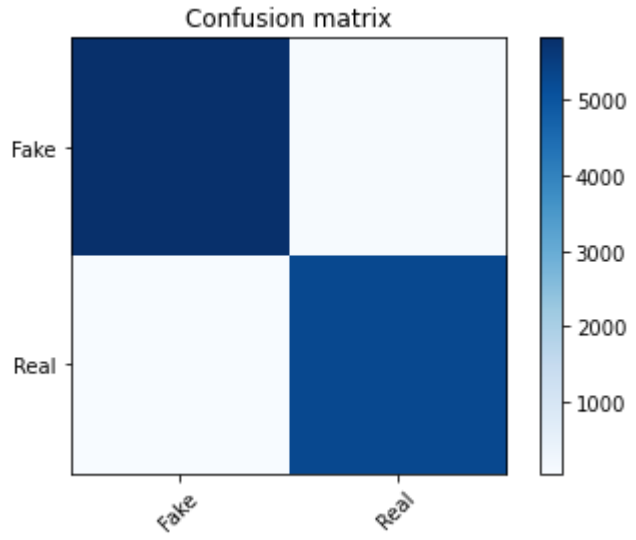
```
list1.append(LR.score(xv_test, y_test))
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred_lr))
```
precision   recall f1-score   support

0	0.99	0.99	0.99	5904
1	0.98	0.99	0.99	5326
accuracy			0.99	11230
macro avg	0.99	0.99	0.99	11230
weighted avg	0.99	0.99	0.99	11230

```
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)
```

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




```
from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit(xv_train,y_train)
```

```
MultinomialNB()
```

```
pred_nb=NB.predict(xv_test)
```

```
NB.score(xv_test, y_test)
```

```
0.9349065004452359
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,pred_nb,squared=False)
```

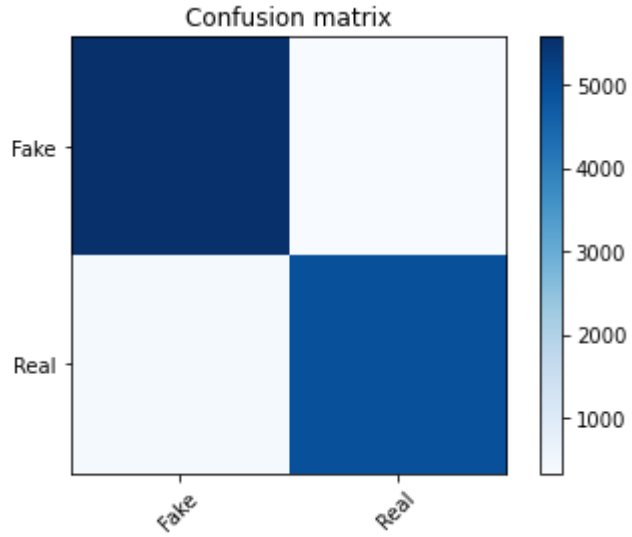
```
0.2551342774986615
```

```
print(classification_report(y_test, pred_nb))
```

```
precision    recall  f1-score   support

           0       0.93      0.94      0.94      5904
           1       0.94      0.92      0.93      5326

    accuracy                           0.93     11230
   macro avg       0.94      0.93      0.93     11230
weighted avg       0.93      0.93      0.93     11230
```

```
list1.append(NB.score(xv_test, y_test))
```

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




```
from sklearn.tree import DecisionTreeClassifier
```

```
DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
```

```
DecisionTreeClassifier()
```

```
pred_dt = DT.predict(xv_test)
```

```
DT.score(xv_test, y_test)
```

```
0.995280498664292
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,pred_dt,squared=False)
```

```
0.06869862688371527
```

