Building a Phishing Detection Model

Preprocess of the dataset

Name	Sivanesh S
BITS WILP Reg. ID	2022ac05046
Email ID	2022ac05046@wilp.bits-pilani.ac.in

1. Import Libraries/Dataset

In [1]:
```
# importing dataset

import os
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn import metrics

import matplotlib.pyplot as plt
import numpy as np

import ipaddress
import re
import urllib.request
from bs4 import BeautifulSoup
import socket
import requests
import google
import whois
from datetime import date, datetime
import time
from dateutil.parser import parse as date_parse
from urllib.parse import urlparse
```

Data Preparation

2. Download the phishing dataset and import it into your Python environment. Explore the dataset to understand its structure, features, and target labels.

In [2]:
```
# Loading the dataset

datasetpath = os.path.join('C:' + os.sep, 'Users' + os.sep, 'user' + os.sep, 'Documents'
print(datasetpath)
```

```
C:\Users\user\Documents\AIML CS\archive\phishing.csv
```

In [3]:
```
cs_data = pd.read_csv(datasetpath, delimiter=',')
```

```
cs_data.head(10)
```

Out[3]:

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegl
0	0	1	1	1	1	1	-1	0	1	
1	1	1	0	1	1	1	-1	-1	-1	
2	2	1	0	1	1	1	-1	-1	-1	
3	3	1	0	-1	1	1	-1	1	1	
4	4	-1	0	-1	1	-1	-1	1	1	
5	5	1	0	-1	1	1	-1	-1	-1	
6	6	1	0	1	1	1	-1	-1	-1	
7	7	1	0	-1	1	1	-1	1	1	
8	8	1	1	-1	1	1	-1	-1	1	
9	9	1	1	1	1	1	-1	0	1	

10 rows × 32 columns

In [4]:
```
# check the type

cs_data.dtypes
```

Out[4]:
```
Index                int64
UsingIP              int64
LongURL              int64
ShortURL             int64
Symbol@              int64
Redirecting//        int64
PrefixSuffix-        int64
SubDomains           int64
HTTPS                int64
DomainRegLen         int64
Favicon              int64
NonStdPort           int64
HTTPSDomainURL       int64
RequestURL           int64
AnchorURL            int64
LinksInScriptTags    int64
ServerFormHandler    int64
InfoEmail            int64
AbnormalURL          int64
WebsiteForwarding    int64
StatusBarCust        int64
DisableRightClick    int64
UsingPopupWindow     int64
IframeRedirection    int64
AgeofDomain          int64
DNSRecording         int64
WebsiteTraffic       int64
PageRank             int64
GoogleIndex          int64
LinksPointingToPage  int64
StatsReport          int64
class                int64
dtype: object
```

In [5]:
```
# exploration of the dataset

cs_data.describe()
```

Out[5]:

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDo
count	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.000000	11054.0
mean	5526.500000	0.313914	-0.633345	0.738737	0.700561	0.741632	-0.734938	0.0
std	3191.159272	0.949495	0.765973	0.674024	0.713625	0.670837	0.678165	0.8
min	0.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.0
25%	2763.250000	-1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	-1.0
50%	5526.500000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	0.0
75%	8289.750000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	1.0
max	11053.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

8 rows × 32 columns

In [6]:
```
# check the rows and columns

cs_data.shape
```

Out[6]:
```
(11054, 32)
```

Data Preprocessing

2. Clean the data: Remove duplicates, handle missing values, etc. Encode categorical variables: Convert categorical features into numerical format if needed. Split the dataset into features (X) and target labels (y)

In [7]:
```
cs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11054 entries, 0 to 11053
Data columns (total 32 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Index                11054 non-null  int64
 1   UsingIP              11054 non-null  int64
 2   LongURL              11054 non-null  int64
 3   ShortURL             11054 non-null  int64
 4   Symbol@              11054 non-null  int64
 5   Redirecting//        11054 non-null  int64
 6   PrefixSuffix-        11054 non-null  int64
 7   SubDomains           11054 non-null  int64
 8   HTTPS                11054 non-null  int64
 9   DomainRegLen         11054 non-null  int64
 10  Favicon              11054 non-null  int64
 11  NonStdPort           11054 non-null  int64
 12  HTTPSDomainURL       11054 non-null  int64
 13  RequestURL           11054 non-null  int64
 14  AnchorURL            11054 non-null  int64
 15  LinksInScriptTags    11054 non-null  int64
 16  ServerFormHandler    11054 non-null  int64
 17  InfoEmail            11054 non-null  int64
 18  AbnormalURL          11054 non-null  int64
 19  WebsiteForwarding    11054 non-null  int64
 20  StatusBarCust        11054 non-null  int64
 21  DisableRightClick    11054 non-null  int64
 22  UsingPopupWindow     11054 non-null  int64
 23  IframeRedirection    11054 non-null  int64
 24  AgeofDomain          11054 non-null  int64
```

```
 25  DNSRecording         11054 non-null  int64
 26  WebsiteTraffic       11054 non-null  int64
 27  PageRank             11054 non-null  int64
 28  GoogleIndex          11054 non-null  int64
 29  LinksPointingToPage  11054 non-null  int64
 30  StatsReport          11054 non-null  int64
 31  class                11054 non-null  int64
dtypes: int64(32)
memory usage: 2.7 MB
```

In [8]:
```
# check for null values in the dataset

cs_data.isnull().sum()
```

Out[8]:
```
Index                0
UsingIP              0
LongURL              0
ShortURL             0
Symbol@              0
Redirecting//        0
PrefixSuffix-        0
SubDomains           0
HTTPS                0
DomainRegLen         0
Favicon              0
NonStdPort           0
HTTPSDomainURL       0
RequestURL           0
AnchorURL            0
LinksInScriptTags    0
ServerFormHandler    0
InfoEmail            0
AbnormalURL          0
WebsiteForwarding    0
StatusBarCust        0
DisableRightClick    0
UsingPopupWindow     0
IframeRedirection    0
AgeofDomain          0
DNSRecording         0
WebsiteTraffic       0
PageRank             0
GoogleIndex          0
LinksPointingToPage  0
StatsReport          0
class                0
dtype: int64
```

In [9]:
```
# Drop the 'class' column to create a new DataFrame X

X = cs_data.drop(["class","Index"],axis =1)
y = cs_data["class"]
X.head(5)
```

Out[9]:

