```
import pandas as pd

import numpy as np

import seaborn as sns

data=pd.read_csv("dataset_website.csv")

data
```

index	having_IP	having_IP_Address	URL	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	...	popUpWidnow	Iframe	age_of_domain	DNSRecord	web_traffic	Page_Rank	Google_Index	Links_pointing_to_page	Statistical_report	Result
0	1	-1	1	1	1	-1	-1	-1	-1	-1	...	1	1	-1	-1	-1	-1	1	1	-1	-1	1
1	2	1	1	1	1	1	-1	0	1	-1	...	1	1	-1	-1	-1	-1	1	1	0	-1	1
2	3	1	0	1	1	1	-1	-1	-1	-1	...	1	1	1	-1	-1	-1	1	0	-1	-1	1
3	4	1	0	1	1	1	-1	-1	-1	1	...	1	1	-1	-1	-1	-1	1	1	-1	1	1
4	5	1	0	-1	1	1	-1	1	1	-1	...	-1	1	-1	-1	-1	-1	1	1	0	-1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11050	11051	1	-1	1	-1	1	1	1	1	-1	...	-1	-1	1	1	-1	1	1	1	-1	-1	1
11051	11052	-1	1	1	-1	-1	-1	1	-1	-1	...	-1	1	1	1	1	1	1	1	-1	-1	1
11052	11053	1	-1	1	1	1	-1	1	-1	-1	...	1	1	1	1	-1	1	1	0	1	-1	1
11053	11054	-1	-1	1	1	1	-1	-1	-1	1	...	-1	1	1	1	-1	1	1	1	-1	-1	1
11054	11055	-1	-1	1	1	1	-1	-1	-1	1	...	1	-1	-1	1	-1	1	-1	-1	1	-1	1

11055 rows × 32 columns

data.head()

index	having_IP	having_IP_Address	URL	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	...	popUpWidnow	Iframe	age_of_domain	DNSRecord	web_traffic	Page_Rank	Google_Index	Links_pointing_to_page	Statistical_report	Result
0	1	-1	1	1	1	-1	-1	-1	-1	-1	...	1	1	-1	-1	-1	-1	1	1	-1	-1	1
1	2	1	1	1	1	1	-1	0	1	-1	...	1	1	-1	-1	-1	0	1	1	1	-1	1
2	3	1	0	1	1	1	-1	-1	-1	-1	...	1	1	1	-1	-1	-1	1	1	0	-1	1
3	4	1	0	1	1	1	-1	-1	-1	1	...	1	1	-1	-1	-1	1	1	1	-1	-1	1
4	5	1	0	-1	1	1	-1	1	1	-1	...	-1	1	-1	-1	-1	0	1	1	1	-1	1

5 rows × 32 columns

Numerical Analysis

data.shape

(11055, 32)

data.size

353760

data.info()

RangeIndex: 11055 entries, 0 to 11054

Data columns (total 32 columns):

#	Column	Non-Null Count	Dtype
0	index	11055 non-null	int64

1 having_IPhaving_IP_Address 11055 non-null int64

2 URLURL_Length 11055 non-null int64

3 Shortining_Service 11055 non-null int64

4 having_At_Symbol 11055 non-null int64

5 double_slash_redirecting 11055 non-null int64

6 Prefix_Suffix 11055 non-null int64

7 having_Sub_Domain 11055 non-null int64

8 SSLfinal_State 11055 non-null int64

9 Domain_registeration_length 11055 non-null int64

10 Favicon 11055 non-null int64

11 port 11055 non-null int64

12 HTTPS_token 11055 non-null int64

13 Request_URL 11055 non-null int64

14 URL_of_Anchor 11055 non-null int64

15 Links_in_tags 11055 non-null int64

16 SFH 11055 non-null int64

17 Submitting_to_email 11055 non-null int64

18 Abnormal_URL 11055 non-null int64

19 Redirect 11055 non-null int64

20 on_mouseover 11055 non-null int64

21 RightClick 11055 non-null int64

22 popUpWidnow 11055 non-null int64

23 Iframe 11055 non-null int64

24 age_of_domain 11055 non-null int64

25 DNSRecord 11055 non-null int64

26 web_traffic 11055 non-null int64

27 Page_Rank 11055 non-null int64

28 Google_Index 11055 non-null int64

29 Links_pointing_to_page 11055 non-null int64

30 Statistical_report 11055 non-null int64

31 Result 11055 non-null int64

dtypes: int64(32)

memory usage: 2.7 MB

data.describe()

index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	...	popUpWidnow	Iframe	age_of_domain	DNSRecord	web_traffic	Page_Rank	Google_Index	Links_pointing_to_page	Statistical_report	Result
count	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	...	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000
mean	5528.000000	0.313795	-0.633198	0.738761	0.700588	0.741474	-0.734962	0.063953	0.250927	-0.336771	...	0.613388	0.816915	0.061239	0.377114	0.287291	-0.483673	0.721574	0.344007	0.719584	0.113885
std	3191.447947	0.949534	0.766095	0.673998	0.713598	0.671011	0.678139	0.817518	0.911892	0.941629	...	0.789818	0.576784	0.998168	0.926209	0.827733	0.875289	0.692369	0.569944	0.694437	0.993539
min	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	...	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
25%	2764.500000	-1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	...	1.000000	1.000000	-1.000000	-1.000000	0.000000	-1.000000	1.000000	0.000000	1.000000	-1.000000
50%	5528.000000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	0.000000	1.000000	-1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	-1.000000	1.000000	0.000000	1.000000	1.000000

