```
#Install the Kaggle library

! pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: kaggle in /usr/local/lib/python3.8/dist-packages (1.5.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from kaggle) (4.64.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages (from kaggle) (2022.12.7)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.8/dist-packages (from kaggle) (1.26.14)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.8/dist-packages (from kaggle) (7.0.0)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from kaggle) (2.25.1)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.8/dist-packages (from kaggle) (1.15.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.8/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.8/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->kaggle) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->kaggle) (2.10)
```

```
#Make a directory named “.kaggle”

! mkdir ~/.kaggle
```

```
mkdir: cannot create directory ‘/root/.kaggle’: File exists
```

```
#Copy the “kaggle.json” into this new directory

! cp kaggle.json ~/.kaggle/
```

```
cp: cannot stat 'kaggle.json': No such file or directory
```

```
#Allocate the required permission for this file.

! chmod 600 ~/.kaggle/kaggle.json
```

```
#downloading dataset
! kaggle datasets download -d ashwithanoble/phishing-sites-url
```

```
phishing-sites-url.zip: Skipping, found more recently modified local copy (use --force to force download)
```

```
! pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: kaggle in /usr/local/lib/python3.8/dist-packages (1.5.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from kaggle) (4.64.1)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.8/dist-packages (from kaggle) (1.26.14)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from kaggle) (2.25.1)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.8/dist-packages (from kaggle) (7.0.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages (from kaggle) (2022.12.7)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.8/dist-packages (from kaggle) (1.15.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.8/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.8/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->kaggle) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->kaggle) (4.0.0)
```

```
! mkdir ~/.kaggle
```

```
mkdir: cannot create directory ‘/root/.kaggle’: File exists
```

```
!cp /content/drive/MyDrive/kaggle.json  ~/.kaggle/kaggle.json
```

```
#downloading dataset
! kaggle datasets download -d ashwithanoble/phishing-sites-url
```

```
phishing-sites-url.zip: Skipping, found more recently modified local copy (use --force to force download)
```

```
#unzipping the file
!unzip phishing-sites-url.zip
```

```
Archive:  phishing-sites-url.zip
replace urls_for_phishing.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename: no
```

```
! pip install selenium
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: selenium in /usr/local/lib/python3.8/dist-packages (4.8.0)
Requirement already satisfied: urllib3[socks]~=1.26 in /usr/local/lib/python3.8/dist-packages (from selenium) (1.26.14)
Requirement already satisfied: trio-websocket~=0.9 in /usr/local/lib/python3.8/dist-packages (from selenium) (0.9.2)
Requirement already satisfied: trio~=0.17 in /usr/local/lib/python3.8/dist-packages (from selenium) (0.22.0)
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.8/dist-packages (from selenium) (2022.12.7)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (22.2.0)
Requirement already satisfied: outcome in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (1.2.0)
```

```
Requirement already satisfied: async-generator>=1.9 in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (1.10)
Requirement already satisfied: idna in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (2.10)
Requirement already satisfied: sniffio in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (1.1.0)
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.8/dist-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: wsproto>=0.14 in /usr/local/lib/python3.8/dist-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.8/dist-packages (from urllib3[socks]~=1.26->selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.9.0 in /usr/local/lib/python3.8/dist-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0
```

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

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from nltk.tokenize import RegexpTokenizer
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import make_pipeline
from PIL import Image
from bs4 import BeautifulSoup
from selenium import webdriver
import networkx as nx
import pickle
import warnings
warnings.filterwarnings('ignore')
```

```
data=pd.read_csv('urls_for_phishing.csv')
```

```
data.head(10)
```

	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad
3	mail.printakid.com/www.online.americanexpress....	bad
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad
5	smilesvoegol.servebbs.org/voegol.php	bad
6	premierpaymentprocessing.com/includes/boleto-2...	bad
7	myxxxcollection.com/v1/js/jih321/bpd.com.do/do...	bad
8	super1000.info/docs	bad
9	horizonsgallery.com/js/bin/ssl1/_id/www.paypal...	bad