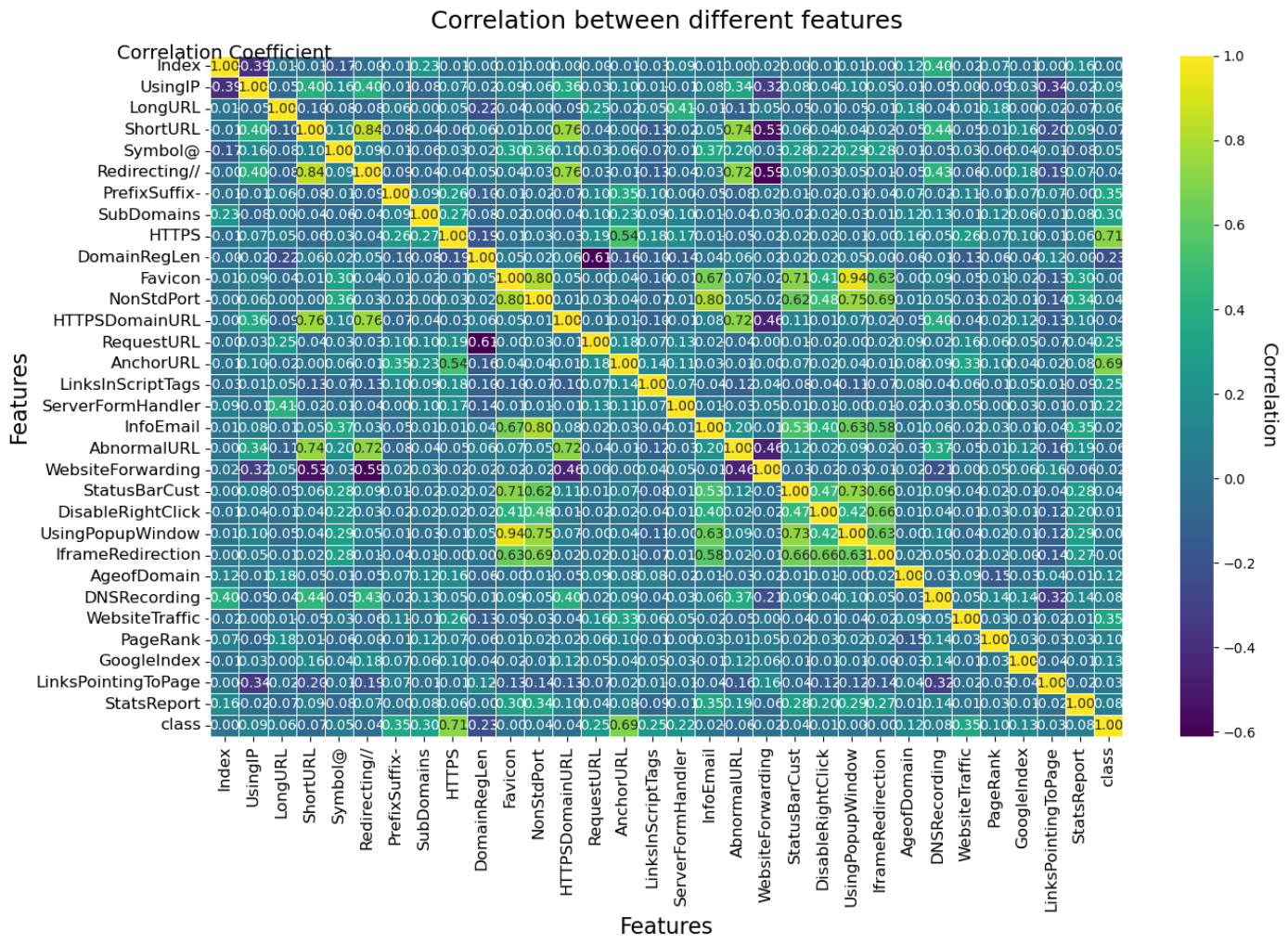
	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	Fa
0	1	1	1	1	1	-1	0	1	-1	
1	1	0	1	1	1	-1	-1	-1	-1	
2	1	0	1	1	1	-1	-1	-1	1	
3	1	0	-1	1	1	-1	1	1	-1	
4	-1	0	-1	1	-1	-1	1	1	-1	

5 rows × 30 columns

In [10]:
```
y.head(5)
print(y)
```

```
0       -1
1       -1
2       -1
3        1
4        1
        ..
11049    1
11050   -1
11051   -1
11052   -1
11053   -1
Name: class, Length: 11054, dtype: int64
```

In [11]:
```
# display the dataset

fig, ax = plt.subplots(figsize=(15, 9))

# Customize the heatmap using seaborn
heatmap = sns.heatmap(cs_data.corr(), annot=True, cmap='viridis', linewidths=.5, fmt='.2
heatmap.set_title('Correlation between different features', fontsize=18, pad=20, color='
heatmap.text(0.5, -0.1, "Correlation Coefficient", ha="center", va="center", fontsize=14

cbar = heatmap.collections[0].colorbar
cbar.set_label('Correlation', rotation=270, labelpad=15, fontsize=14, color='black')


ax.tick_params(axis='both', labelsize=12, colors='black')

ax.set_xlabel('Features', fontsize=16, color='black')
ax.set_ylabel('Features', fontsize=16, color='black')

plt.show()
```



In [12]:
```
corr = cs_data.corr()
corr.head(5)
```

Out[12]:

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTT
Index	1.000000	-0.388620	0.006441	-0.006221	-0.169437	-0.003771	-0.007402	0.233936	-0.0068
UsingIP	-0.388620	1.000000	-0.052159	0.403547	0.158766	0.397220	-0.005306	-0.080921	0.0712
LongURL	0.006441	-0.052159	1.000000	-0.097976	-0.075205	-0.080788	0.055334	0.004249	0.0490
ShortURL	-0.006221	0.403547	-0.097976	1.000000	0.104433	0.843149	-0.080458	-0.041874	-0.0613
Symbol@	-0.169437	0.158766	-0.075205	0.104433	1.000000	0.087086	-0.011711	-0.058931	0.0312

5 rows × 32 columns

In [13]:
```
corr['class']=abs(corr['class'])
corr.head(5)
```

Out[13]:

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTT
Index	1.000000	-0.388620	0.006441	-0.006221	-0.169437	-0.003771	-0.007402	0.233936	-0.0068
UsingIP	-0.388620	1.000000	-0.052159	0.403547	0.158766	0.397220	-0.005306	-0.080921	0.0712
LongURL	0.006441	-0.052159	1.000000	-0.097976	-0.075205	-0.080788	0.055334	0.004249	0.0490
ShortURL	-0.006221	0.403547	-0.097976	1.000000	0.104433	0.843149	-0.080458	-0.041874	-0.0613
Symbol@	-0.169437	0.158766	-0.075205	0.104433	1.000000	0.087086	-0.011711	-0.058931	0.0312

5 rows × 32 columns

In [14]:
```
incCorr=corr.sort_values(by='class',ascending=False)
incCorr.head(5)
```

Out[14]:

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	
class	0.000802	0.094033	0.057661	-0.067931	0.052994	-0.038885	0.348588	0.298231	0.1
HTTPS	-0.006899	0.071255	0.049033	-0.061383	0.031275	-0.036536	0.261366	0.267531	1.(
AnchorURL	-0.005275	0.099701	-0.023153	0.000607	0.057968	-0.005341	0.348854	0.229374	0.5
PrefixSuffix-	-0.007402	-0.005306	0.055334	-0.080458	-0.011711	-0.085709	1.000000	0.087852	0.2
WebsiteTraffic	-0.015147	0.002728	0.009296	-0.047025	0.032981	-0.062761	0.110555	-0.005948	0.2

5 rows × 32 columns

Feature Extraction

3. Extract relevant features from URLs and email content. Transform URLs and email content into numerical representations (e.g., using TF-IDF).

In [15]:
```
# Sorting columns in decreasing order based on association

incCorr['class']
```

Out[15]:
```
class                 1.000000
HTTPS                 0.714704
AnchorURL             0.692895
PrefixSuffix-         0.348588
WebsiteTraffic        0.346003
SubDomains            0.298231
RequestURL            0.253478
LinksInScriptTags     0.248415
DomainRegLen          0.225879
ServerFormHandler     0.221380
GoogleIndex           0.129000
AgeofDomain           0.121402
PageRank              0.104593
UsingIP               0.094033
StatsReport           0.079632
DNSRecording          0.075579
ShortURL              0.067931
AbnormalURL           0.060751
LongURL               0.057661
Symbol@               0.052994
StatusBarCust         0.041878
HTTPSDomainURL        0.040096
Redirecting//         0.038885
NonStdPort            0.036461
LinksPointingToPage   0.032694
WebsiteForwarding     0.020151
InfoEmail             0.018039
DisableRightClick     0.012675
IframeRedirection     0.003362
Index                 0.000802
Favicon               0.000231
UsingPopupWindow      0.000136
Name: class, dtype: float64
```

