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 (4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171) 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)

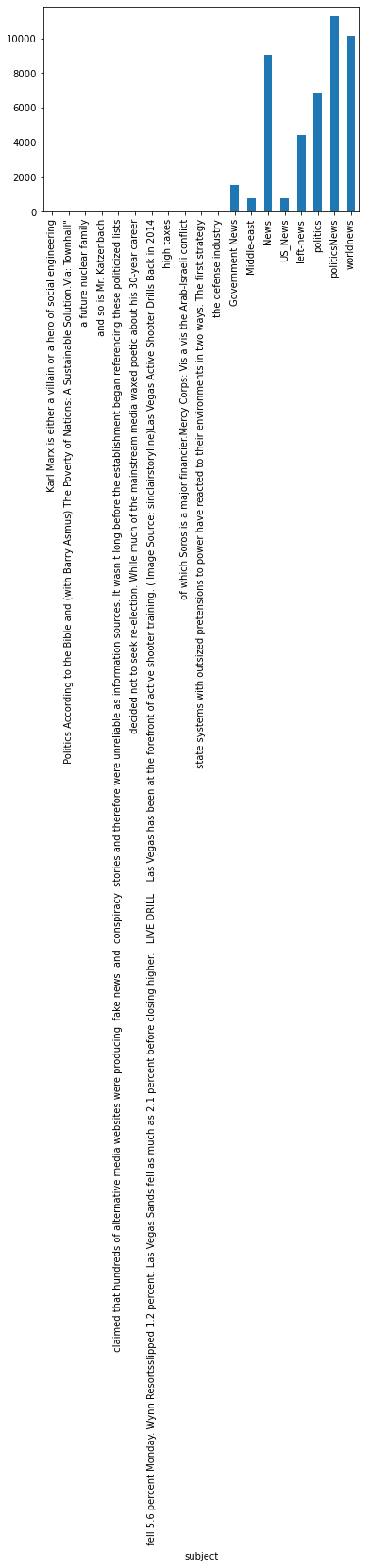
    return text

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

print(News.groupby(['subject'])['text'].count())

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

plt.show()



# How many fake and real articles?

print(News.groupby(['label'])['text'].count())

News.groupby(['label'])['text'].count().plot(kind="bar")

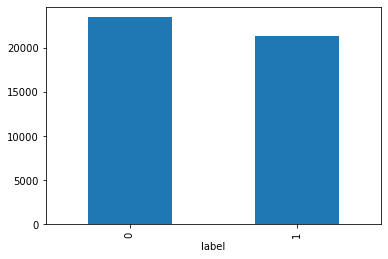
plt.show()

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 brussels reuters european union executive offe... worldnews 1

1 jerusalem reuters israel ordered deportation t... worldnews 1

2 tweet caught eye exploring hate violence poste... politics 0

3 kabul reuters human rights group amnesty inter... worldnews 1

4 rome reuters silvio berlusconi suggested carab... worldnews 1

5 tune alternate current radio network acr anoth... Middle-east 0

6 detroit reuters question whether u representat... politicsNews 1

7 many times press left admonished trump saying ... politics 0

8 simi valley california reuters often fiercely ... politicsNews 1

9 doubt left whether fbi director james comey in... News 0

#pip install WordCloud

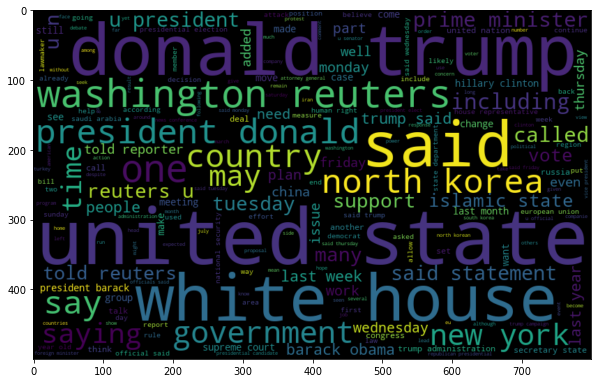
from wordcloud import WordCloud

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')