```
data.tail(10)
```

	URL	Label
507102	stefanocardone.com/wp-includes/SimplePie/HTTP/...	bad
507103	shapingsoftware.com/2009/02/09/architectural-s...	bad
507104	free.ulohapp.info/?br_fl=2872&tuif=5539&am...	bad
507105	free.ulohapp.info/?oq=CEh3h_PskJLFZaQWwjEKBegU...	bad
507106	mol.com-ho.me/cv_itworx.doc	bad
507107	23.227.196.215/	bad
507108	apple-checker.org/	bad
507109	apple-iclods.org/	bad
507110	apple-uptoday.org/	bad
507111	apple-search.info	bad

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 507112 entries, 0 to 507111
```

```
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   URL     507112 non-null  object
 1   Label   507112 non-null  object
dtypes: object(2)
memory usage: 7.7+ MB
```

```
data.shape
```

```
(507112, 2)
```

```
data.isnull().sum() #to find null value
```

```
URL      0
Label    0
dtype: int64
```

```
data.duplicated()
```

```
0         False
1         False
2         False
3         False
4         False
          ...
507107    False
507108    False
507109    False
507110    False
507111    False
Length: 507112, dtype: bool
```

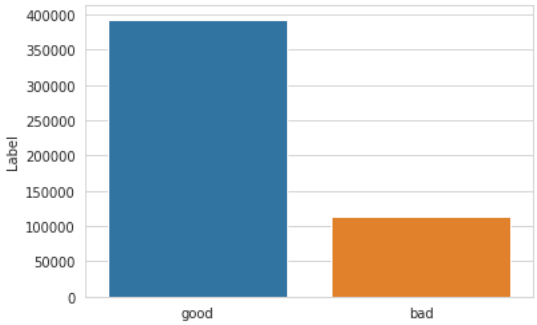
```
data.duplicated().sum()
```

```
0
```

```
label_counts = pd.DataFrame(data.Label.value_counts())
```

```
sns.set_style('whitegrid')
sns.barplot(label_counts.index,label_counts.Label)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6cbab4d940>
```




```
tokenizer = RegexpTokenizer(r'[A-Za-z]+')
```

```
# this will be pull letter which matches to expression
tokenizer.tokenize(data.URL[0]) # using first row
```

```
['nobell',
 'it',
 'ffb',
 'd',
 'dca',
 'cce',
 'f',
 'login',
 'SkyPe',
 'com',
 'en',
 'cgi',
 'bin',
 'verification',
 'login',
 'ffb',
 'd',
 'dca',
 'cce',
 'f',
 'index',
 'php',
```

```
 'cmd',
 'profile',
 'ach',
 'outdated',
 'page',
 'tmpl',
 'p',
 'gen',
 'failed',
 'to',
 'load',
 'nav',
 'login',
 'access']
```

```
print('Getting words tokenized ...')
t0= time.perf_counter()
data['text_tokenized'] = data.URL.map(lambda t: tokenizer.tokenize(t)) # doing with all rows
t1 = time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Getting words tokenized ...
Time taken 2.7827199940002174 sec
```

```
data.sample(10)
```

	URL	Label	text_tokenized
39569	audit-internet.com/templates/beez/3409555044/7...	bad	[audit, internet, com, templates, beez]
193145	iagenweb.org/iowa/bmd/	good	[iagenweb, org, iowa, bmd]
458368	whosdatedwho.com/tpx_616469/black-cat-white-cat/	good	[whosdatedwho, com, tpx, black, cat, white, cat]
292829	canadianmysteries.ca/sites/gagnon/echos/classi...	good	[canadianmysteries, ca, sites, gagnon, echos, ...
167168	en.wikipedia.org/wiki/Category:South_African_p...	good	[en, wikipedia, org, wiki, Category, South, Af...
246929	uk.linkedin.com/in/edstileman	good	[uk, linkedin, com, in, edstileman]
92788	www.gamerz.net/pbmserv/	good	[www, gamerz, net, pbmserv]
	santabanta.com/wallpapers/category.asp?		[santabanta, com, wallpapers

```
stemmer = SnowballStemmer("english")
```

```
print('Getting words stemmed ...')
t0= time.perf_counter()
data['text_stemmed'] = data['text_tokenized'].map(lambda l: [stemmer.stem(word) for word in l])
t1= time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Getting words stemmed ...
Time taken 64.49509245900026 sec
```

```
data.sample(10)
```