In [16]:
```
# Select the top N attributes based on correlation.
```

```
top_10_features = incCorr[1:11].index
top_20_features = incCorr[1:21].index

# Print or use the selected features as needed
print("Top 10 features based on correlation:")
print(top_10_features)

print("\nTop 20 features based on correlation:")
print(top_20_features)
```

```
Top 10 features based on correlation:
Index(['HTTPS', 'AnchorURL', 'PrefixSuffix-', 'WebsiteTraffic', 'SubDomains',
       'RequestURL', 'LinksInScriptTags', 'DomainRegLen', 'ServerFormHandler',
       'GoogleIndex'],
      dtype='object')

Top 20 features based on correlation:
Index(['HTTPS', 'AnchorURL', 'PrefixSuffix-', 'WebsiteTraffic', 'SubDomains',
       'RequestURL', 'LinksInScriptTags', 'DomainRegLen', 'ServerFormHandler',
       'GoogleIndex', 'AgeofDomain', 'PageRank', 'UsingIP', 'StatsReport',
       'DNSRecording', 'ShortURL', 'AbnormalURL', 'LongURL', 'Symbol@',
       'StatusBarCust'],
      dtype='object')
```

Model Selection

4. Choose suitable machine learning algorithms for phishing detection (e.g., Logistic Regression, Naive Bayes, Random Forest, etc.).

In [17]:
```
# Define the model

ML_Model = []
accuracy = []
f1_score = []
precision = []

def storeResults(model, a,b,c):
  ML_Model.append(model)
  accuracy.append(round(a, 3))
  f1_score.append(round(b, 3))
  precision.append(round(c, 3))
```

Model Training

5. Split the data into training and testing/validation sets. Train the selected models on the training data.

In [18]:
```
def knn_evaluation(X):
  x=[a for a in range(1,10,2)]
  knntrain=[]
  knntest=[]
  from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_stat
  X_train.shape, y_train.shape, X_test.shape, y_test.shape
  for i in range(1,10,2):
    from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    y_train_knn = knn.predict(X_train)
    y_test_knn = knn.predict(X_test)
    acc_train_knn = metrics.accuracy_score(y_train,y_train_knn)
    acc_test_knn = metrics.accuracy_score(y_test,y_test_knn)
```

```
    print("K-Nearest Neighbors with k={}: Accuracy on training Data: {:.3f}".format(i,ac
    print("K-Nearest Neighbors with k={}: Accuracy on test Data: {:.3f}".format(i,acc_te
    knntrain.append(acc_train_knn)
    knntest.append(acc_test_knn)
    print()
  import matplotlib.pyplot as plt
  plt.plot(x,knntrain,label="Train accuracy")
  plt.plot(x,knntest,label="Test accuracy")
  plt.legend()
  plt.show()
```

In [19]:
```
Xmain=X
Xten=X[top_10_features]
Xtwen=X[top_20_features]

knn_evaluation(Xmain)
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=1: Accuracy on training Data: 0.989
K-Nearest Neighbors with k=1: Accuracy on test Data: 0.956
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=3: Accuracy on training Data: 0.976
K-Nearest Neighbors with k=3: Accuracy on test Data: 0.946
```

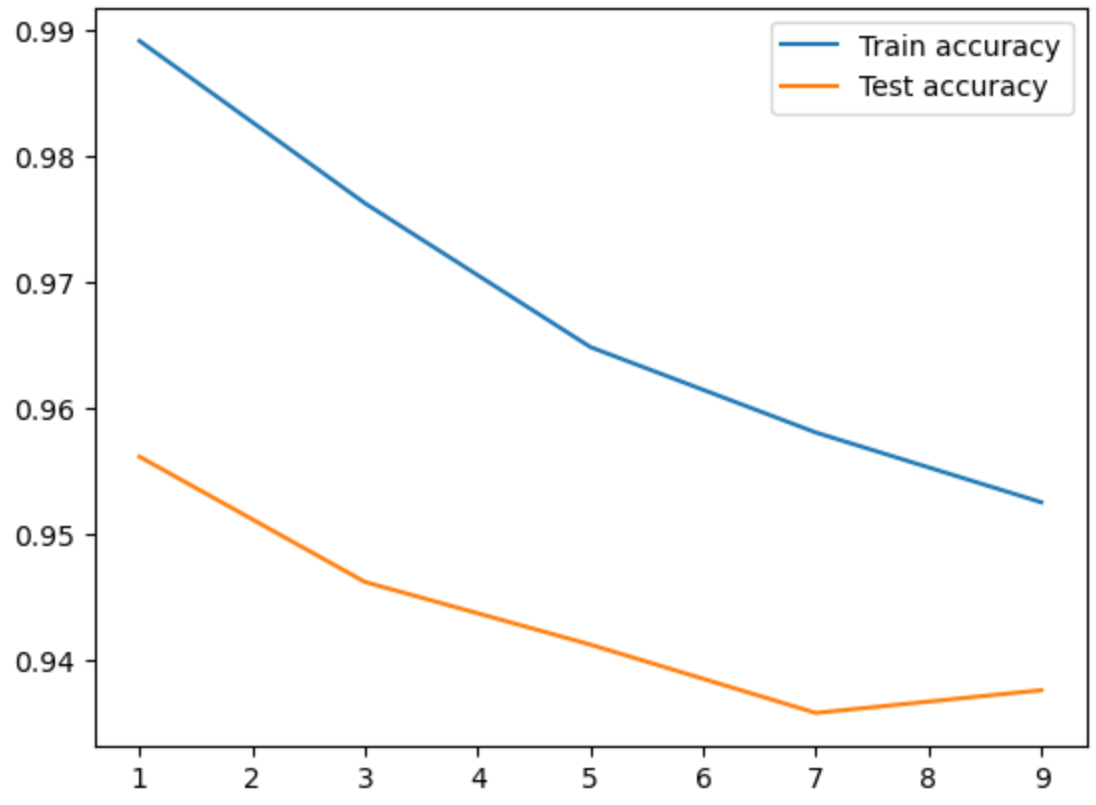
```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
```

```
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=5: Accuracy on training Data: 0.965
K-Nearest Neighbors with k=5: Accuracy on test Data: 0.941
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=7: Accuracy on training Data: 0.958
K-Nearest Neighbors with k=7: Accuracy on test Data: 0.936
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=9: Accuracy on training Data: 0.953
K-Nearest Neighbors with k=9: Accuracy on test Data: 0.938
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```



In [20]:
```
knn_evaluation(Xten)
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=1: Accuracy on training Data: 0.943
K-Nearest Neighbors with k=1: Accuracy on test Data: 0.928
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=3: Accuracy on training Data: 0.946
K-Nearest Neighbors with k=3: Accuracy on test Data: 0.933
```

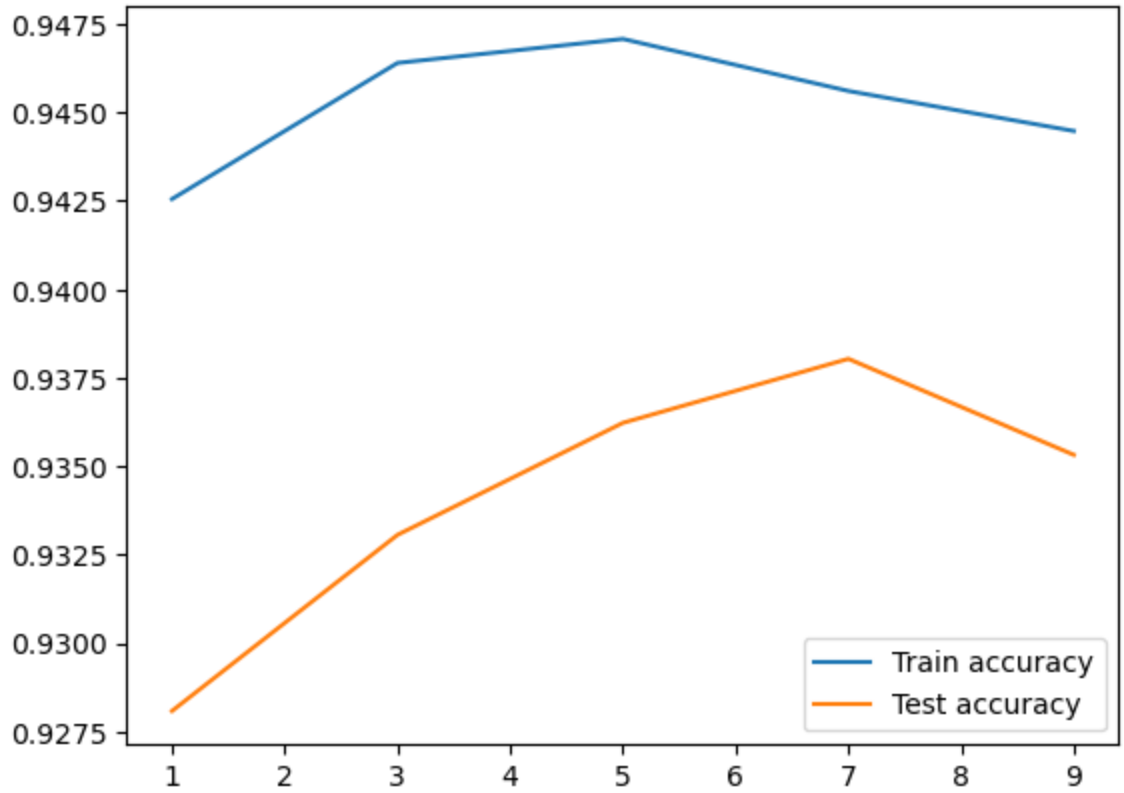
```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=5: Accuracy on training Data: 0.947
K-Nearest Neighbors with k=5: Accuracy on test Data: 0.936
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
```