75%	8291.500000	1.000000	-1.000000	1.000000	1.000000	1.000000
	-1.000000	1.000000	1.000000	1.000000	...	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000			
max	11055.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000	1.000000	...	1.000000
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	1.000000	1.000000	1.000000			

8 rows × 32 columns

Handling Null Values

Checking for Null values in a dataset and handling if any

```
data.isnull().any()
```

```
index                        False
having_IPhaving_IP_Address    False
URLURL_Length                False
Shortining_Service           False
having_At_Symbol             False
double_slash_redirecting     False
Prefix_Suffix                False
having_Sub_Domain            False
SSLfinal_State               False
Domain_registeration_length   False
Favicon                      False
port                         False
HTTPS_token                  False
Request_URL                  False
URL_of_Anchor                False
Links_in_tags                False
SFH                          False
```

Submitting_to_email False

Abnormal_URL False

Redirect False

on_mouseover False

RightClick False

popUpWidnow False

Iframe False

age_of_domain False

DNSRecord False

web_traffic False

Page_Rank False

Google_Index False

Links_pointing_to_page False

Statistical_report False

Result False

dtype: bool

data.isnull().sum()

index 0

having_IPhaving_IP_Address 0

URLURL_Length 0

Shortining_Service 0

having_At_Symbol 0

double_slash_redirecting 0

Prefix_Suffix 0

having_Sub_Domain 0

SSLfinal_State 0

Domain_registeration_length 0

```
Favicon                 0
port                    0
HTTPS_token             0
Request_URL             0
URL_of_Anchor           0
Links_in_tags           0
SFH                     0
Submitting_to_email     0
Abnormal_URL            0
Redirect                0
on_mouseover            0
RightClick              0
popUpWidnow             0
Iframe                  0
age_of_domain           0
DNSRecord               0
web_traffic             0
Page_Rank               0
Google_Index            0
Links_pointing_to_page  0
Statistical_report      0
Result                  0
dtype: int64
```

Splitting The Data

Splitting data into independent and dependent variables

```
x=data.iloc[:,1:31].values
y=data.iloc[:,-1].values
```

```
print(x)

[[-1  1  1 ...  1  1 -1]

 [ 1  1  1 ...  1  1  1]

 [ 1  0  1 ...  1  0 -1]

 ...

 [ 1 -1  1 ...  1  0  1]

 [-1 -1  1 ...  1  1  1]

 [-1 -1  1 ... -1  1 -1]]

print(y)

[-1 -1 -1 ... -1 -1 -1]

splitting data into train and test

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_state=0)

MODEL BUILDING

from sklearn.metrics import accuracy_score, classification_report

Decision Tree

from sklearn.tree import DecisionTreeClassifier


dt=DecisionTreeClassifier()


dt.fit(x_train,y_train)

prediction_dt = dt.predict(x_test)


accuracy_dt = accuracy_score(y_test,prediction_dt)*100

scores_dict = {}

print('Accuracy score : ',accuracy_dt)

scores_dict['DecisionTreeClassifier'] = accuracy_dt
```

```
print(classification_report(y_test,prediction_dt))
```

```
Accuracy score :  96.24604251469923
```

```
              precision    recall  f1-score   support

          -1       0.97      0.95      0.96      1014
           1       0.96      0.97      0.97      1197

    accuracy                           0.96      2211
   macro avg       0.96      0.96      0.96      2211
weighted avg       0.96      0.96      0.96      2211
```

```
dt.feature_importances_
```

```
array([0.00746211, 0.00904331, 0.00231798, 0.003307  , 0.00207303,
       0.01885018, 0.03158893, 0.62671122, 0.01616683, 0.00449978,
       0.00090142, 0.00443275, 0.00994937, 0.10832097, 0.03308501,
       0.00978014, 0.00629146, 0.00267705, 0.00464797, 0.00274685,
       0.00153912, 0.00213057, 0.00164436, 0.01376924, 0.00823433,
       0.02800361, 0.0053712 , 0.01051683, 0.01970854, 0.00422883])
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

```
LogisticRegression()
```

```
y_pred1=lr.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score(y_test,y_pred1)*100
log_reg
```

91.67797376752601

```
scores_dict['LogisticRegression'] = log_reg

algo_name = list(scores_dict.keys())

accuracy_list = list(scores_dict.values())


sns.set(rc={'figure.figsize':(12.4,6.5)})

with sns.color_palette('muted'):

    sns.barplot(x=algo_name,y=accuracy_list)
```