	URL	Label	text_tokenized	
451079	usgennet.org/usa/mi/county/tuscola/waymar/bigr...	good	[usgennet, org, usa, mi, county, tuscola, waym...	[usge mi,
186514	genforum.com/mcneil/page3.html	good	[genforum, com, mcneil, page, html]	[g mcn
384836	modernhaiku.org/friends.html	good	[modernhaiku, org, friends, html]	[mo
463996	yelp.com/biz/walters-hot-dog-stand-mamaroneck	good	[yelp, com, biz, walters, hot, dog, stand, mam...	[yelp, c h
160748	dramascenemagazine.com/	good	[dramascenemagazine, com]	[dramas
233140	shop.surreycompany.com/6-Person-Surreys.c4.htm	good	[shop, surreycompany, com, Person, Surreys,	[shop, s com,

```
print('Getting joiningwords ...')
t0= time.perf_counter()
data['text_sent'] = data['text_stemmed'].map(lambda l: ' '.join(l))
t1= time.perf_counter() - t0
print('Time taken',t1 ,'sec')
```

```
Getting joiningwords ...
Time taken 0.30578428999979224 sec
```

```
data.sample(5)
```

	URL	Label	text_tokenized	text_stemmed	text_sent
147548	bruce-springsteen-blog.blogspot.com/	good	[bruce, springsteen, blog, blogspot, com]	[bruce, springsteen, blog, blogspot, com]	bruce springsteen blog blogspot com
254325	wn.com/CJNT-TV	good	[wn, com, CJNT, TV]	[wn, com, cjnt, tv]	wn com cjnt tv

```
bad_sites = data[data.Label == 'bad']
good_sites =data[data.Label == 'good']
```

```
bad_sites.head(10)
```

	URL	Label	text_tokenized	
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad	[nobell, it, ffb, d, dca, cce, f, login, SkyPe...	[nobel
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad	[www, dghjdgf, com, paypal, co, uk, cycgi, bin...	[pa
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad	[serviciosbys, com, paypal, cgi, bin, get, int...	[servic
3	mail.printakid.com/www.online.americanexpress....	bad	[mail, printakid, com, www, online, americanex...	[n
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad	[thewhiskeydregs, com, wp, content, themes, wi...	[the
5	smilesvoegol.servebbs.org/voegol.php	bad	[smilesvoegol, servebbs,	[smil

```
from os import path
from wordcloud import WordCloud, STOPWORDS
```

```
def google_authenticate():
  from google.colab import auth
  auth.authenticate_user()

  from googleapiclient.discovery import build
  drive_service = build('drive', 'v3')
  return drive_service
```

```
drive_service = google_authenticate()
#authenticate from Ashwitha Noble
```

```
def read_file(file_id):
  """
  Download file from Google Drive
  Argument: file_id
  Returns: downloaded file
  """

  file_id = file_id

  import io
  from googleapiclient.http import MediaIoBaseDownload

  request = drive_service.files().get_media(fileId=file_id)
  downloaded = io.BytesIO()
  downloader = MediaIoBaseDownload(downloaded, request)
  done = False
  while done is False:
    # _ is a placeholder for a progress object that we ignore.
    # (Our file is small, so we skip reporting progress.)
    _, done = downloader.next_chunk()

  downloaded.seek(0)
  return downloaded
  #print 'Downloaded file contents are:', downloaded.read()
```