```
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=7: Accuracy on training Data: 0.946
K-Nearest Neighbors with k=7: Accuracy on test Data: 0.938
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=9: Accuracy on training Data: 0.944
K-Nearest Neighbors with k=9: Accuracy on test Data: 0.935
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

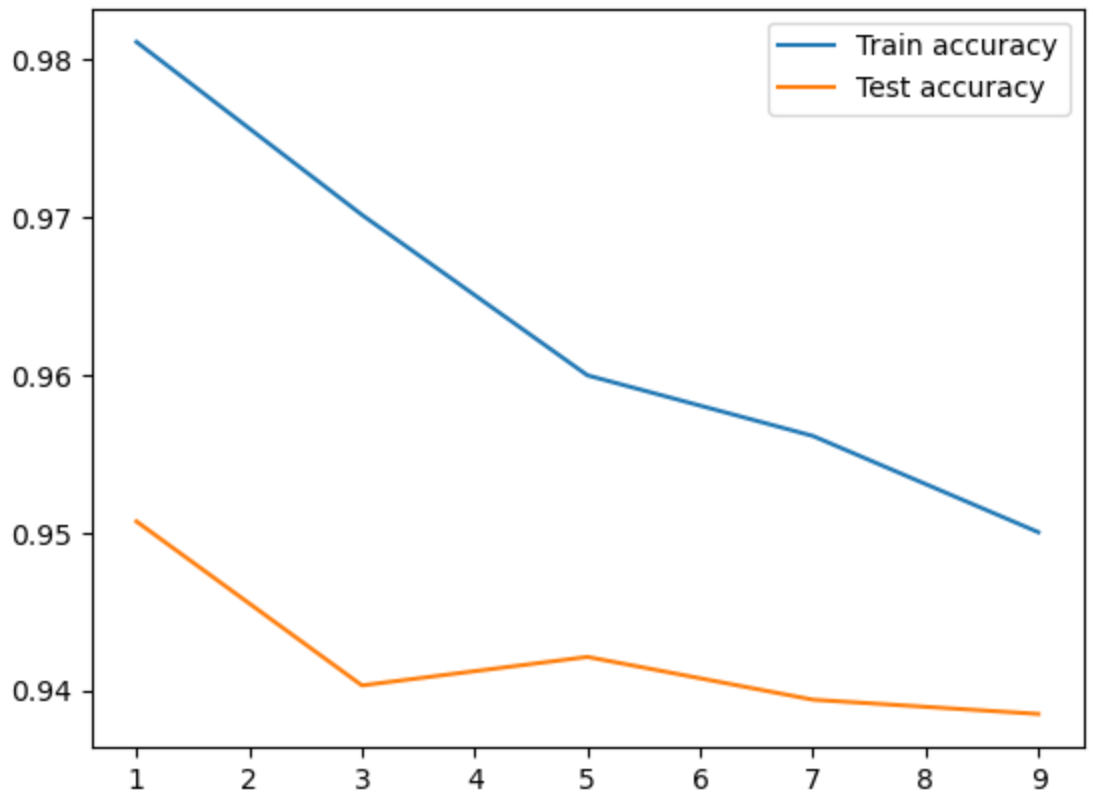


In [21]:
```
knn_evaluation(Xtwen)
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
```

```
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=1: Accuracy on training Data: 0.981
K-Nearest Neighbors with k=1: Accuracy on test Data: 0.951

C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=3: Accuracy on training Data: 0.970
K-Nearest Neighbors with k=3: Accuracy on test Data: 0.940

C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=5: Accuracy on training Data: 0.960
K-Nearest Neighbors with k=5: Accuracy on test Data: 0.942

C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=7: Accuracy on training Data: 0.956
K-Nearest Neighbors with k=7: Accuracy on test Data: 0.939

C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
```

```
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
K-Nearest Neighbors with k=9: Accuracy on training Data: 0.950
K-Nearest Neighbors with k=9: Accuracy on test Data: 0.938
```



As K value increases, Accuracy keeps on decreasing. With K as 1 giving maximum accuracy, it will overfit. The value of k in the KNN algorithm is related to the error rate of the model. A small value of K could lead to overfitting as well as a big value of k can lead to underfitting. X[10] has comparatively more accuracy.

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
X_train.shape, y_train.shape, X_test.shape, y_test.shape

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train,y_train)

y_train_knn = knn.predict(X_train)
y_test_knn = knn.predict(X_test)

acc_train_knn = metrics.accuracy_score(y_train,y_train_knn)
acc_test_knn = metrics.accuracy_score(y_test,y_test_knn)

f1_score_train_knn = metrics.f1_score(y_train,y_train_knn)
f1_score_test_knn = metrics.f1_score(y_test,y_test_knn)