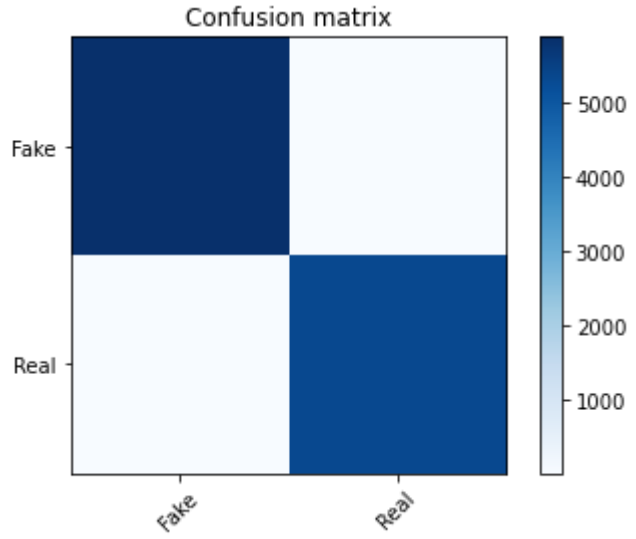
```
list1.append(DT.score(xv_test, y_test))
```

```
print(classification_report(y_test, pred_dt))
```

```
precision    recall  f1-score   support

           0       1.00      0.99      1.00      5904
           1       0.99      1.00      1.00      5326
```

```
    accuracy                           1.00     11230
   macro avg       1.00      1.00      1.00     11230
weighted avg       1.00      1.00      1.00     11230
```

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




```
from sklearn import svm
```

```
SVM= svm.SVC(kernel='linear')
SVM.fit(xv_train, y_train)
```
```
SVC(kernel='linear')
```

```
pred_svm = SVM.predict(xv_test)
```

```
SVM.score(xv_test, y_test)
```
```
0.9951914514692787
```

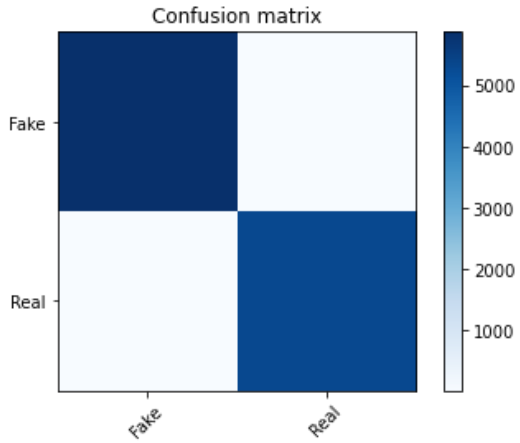
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,pred_svm,squared=False)
```
```
0.06934369856534393
```

```
list1.append(SVM.score(xv_test, y_test))
```

```
print(classification_report(y_test, pred_svm))
```

```
precision    recall  f1-score   support

           0       1.00      0.99      1.00      5904
           1       0.99      1.00      0.99      5326

    accuracy                           1.00     11230
   macro avg       1.00      1.00      1.00     11230
weighted avg       1.00      1.00      1.00     11230
```

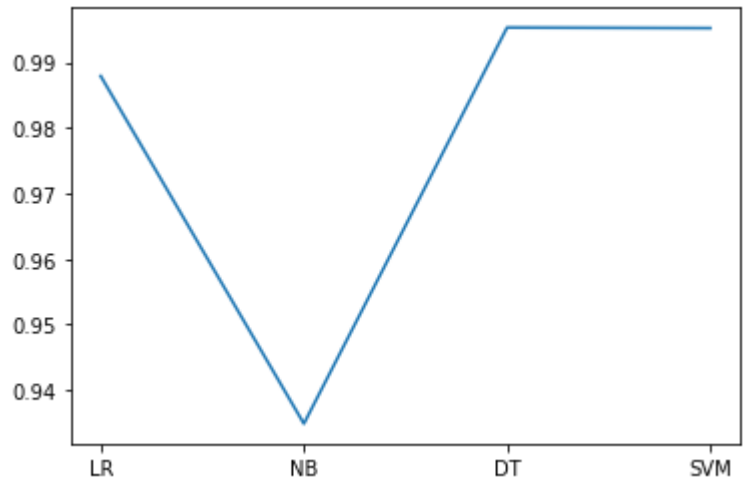
```
cm = metrics.confusion_matrix(y_test,pred_svm)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```




```
print(list1)
list2=["LR","NB","DT","SVM"]
plt.plot(list2,list1)
```

```
[0.9878895814781834, 0.9349065004452359, 0.995280498664292,
0.9951914514692787]
```

```
[<matplotlib.lines.Line2D at 0x23b145fceb0>]
```




```
def output_lable(n):
    if n == 0:
        return "FAKE NEWS"
    else:
        return "TRUE NEWS"

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_dt = DT.predict(new_xv_test)

    return print( "\nDT Prediction: {} ".format(output_lable(pred_dt[0])))
```

```
news = str(input())
manual_testing(news)
```