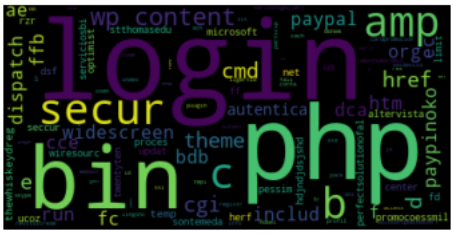
```
text_file = read_file("1SvLFtrpbxWgP7OTh5USrYQPebQRcCSEk")
```

```
document = text_file.read().decode('utf-8')
print(len(document))
print(document[0:100])
```

```
1037
thewhiskeydreg com wp content theme widescreen includ temp promocoessmil hdjndjdsjshd stthomasedu uc
```

```
wordcloud = WordCloud().generate(document)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
```

```
(-0.5, 399.5, 199.5, -0.5)
```



```
!apt-get update # to update ubuntu to correctly run apt install
!apt install chromium-chromedriver
!cp /usr/local/bin/chromedriver.exe /usr/bin
```

```
Hit:1 https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/ InRelease
Ign:2 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu2004/x86_64  InRelease
Hit:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64  InRelease
Hit:4 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu2004/x86_64  Release
Hit:5 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal InRelease
Hit:6 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:8 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:9 http://ppa.launchpad.net/cran/libgit2/ubuntu focal InRelease
Hit:11 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Hit:12 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu focal InRelease
Get:13 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2,910 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,286 kB]
Fetched 4,419 kB in 5s (921 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
chromium-chromedriver is already the newest version (1:85.0.4183.83-0ubuntu0.20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
cp: cannot stat '/usr/local/bin/chromedriver.exe': No such file or directory
```

```
import requests
import textwrap
```

```
cv = CountVectorizer()
```

```
feature = cv.fit_transform(data.text_sent)
```

```
feature[:5].toarray()
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
trainX, testX, trainY, testY = train_test_split(feature, data.Label,test_size=0.3, random_state=1)
#Xtrain, Xtest, Ytrain, Ytest = train_test_split(feature,data.Label, test_size=0.3, random_state=1) # 70% training and 30% test
```

```
trainX.shape
```

```
(354978, 350836)
```

```
testY.shape
```

```
(152134,)
```

```
trainX.shape
```

```
(354978, 350836)
```

```
testY.shape
```

```
(152134,)
```

```
lr = LogisticRegression()
```

```
lr.fit(trainX,trainY)
```

```
LogisticRegression()
```

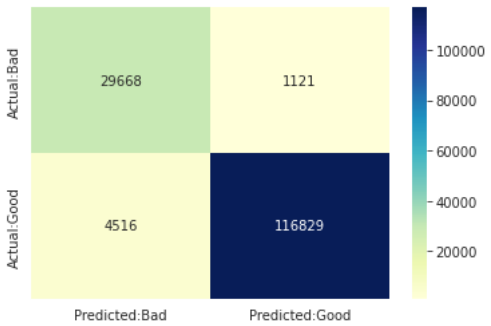
```
lr.score(testX,testY)
```

```
0.9629471387066665
```

```
Scores_ml = {}
Scores_ml['Logistic Regression'] = np.round(lr.score(testX,testY),2)
```

```
print('Training Accuracy :',lr.score(trainX,trainY))
print('Testing Accuracy :',lr.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(lr.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(lr.predict(testX), testY,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9756999025291708
Testing Accuracy : 0.9629471387066665

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.87      0.96      0.91     30789
        Good       0.99      0.96      0.98    121345

    accuracy                           0.96    152134
   macro avg       0.93      0.96      0.94    152134
weighted avg       0.97      0.96      0.96    152134


CONFUSION MATRIX
<matplotlib.axes._subplots.AxesSubplot at 0x7f6cb6388c40>
```




```
mnb = MultinomialNB()
```

```
mnb.fit(trainX,trainY)
```

```
MultinomialNB()
```

```
mnb.score(testX,testY)
```

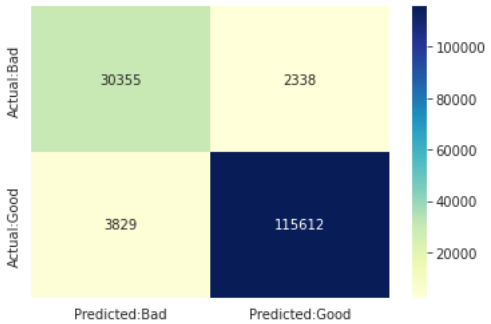
```
0.959463367820474
```

```
Scores_ml['MultinomialNB'] = np.round(mnb.score(testX,testY),2)
```

```
print('Training Accuracy :',mnb.score(trainX,trainY))
print('Testing Accuracy :',mnb.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(mnb.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
```