precision_score_train_knn = metrics.precision_score(y_train,y_train_knn)
precision_score_test_knn = metrics.precision_score(y_test,y_test_knn)
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
reWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behav
ior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior
will change: the default value of `keepdims` will become False, the `axis` over which th
e statistic is taken will be eliminated, and the value None will no longer be accepted.
Set `keepdims` to True or False to avoid this warning.
```

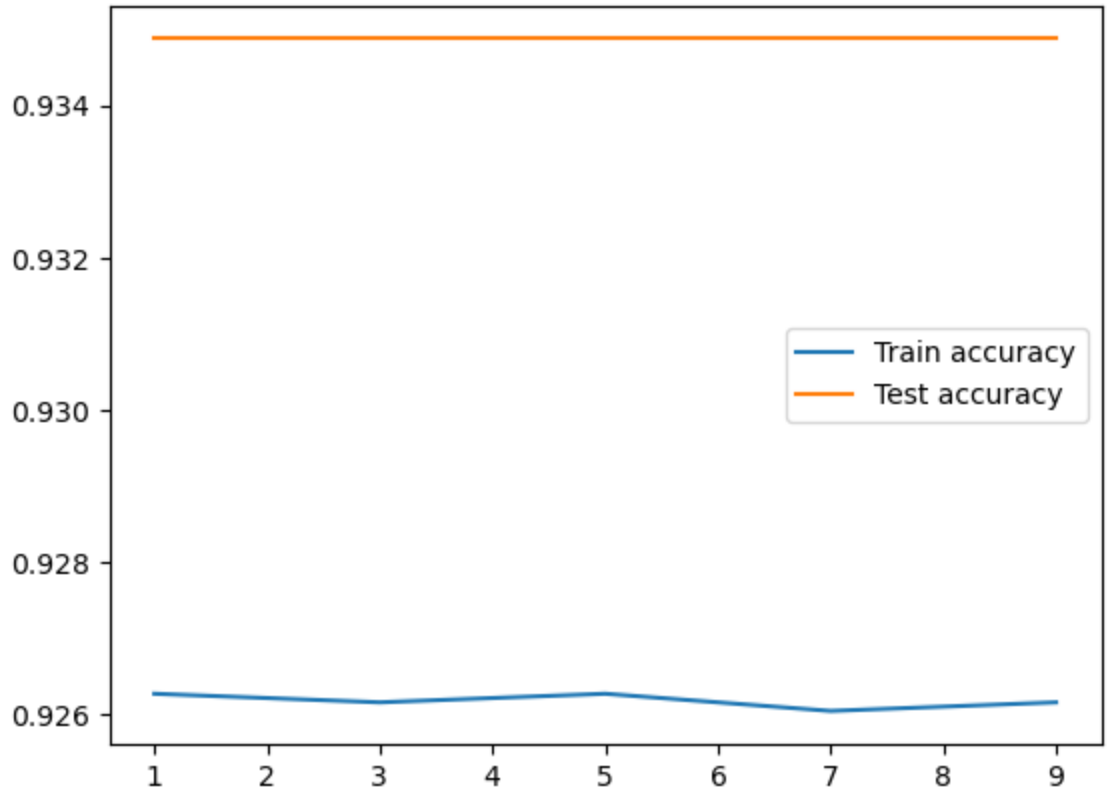
```
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\user\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

In [23]:
```
storeResults('K-Nearest Neighbors',acc_test_knn,f1_score_test_knn,precision_score_train_
```

In [24]:
```
# Support Vector Machine

def svm_evaluation(X, y):
    x=[a for a in range(1,10,2)]
    svmtrain=[]
    svmtest=[]
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_st
    X_train.shape, y_train.shape, X_test.shape, y_test.shape
    from sklearn.svm import SVC
    for i in range(1,10,2):
        svm = SVC(kernel='linear', C=i)
        svm.fit(X_train, y_train)
        y_train_svm = svm.predict(X_train)
        y_test_svm = svm.predict(X_test)
        acc_train_svm = metrics.accuracy_score(y_train, y_train_svm)
        acc_test_svm = metrics.accuracy_score(y_test, y_test_svm)
        print("SVM with C={}: Accuracy on training Data: {:.3f}".format(i,acc_train_svm)
        print("SVM with C={}: Accuracy on test Data: {:.3f}".format(i,acc_test_svm))
        svmtrain.append(acc_train_svm)
        svmtest.append(acc_test_svm)
        print()
    import matplotlib.pyplot as plt
    plt.plot(x,svmtrain,label="Train accuracy")
    plt.plot(x,svmtest,label="Test accuracy")
    plt.legend()
    plt.show()
```

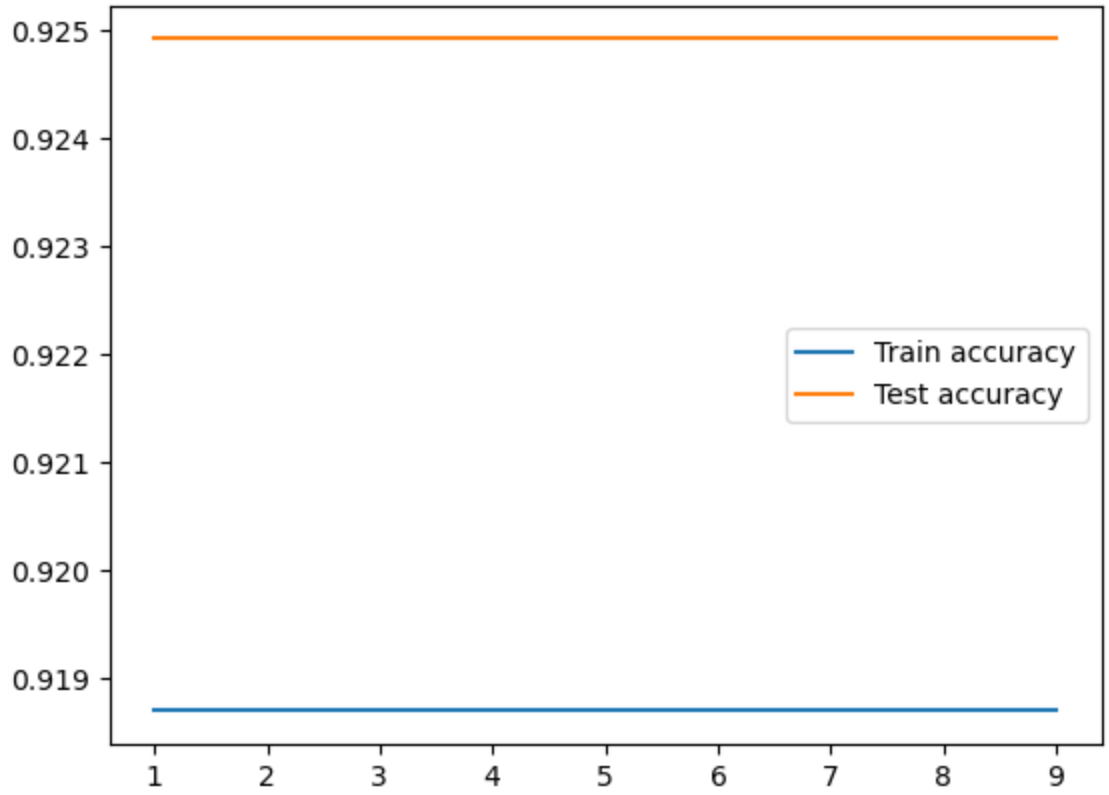
In [25]:
```
Xmain=X
Xten=X[top_10_features]
Xtwen=X[top_20_features]

svm_evaluation(Xmain,y)
```

```
SVM with C=1: Accuracy on training Data: 0.926
SVM with C=1: Accuracy on test Data: 0.935

SVM with C=3: Accuracy on training Data: 0.926
SVM with C=3: Accuracy on test Data: 0.935

SVM with C=5: Accuracy on training Data: 0.926
SVM with C=5: Accuracy on test Data: 0.935

SVM with C=7: Accuracy on training Data: 0.926
SVM with C=7: Accuracy on test Data: 0.935

SVM with C=9: Accuracy on training Data: 0.926
SVM with C=9: Accuracy on test Data: 0.935
```