GENEVA (Reuters) - North Korea and the United States accused each other on Tuesday of posing a nuclear threat, with Pyongyang s envoy declaring it would never put its atomic arsenal up for negotiation. The debate at the United Nations began when the U.S. envoy said President Donald Trump s top priority was to protect the United States and its allies against the growing threat from North Korea. To do so, he said, the country was ready to use the full range of capabilities at our disposal . U.S. Ambassador Robert Wood told the Conference on Disarmament that the path to dialogue still remains an option for Pyongyang, but that Washington was undeterred in defending against the threat North Korea poses . Fears have grown over North Korea s development of missiles and nuclear weapons since Pyongyang test-launched intercontinental ballistic missiles (ICBMs) in July. Those fears worsened after Trump warned that North Korea would face fire and fury if it threatened the United States. His remarks led North Korea to say it was considering plans to fire missiles towards the U.S. Pacific territory of Guam. Trump responded by tweeting that the U.S. military was locked and loaded, should North Korea act

unwisely . A few days later, North Korean media reported the country s leader, Kim Jong Un, had delayed any decision on whether to fire missiles towards Guam while he waited to see what the United States would do. Experts warned Pyongyang could still go ahead with the missile launches. North Korea s ballistic missile and nuclear weapons programs pose grave threats to the entire world, Wood told the Geneva forum. Its recent ICBM tests are another example of the dangerous reckless behavior of the North that is destabilizing the region and beyond. North Korea had openly stated that its missiles are intended to strike cities in the United States and its allies South Korea and Japan, he said. My president s top priority remains protecting the homeland, U.S. territories and our allies against North Korean aggression. We remain prepared to use the full range of capabilities at our disposal against the growing threat from North Korea. North Korea diplomat Ju Yong Chol said that measures taken by his country to strengthen its nuclear deterrent and develop inter-continental rockets were justifiable and a legitimate option . As long as the U.S. hostile policy and nuclear threat remains unchallenged, the DPRK will never place its self-defensive nuclear deterrence on the negotiating table or step back an inch from the path it took to bolster the national nuclear force, Ju said. In a subsequent speech, Ju said: The United States should clearly understand that military threats and pressure are only serving as a momentum that pushes the DPRK further into developing fully strengthened nuclear deterrence. Regarding joint U.S.-South Korean military exercises that began on Monday, he said: The ongoing military adventure would certainly add gasoline to the fire, driving the current tense situation to further deterioration. China s disarmament ambassador, Fu Cong, called for support for its proposal to defuse the crisis affecting its Pyongyang ally. China has called for dual suspension , that is of North Korea s nuclear activities and joint military exercises between the Republic of Korea and United States. This seeks to denuclearize the peninsula and promote a security mechanism. Wood rejected Beijing s freeze for freeze plan. This proposal unfortunately creates a false equivalency between states that are engaging in legitimate exercises of self-defense who have done so for many years with a regime that has basically violated countless Security Council resolutions with regard to its proscribed nuclear and ballistic missile programs, he told the gathering. That is a false equivalency that we cannot accept and will not accept, he said. Fu retorted: I just want to say that we re not creating equivalency between anything. We are just actually making the proposal to facilitate a dialogue and to reduce the tension. We need a starting point to really launch the dialogue.

DT Prediction: TRUE NEWS