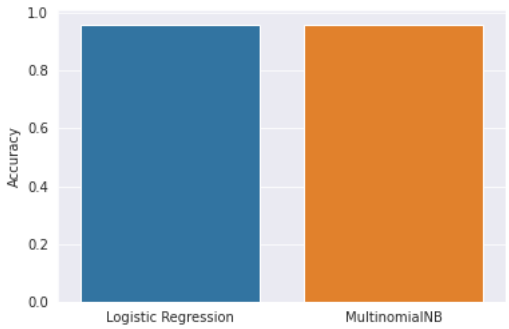
```
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(mnb.predict(testX), testY,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9748491455808529
Testing Accuracy : 0.959463367820474

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.89      0.93      0.91     32693
        Good       0.98      0.97      0.97    119441

    accuracy                           0.96    152134
   macro avg       0.93      0.95      0.94    152134
weighted avg       0.96      0.96      0.96    152134


CONFUSION MATRIX
<matplotlib.axes._subplots.AxesSubplot at 0x7f6cff866700>
```




```
acc = pd.DataFrame.from_dict(Scores_ml,orient = 'index',columns=['Accuracy'])
sns.set_style('darkgrid')
sns.barplot(acc.index,acc.Accuracy)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6c8d003d90>
```




```
pipeline_ls = make_pipeline(CountVectorizer(tokenizer = RegexpTokenizer(r'[A-Za-z]+').tokenize,stop_words='english'), LogisticRegression())
```

```
trainX, testX, trainY, testY = train_test_split(data.URL,data.Label)
```

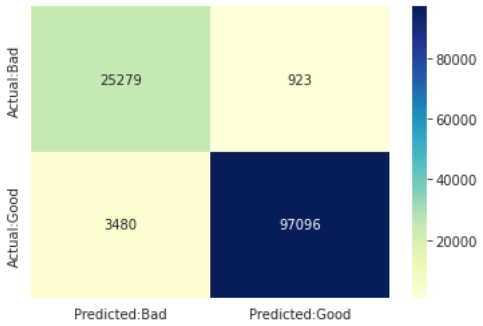
```
pipeline_ls.fit(trainX,trainY)
```

```
Pipeline(steps=[('countvectorizer',
                 CountVectorizer(stop_words='english',
                                 tokenizer=<bound method RegexpTokenizer.tokenize of RegexpTokenizer(pattern='[A-Za-z]+', gaps=False,
discard_empty=True, flags=re.UNICODE|re.MULTILINE|re.DOTALL)>)),
                ('logisticregression', LogisticRegression())])
```

```
print('Training Accuracy :',pipeline_ls.score(trainX,trainY))
print('Testing Accuracy :',pipeline_ls.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(pipeline_ls.predict(testX), testY),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(pipeline_ls.predict(testX), testY,
                            target_names =['Bad','Good']))
```

```
print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9798308854848633
Testing Accuracy : 0.9652699995267318

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.88      0.96      0.92     26202
        Good       0.99      0.97      0.98    100576

    accuracy                           0.97    126778
   macro avg       0.93      0.97      0.95    126778
weighted avg       0.97      0.97      0.97    126778


CONFUSION MATRIX
<matplotlib.axes._subplots.AxesSubplot at 0x7f6cb98b22b0>
```




```
pickle.dump(pipeline_ls,open('phishing.pkl','wb'))
```

```
loaded_model = pickle.load(open('phishing.pkl', 'rb'))
result = loaded_model.score(testX,testY)
print(result)
```

```
0.9652699995267318
```

```
predict_bad = ['http://sivaneshwaran.com','https://www.amazon.in/Raspberry-Pi-8GB-Desktop-Computer/dp/B08B9XS3B6/?_encoding=UTF8&pd_rd_w=i7KZx&content-
predict_good = ['www.w3.org/TR/speech-grammar/','www.textuality.com/index.html']
loaded_model = pickle.load(open('phishing.pkl', 'rb'))
#predict_bad = vectorizers.transform(predict_bad)
#predict_good = vectorizer.transform(predict_good)
result = loaded_model.predict(predict_bad)
result2 = loaded_model.predict(predict_good)
print(result)
print("-"*30)
print(result2)
```

```
['bad' 'bad']
------------------------------
['good' 'good']
```

```
data.columns
```

```
Index(['URL', 'Label', 'text_tokenized', 'text_stemmed', 'text_sent'], dtype='object')
```

```
type(data)
```

```
pandas.core.frame.DataFrame
```

```
data.Label
```

```
0         bad
1         bad
2         bad
3         bad
4         bad
         ...
507107    bad
507108    bad
507109    bad
507110    bad
507111    bad
Name: Label, Length: 507112, dtype: object
```

```
from sklearn.metrics import accuracy_score
```