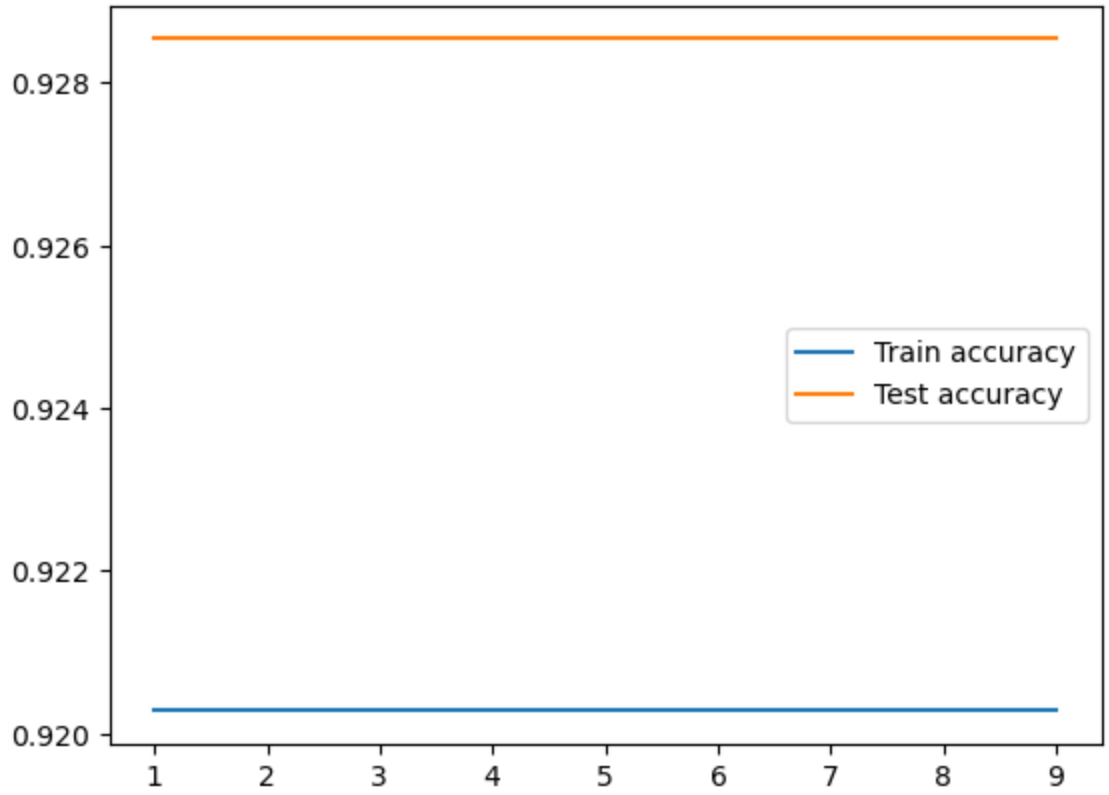
In [26]:
```
svm_evaluation(Xten,y)
```

```
SVM with C=1: Accuracy on training Data: 0.919
SVM with C=1: Accuracy on test Data: 0.925

SVM with C=3: Accuracy on training Data: 0.919
SVM with C=3: Accuracy on test Data: 0.925

SVM with C=5: Accuracy on training Data: 0.919
SVM with C=5: Accuracy on test Data: 0.925

SVM with C=7: Accuracy on training Data: 0.919
SVM with C=7: Accuracy on test Data: 0.925

SVM with C=9: Accuracy on training Data: 0.919
SVM with C=9: Accuracy on test Data: 0.925
```



In [27]:
```
svm_evaluation(Xtwen,y)
```

```
SVM with C=1: Accuracy on training Data: 0.920
SVM with C=1: Accuracy on test Data: 0.929

SVM with C=3: Accuracy on training Data: 0.920
SVM with C=3: Accuracy on test Data: 0.929

SVM with C=5: Accuracy on training Data: 0.920
SVM with C=5: Accuracy on test Data: 0.929

SVM with C=7: Accuracy on training Data: 0.920
SVM with C=7: Accuracy on test Data: 0.929

SVM with C=9: Accuracy on training Data: 0.920
SVM with C=9: Accuracy on test Data: 0.929
```



In [28]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

svm = SVC(kernel='linear', C=1, random_state=42)
svm.fit(X_train, y_train)


y_train_svm = svm.predict(X_train)
y_test_svm = svm.predict(X_test)


acc_train_svm = metrics.accuracy_score(y_train, y_train_svm)
acc_test_svm = metrics.accuracy_score(y_test, y_test_svm)

f1_score_train_svm = metrics.f1_score(y_train, y_train_svm)
f1_score_test_svm = metrics.f1_score(y_test, y_test_svm)

precision_score_train_svm = metrics.precision_score(y_train, y_train_svm)
precision_score_test_svm = metrics.precision_score(y_test, y_test_svm)

print("SVM with C={}: Accuracy on training data: {:.3f}".format(1, acc_train_svm))
print("SVM with C={}: Accuracy on test data: {:.3f}".format(1, acc_test_svm))
print("SVM with C={}: F1 score on training data: {:.3f}".format(1, f1_score_train_svm))
print("SVM with C={}: F1 score on test data: {:.3f}".format(1, f1_score_test_svm))
print("SVM with C={}: Precision on training data: {:.3f}".format(1, precision_score_trai
print("SVM with C={}: Precision on test data: {:.3f}".format(1, precision_score_test_svm
```

```
SVM with C=1: Accuracy on training data: 0.926
SVM with C=1: Accuracy on test data: 0.935
SVM with C=1: F1 score on training data: 0.934
SVM with C=1: F1 score on test data: 0.942
SVM with C=1: Precision on training data: 0.926
SVM with C=1: Precision on test data: 0.930
```

In [29]:
```
storeResults('Support Vector Machines',acc_test_svm,f1_score_test_svm,precision_score_tr
```

In [30]:
```
# Gradient Boost (Boosting Based)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

Out[30]:
```
((8843, 30), (8843,), (2211, 30), (2211,))
```

In [31]:
```
gbc = GradientBoostingClassifier(max_depth=4,learning_rate=0.7)
gbc.fit(X_train,y_train)
```

Out[31]:
```
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

In [32]:
```
y_train_gbc = gbc.predict(X_train)
y_test_gbc = gbc.predict(X_test)
```

In [33]:
```
acc_train_gbc = metrics.accuracy_score(y_train,y_train_gbc)
acc_test_gbc = metrics.accuracy_score(y_test,y_test_gbc)
print("Gradient Boosting Classifier : Accuracy on training Data: {:.3f}".format(acc_trai
print("Gradient Boosting Classifier : Accuracy on test Data: {:.3f}".format(acc_test_gbc
print()

f1_score_train_gbc = metrics.f1_score(y_train,y_train_gbc)
f1_score_test_gbc = metrics.f1_score(y_test,y_test_gbc)

precision_score_train_gbc = metrics.precision_score(y_train,y_train_gbc)
precision_score_test_gbc = metrics.precision_score(y_test,y_test_gbc)