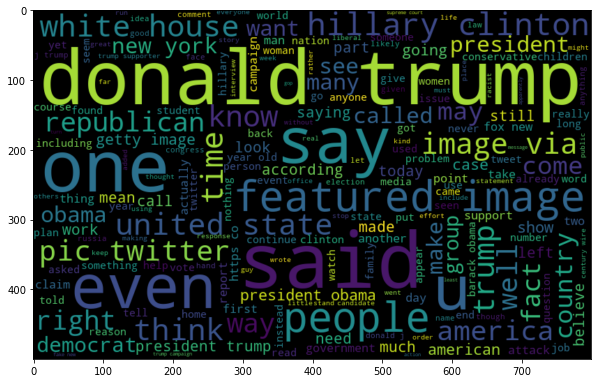
<matplotlib.image.AxesImage at 0x23b36c12f70>



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')

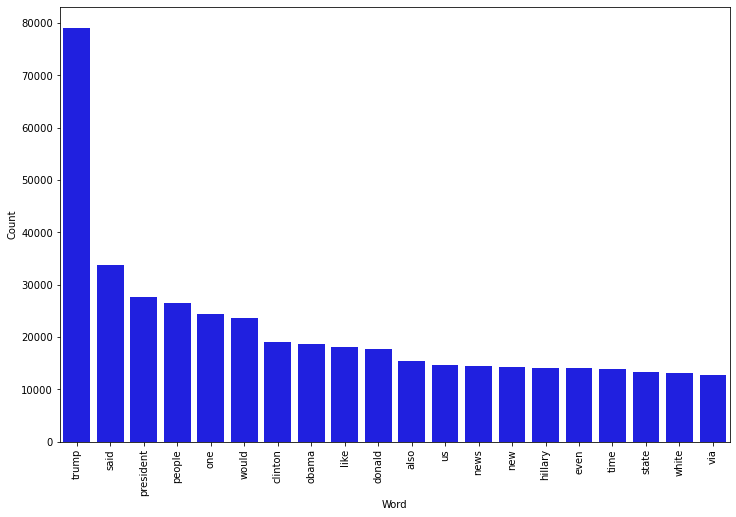
    ax.set(ylabel = "Count")

    plt.xticks(rotation='vertical')

    plt.show()

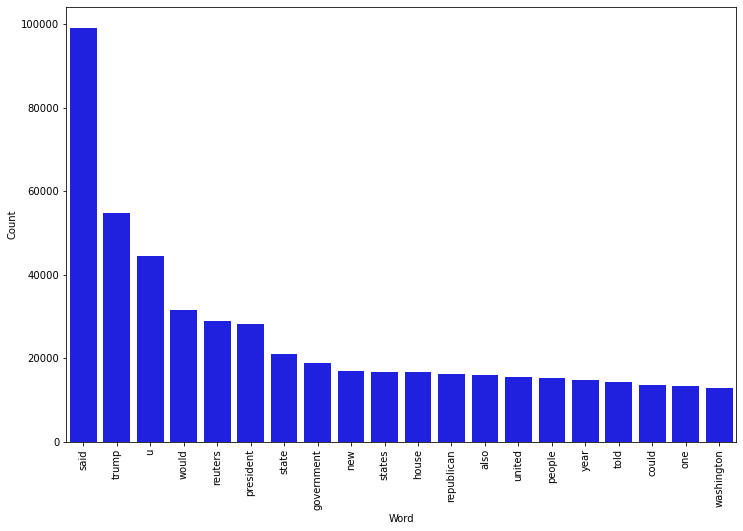
# Most frequent words in fake news

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'])



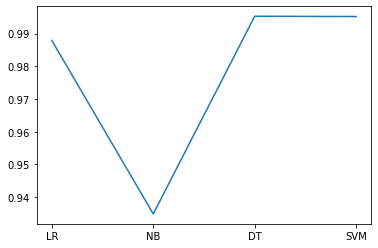
print(list1)

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