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

```
data.dtypes
```

```
URL               object
Label             object
text_tokenized    object
text_stemmed      object
text_sent         object
dtype: object
```

```
import numpy
import re
```

```
df.describe()
```

	from	to
count	0	0
unique	0	0
top	NaN	NaN
freq	NaN	NaN

```
df.info
```

```
<bound method DataFrame.info of Empty DataFrame
Columns: [from, to]
Index: []>
```

```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(feature,data.Label, test_size=0.3, random_state=1) # 70% training and 30% test
```

```
# Decision Tree model
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth = 5)
# fit the model
tree.fit(Xtrain, Ytrain)
```

```
DecisionTreeClassifier(max_depth=5)
```

```
tree.score(Xtest,Ytest)
```

```
0.8500400962309543
```

```
Scores_ml['Decision Tree Classifier'] = np.round(tree.score(Xtest,Ytest),20)
```

```
print('Training Accuracy :',tree.score(Xtrain,Ytrain))
print('Testing Accuracy :',tree.score(Xtest,Ytest))
con_mat = pd.DataFrame(confusion_matrix(tree.predict(Xtest), Ytest),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(tree.predict(Xtest), Ytest,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.8479173357222137
Testing Accuracy : 0.8500400962309543

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.48      0.77      0.59     21194
        Good       0.96      0.86      0.91    130940

    accuracy                           0.85    152134
   macro avg       0.72      0.82      0.75    152134
weighted avg       0.89      0.85      0.86    152134
```

```
# Random Forest model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(max_depth=5)

# fit the model
forest.fit(Xtrain, Ytrain)
```

```
RandomForestClassifier(max_depth=5)
```