storeResults('Gradient Boosting Classifier',acc_test_gbc,f1_score_test_gbc,precision_sco
```

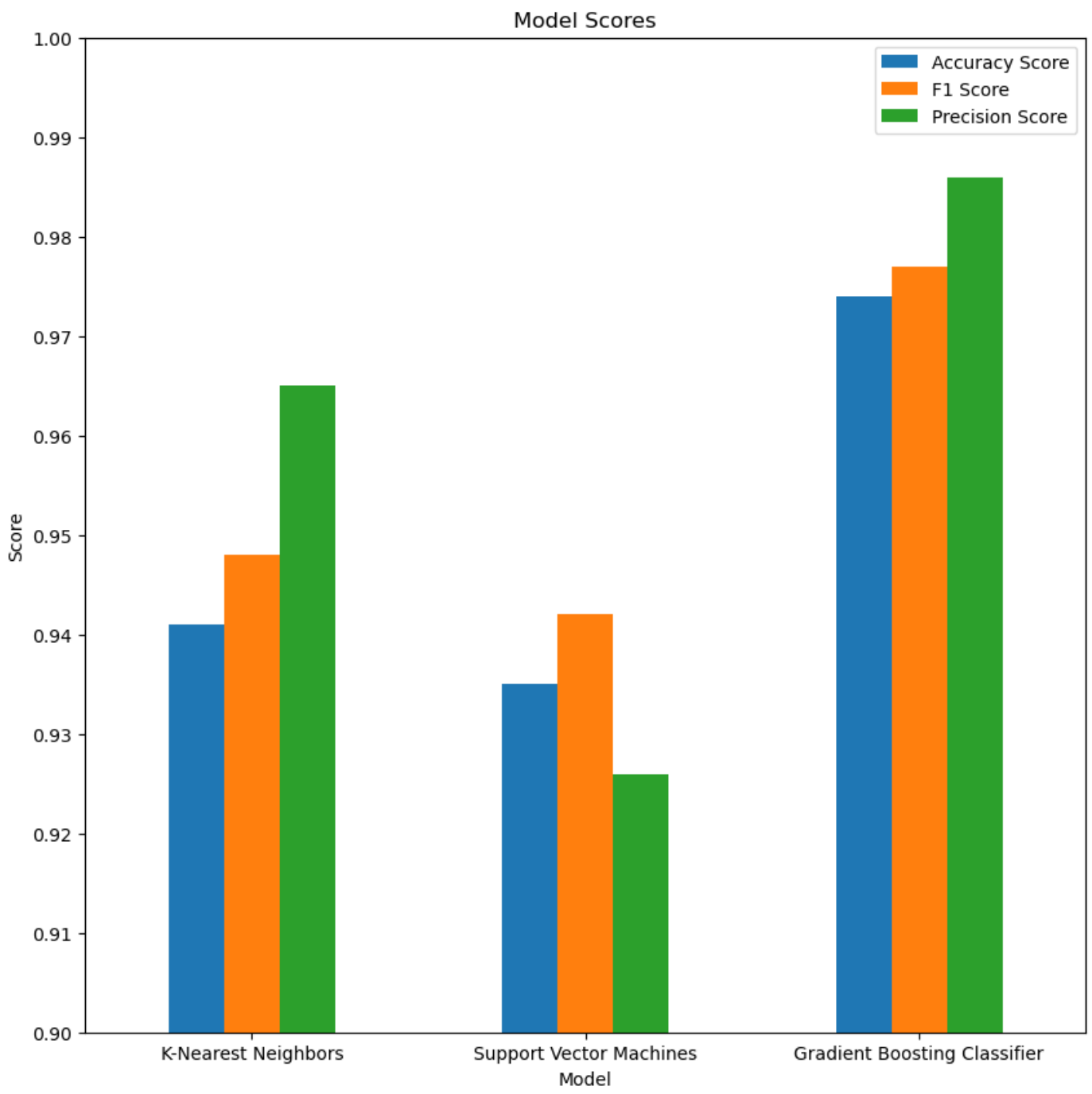
```
Gradient Boosting Classifier : Accuracy on training Data: 0.989
Gradient Boosting Classifier : Accuracy on test Data: 0.974
```

Gradient Boost classifier has a greater accuracy, hence this model will be utilized afterward.

Model Evaluation

6. Evaluate the models' performance on the testing/validation set. Calculate metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

In [34]:
```
# Evaluate the model on the test set

df = pd.DataFrame({
    'Modelname': ML_Model,
    'Accuracy Score': accuracy,
    'F1 Score': f1_score,
    'Precision Score': precision
})
df.set_index('Modelname', inplace=True)

# plot the scores for each model

fig, ax = plt.subplots(figsize=(10,10))
df.plot(kind='bar', ax=ax)
ax.set_xticklabels(df.index, rotation=0)
ax.set_ylim([0.9, 1])
ax.set_yticks([0.9,0.91,0.92,0.93,0.94,0.95,0.96,0.97,0.98,0.99,1])
ax.set_xlabel('Model')
ax.set_ylabel('Score')
ax.set_title('Model Scores')
plt.show()
```



In [35]:
```
class FeatureExtraction:
    features = []
    def __init__(self,url):
        self.features = []
        self.url = url
        self.domain = ""
        self.whois_response = ""
        self.urlparse = ""
        self.response = ""
        self.soup = ""

        try:
            self.response = requests.get(url)
            self.soup = BeautifulSoup(response.text, 'html.parser')
        except:
            pass

        try:
            self.urlparse = urlparse(url)
            self.domain = self.urlparse.netloc
        except:
```

```
        pass

    try:
        self.whois_response = whois.whois(self.domain)
    except:
        pass



    self.features.append(self.UsingIp())
    self.features.append(self.longUrl())
    self.features.append(self.shortUrl())
    self.features.append(self.symbol())
    self.features.append(self.redirecting())
    self.features.append(self.prefixSuffix())
    self.features.append(self.SubDomains())
    self.features.append(self.Hppts())
    self.features.append(self.DomainRegLen())
    self.features.append(self.Favicon())


    self.features.append(self.NonStdPort())
    self.features.append(self.HTTPSDomainURL())
    self.features.append(self.RequestURL())
    self.features.append(self.AnchorURL())
    self.features.append(self.LinksInScriptTags())
    self.features.append(self.ServerFormHandler())
    self.features.append(self.InfoEmail())
    self.features.append(self.AbnormalURL())
    self.features.append(self.WebsiteForwarding())
    self.features.append(self.StatusBarCust())

    self.features.append(self.DisableRightClick())
    self.features.append(self.UsingPopupWindow())
    self.features.append(self.IframeRedirection())
    self.features.append(self.AgeofDomain())
    self.features.append(self.DNSRecording())
    self.features.append(self.WebsiteTraffic())
    self.features.append(self.PageRank())
    self.features.append(self.GoogleIndex())
    self.features.append(self.LinksPointingToPage())
    self.features.append(self.StatsReport())


 # 1.UsingIp
def UsingIp(self):
    try:
        ipaddress.ip_address(self.url)
        return -1
    except:
        return 1

# 2.longUrl
def longUrl(self):
    if len(self.url) < 54:
        return 1
    if len(self.url) >= 54 and len(self.url) <= 75:
        return 0
    return -1

# 3.shortUrl
def shortUrl(self):
    match = re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyu
                'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl
                'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.
```

```
                    'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.
                    'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.l
                    'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cu
                    'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.n
        if match:
            return -1
        return 1

    # 4.Symbol@
    def symbol(self):
        if re.findall("@",self.url):
            return -1
        return 1

    # 5.Redirecting//
    def redirecting(self):
        if self.url.rfind('//')>6:
            return -1
        return 1

    # 6.prefixSuffix
    def prefixSuffix(self):
        try:
            match = re.findall('\-', self.domain)
            if match:
                return -1
            return 1
        except:
            return -1

    # 7.SubDomains
    def SubDomains(self):
        dot_count = len(re.findall("\.", self.url))
        if dot_count == 1:
            return 1
        elif dot_count == 2:
            return 0
        return -1

    # 8.HTTPS
    def Hppts(self):
        try:
            https = self.urlparse.scheme
            if 'https' in https:
                return 1
            return -1
        except:
            return 1

    # 9.DomainRegLen
    def DomainRegLen(self):
        try:
            expiration_date = self.whois_response.expiration_date
            creation_date = self.whois_response.creation_date
            try:
                if(len(expiration_date)):
                    expiration_date = expiration_date[0]
            except:
                pass
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            age = (expiration_date.year-creation_date.year)*12+ (expiration_date.month-c
```

```
            if age >=12:
                return 1
            return -1
        except:
            return -1

    # 10. Favicon
    def Favicon(self):
        try:
            for head in self.soup.find_all('head'):
                for head.link in self.soup.find_all('link', href=True):
                    dots = [x.start(0) for x in re.finditer('\.', head.link['href'])]
                    if self.url in head.link['href'] or len(dots) == 1 or domain in head
                        return 1
            return -1
        except:
            return -1