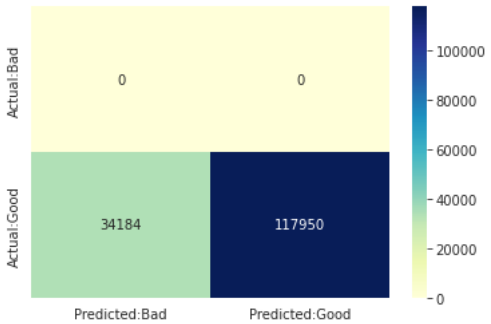
```
forest.score(Xtest,Ytest)
```

```
0.7753033509932034
```

```
Scores_ml['Random Forest Classifier'] = np.round(forest.score(Xtest,Ytest),20)
```

```
print('Training Accuracy :',forest.score(Xtrain,Ytrain))
print('Testing Accuracy :',forest.score(Xtest,Ytest))
con_mat = pd.DataFrame(confusion_matrix(forest.predict(Xtest), Ytest),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(forest.predict(Xtest), Ytest,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.7745465916197624
Testing Accuracy : 0.7753033509932034

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.00      0.00      0.00         0
        Good       1.00      0.78      0.87    152134

    accuracy                           0.78    152134
   macro avg       0.50      0.39      0.44    152134
weighted avg       1.00      0.78      0.87    152134


CONFUSION MATRIX
<matplotlib.axes._subplots.AxesSubplot at 0x7f6caf9b8f40>
```




```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2017 NVIDIA Corporation
Built on Fri_Sep__1_21:08:03_CDT_2017
Cuda compilation tools, release 9.0, V9.0.176
```

```
!pip install thundersvm
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting thundersvm
  Downloading thundersvm-0.3.12-py3-none-any.whl (507 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.4/507.4 KB 3.2 MB/s eta 0:00:00
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from thundersvm) (1.7.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from thundersvm) (1.21.6)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from thundersvm) (1.0.2)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->thundersvm) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->thundersvm) (3.1.0)
Installing collected packages: thundersvm
Successfully installed thundersvm-0.3.12
```

```
!pip install thundersvm-cpu
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting thundersvm-cpu
  Downloading thundersvm_cpu-0.3.3-py3-none-any.whl (227 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 227.3/227.3 KB 5.5 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from thundersvm-cpu) (1.21.6)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from thundersvm-cpu) (1.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from thundersvm-cpu) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->thundersvm-cpu) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->thundersvm-cpu) (1.2.0)
Installing collected packages: thundersvm-cpu
Successfully installed thundersvm-cpu-0.3.3
```

```
from thundersvm import SVC
from sklearn.datasets import make_classification
X,Y= make_classification(n_samples=100000, n_features=20, n_informative=17, n_redundant=3, random_state=5)
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X,Y, test_size=0.3, random_state=5) #put 80% data in training set
# Initialize model
svcs = SVC(C=100, kernel='rbf')
# Fit the model to training data
svcs.fit(Xtrain, Ytrain)
# Check test set accuracy
svcs.score(Xtest, Ytest)
```

```
0.9795666666666667
```

```
Scores_ml['Support Vector Classification'] = np.round(svcs.score(Xtest,Ytest),2)
print('Training Accuracy :',svcs.score(Xtrain,Ytrain))
print('Testing Accuracy :',svcs.score(Xtest,Ytest))
con_mat = pd.DataFrame(confusion_matrix(svcs.predict(Xtest), Ytest),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(svcs.predict(Xtest), Ytest,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

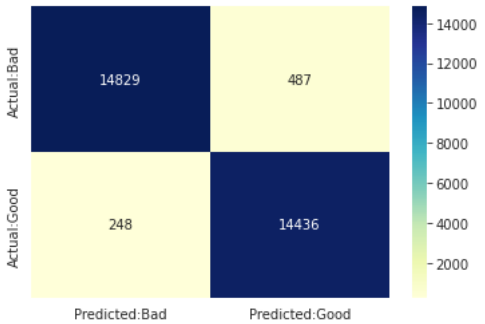
```
Training Accuracy : 1.0
```

```
knn = KNeighborsClassifier(n_neighbors = 5)
knn.fit(Xtrain,Ytrain)
knn.score(Xtest,Ytest)
```

```
0.9755
```

```
Scores_ml['K-Nearest Neighbor'] = np.round(knn.score(Xtest,Ytest),2)
print('Training Accuracy :',knn.score(Xtrain,Ytrain))
print('Testing Accuracy :',knn.score(Xtest,Ytest))
con_mat = pd.DataFrame(confusion_matrix(knn.predict(Xtest), Ytest),
            columns = ['Predicted:Bad', 'Predicted:Good'],
            index = ['Actual:Bad', 'Actual:Good'])


print('\nCLASSIFICATION REPORT\n')
print(classification_report(knn.predict(Xtest), Ytest,
                            target_names =['Bad','Good']))

print('\nCONFUSION MATRIX')
plt.figure(figsize= (6,4))
sns.heatmap(con_mat, annot = True,fmt='d',cmap="YlGnBu")
```

```
Training Accuracy : 0.9833142857142857
Testing Accuracy : 0.9755

CLASSIFICATION REPORT

              precision    recall  f1-score   support

         Bad       0.98      0.97      0.98     15316
        Good       0.97      0.98      0.98     14684

    accuracy                           0.98     30000
   macro avg       0.98      0.98      0.98     30000
weighted avg       0.98      0.98      0.98     30000


CONFUSION MATRIX
<matplotlib.axes._subplots.AxesSubplot at 0x7f6cb99c3eb0>
```




```
predict_bad = ['steamglfts.hut2.ru/']
predict_good = ['www.auburnmedia.com/wordpress/']
loaded_model = pickle.load(open('phishing.pkl', 'rb'))
result = loaded_model.predict(predict_bad)
result2 = loaded_model.predict(predict_good)
print(result)
print("-"*10)
print(result2)
```

```
['bad']
----------
['good']
```

0s completed at 9:10 PM