    # 11. NonStdPort
    def NonStdPort(self):
        try:
            port = self.domain.split(":")
            if len(port)>1:
                return -1
            return 1
        except:
            return -1

    # 12. HTTPSDomainURL
    def HTTPSDomainURL(self):
        try:
            if 'https' in self.domain:
                return -1
            return 1
        except:
            return -1

    # 13. RequestURL
    def RequestURL(self):
        try:
            for img in self.soup.find_all('img', src=True):
                dots = [x.start(0) for x in re.finditer('\.', img['src'])]
                if self.url in img['src'] or self.domain in img['src'] or len(dots) == 1
                    success = success + 1
                i = i+1

            for audio in self.soup.find_all('audio', src=True):
                dots = [x.start(0) for x in re.finditer('\.', audio['src'])]
                if self.url in audio['src'] or self.domain in audio['src'] or len(dots)
                    success = success + 1
                i = i+1

            for embed in self.soup.find_all('embed', src=True):
                dots = [x.start(0) for x in re.finditer('\.', embed['src'])]
                if self.url in embed['src'] or self.domain in embed['src'] or len(dots)
                    success = success + 1
                i = i+1

            for iframe in self.soup.find_all('iframe', src=True):
                dots = [x.start(0) for x in re.finditer('\.', iframe['src'])]
                if self.url in iframe['src'] or self.domain in iframe['src'] or len(dots
                    success = success + 1
                i = i+1

            try:
                percentage = success/float(i) * 100
```

```
            if percentage < 22.0:
                return 1
            elif((percentage >= 22.0) and (percentage < 61.0)):
                return 0
            else:
                return -1
        except:
            return 0
    except:
        return -1

# 14. AnchorURL
def AnchorURL(self):
    try:
        i,unsafe = 0,0
        for a in self.soup.find_all('a', href=True):
            if "#" in a['href'] or "javascript" in a['href'].lower() or "mailto" in
                unsafe = unsafe + 1
            i = i + 1

        try:
            percentage = unsafe / float(i) * 100
            if percentage < 31.0:
                return 1
            elif ((percentage >= 31.0) and (percentage < 67.0)):
                return 0
            else:
                return -1
        except:
            return -1

    except:
        return -1

# 15. LinksInScriptTags
def LinksInScriptTags(self):
    try:
        i,success = 0,0

        for link in self.soup.find_all('link', href=True):
            dots = [x.start(0) for x in re.finditer('\.', link['href'])]
            if self.url in link['href'] or self.domain in link['href'] or len(dots)
                success = success + 1
            i = i+1

        for script in self.soup.find_all('script', src=True):
            dots = [x.start(0) for x in re.finditer('\.', script['src'])]
            if self.url in script['src'] or self.domain in script['src'] or len(dots
                success = success + 1
            i = i+1

        try:
            percentage = success / float(i) * 100
            if percentage < 17.0:
                return 1
            elif((percentage >= 17.0) and (percentage < 81.0)):
                return 0
            else:
                return -1
        except:
            return 0
    except:
        return -1

# 16. ServerFormHandler
def ServerFormHandler(self):
```

```
        try:
            if len(self.soup.find_all('form', action=True))==0:
                return 1
            else :
                for form in self.soup.find_all('form', action=True):
                    if form['action'] == "" or form['action'] == "about:blank":
                        return -1
                    elif self.url not in form['action'] and self.domain not in form['act
                        return 0
                    else:
                        return 1
        except:
            return -1

    # 17. InfoEmail
    def InfoEmail(self):
        try:
            if re.findall(r"[mail\(\)|mailto:?]", self.soap):
                return -1
            else:
                return 1
        except:
            return -1

    # 18. AbnormalURL
    def AbnormalURL(self):
        try:
            if self.response.text == self.whois_response:
                return 1
            else:
                return -1
        except:
            return -1

    # 19. WebsiteForwarding
    def WebsiteForwarding(self):
        try:
            if len(self.response.history) <= 1:
                return 1
            elif len(self.response.history) <= 4:
                return 0
            else:
                return -1
        except:
             return -1

    # 20. StatusBarCust
    def StatusBarCust(self):
        try:
            if re.findall("<script>.+onmouseover.+</script>", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 21. DisableRightClick
    def DisableRightClick(self):
        try:
            if re.findall(r"event.button ?== ?2", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1
```

```
    # 22. UsingPopupWindow
    def UsingPopupWindow(self):
        try:
            if re.findall(r"alert\(", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 23. IframeRedirection
    def IframeRedirection(self):
        try:
            if re.findall(r"[<iframe>|<frameBorder>]", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 24. AgeofDomain
    def AgeofDomain(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 25. DNSRecording
    def DNSRecording(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 26. WebsiteTraffic
    def WebsiteTraffic(self):
        try:
            rank = BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=
            if (int(rank) < 100000):
                return 1
            return 0
        except :
            return -1
```

```
    # 27. PageRank
    def PageRank(self):
        try:
            prank_checker_response = requests.post("https://www.checkpagerank.net/index.

            global_rank = int(re.findall(r"Global Rank: ([0-9]+)", rank_checker_response
            if global_rank > 0 and global_rank < 100000:
                return 1
            return -1
        except:
            return -1


    # 28. GoogleIndex
    def GoogleIndex(self):
        try:
            site = search(self.url, 5)
            if site:
                return 1
            else:
                return -1
        except:
            return 1

    # 29. LinksPointingToPage
    def LinksPointingToPage(self):
        try:
            number_of_links = len(re.findall(r"<a href=", self.response.text))
            if number_of_links == 0:
                return 1
            elif number_of_links <= 2:
                return 0
            else:
                return -1
        except:
            return -1

    # 30. StatsReport
    def StatsReport(self):
        try:
            url_match = re.search(
        'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\.es|sweddy\.com|my
            ip_address = socket.gethostbyname(self.domain)
            ip_match = re.search('146\.112\.61\.108|213\.174\.157\.151|121\.50\.168\.88|
                                 '107\.151\.148\.44|107\.151\.148\.107|64\.70\.19\.203|19
                                 '118\.184\.25\.86|67\.208\.74\.71|23\.253\.126\.58|104\.
                                 '216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|1
                                 '34\.196\.13\.28|103\.224\.212\.222|172\.217\.4\.225|54\
                                 '216\.38\.62\.18|104\.130\.124\.96|47\.89\.58\.141|78\.4
            if url_match:
                return -1
            elif ip_match:
                return -1
            return 1
        except:
            return 1

    def getFeaturesList(self):
        return self.features
```

In [36]:
```
gbc = GradientBoostingClassifier(max_depth=4,learning_rate=0.7)
gbc.fit(X_train,y_train)
```

Out[36]:
```
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

In [37]:
```
url="72f8123706b6.godaddysites.com/"
#can provide any URL. this URL was taken from PhishTank
obj = FeatureExtraction(url)
x = np.array(obj.getFeaturesList()).reshape(1,30)
y_pred =gbc.predict(x)[0]
if y_pred==1:
  print("We assume it is a safe website.")
else:
  print("Caution! Suspicious website discovered.")
```

```
Caution! Suspicious website discovered.
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not h
ave valid feature names, but GradientBoostingClassifier was fitted with feature names
  warnings.warn(
```

In [38]:
```
url="https://elearn.bits-pilani.ac.in/"
#can provide any URL. this URL was taken from PhishTank
obj = FeatureExtraction(url)
x = np.array(obj.getFeaturesList()).reshape(1,30)
y_pred =gbc.predict(x)[0]
if y_pred==1:
  print("We assume it is a safe website.")
else:
  print("Caution! Suspicious website discovered.")
```

```
We assume it is a safe website.
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not h
ave valid feature names, but GradientBoostingClassifier was fitted with feature names
  warnings.warn(
```

In [ ]: