

Importing Libraries & Dataset

```
In [12]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

```
In [14]: data=pd.read_csv("D:/Collection Of Dataset/dataset_website.csv")
```

```
In [15]: data
```

```
Out[15]:
```

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Doma
0	1	-1	1	1	1	-1	-1	-1	-1	
1	2	1	1	1	1	1	-1	0	1	
2	3	1	0	1	1	1	-1	-1	-1	
3	4	1	0	1	1	1	-1	-1	-1	
4	5	1	0	-1	1	1	-1	1	1	
...
11050	11051	1	-1	1	-1	1	1	1	1	
11051	11052	-1	1	1	-1	-1	-1	1	-1	
11052	11053	1	-1	1	1	1	-1	1	-1	
11053	11054	-1	-1	1	1	1	-1	-1	-1	
11054	11055	-1	-1	1	1	1	-1	-1	-1	

```
Out[15]:
```

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Doma
0	1	-1	1	1	1	-1	-1	-1	-1	
1	2	1	1	1	1	1	-1	0	1	
2	3	1	0	1	1	1	-1	-1	-1	
3	4	1	0	1	1	1	-1	-1	-1	
4	5	1	0	-1	1	1	-1	1	1	
...
11050	11051	1	-1	1	-1	1	1	1	1	
11051	11052	-1	1	1	-1	-1	-1	1	-1	
11052	11053	1	-1	1	1	1	-1	1	-1	
11053	11054	-1	-1	1	1	1	-1	-1	-1	
11054	11055	-1	-1	1	1	1	-1	-1	-1	

11055 rows × 32 columns



```
In [16]: data.head()
```

```
Out[16]:
```

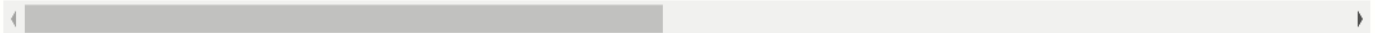
	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_re
0	1	-1	1	1	1	-1	-1	-1	-1	
1	2	1	1	1	1	1	-1	0	1	
2	3	1	0	1	1	1	-1	-1	-1	
3	4	1	0	1	1	1	-1	-1	-1	

In [20]: `data.describe()`

Out[20]:

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State
count	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
std	3191.447947	0.949534	0.766095	0.673998	0.713598	0.671011	0.678139	0.817518	0.911892
min	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
50%	5528.000000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	0.000000	1.000000
75%	8291.500000	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

8 rows × 10 columns



In [21]: `data.isnull().any()`

Out[21]:

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_registration_length	False
Favicon	False

```

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

```

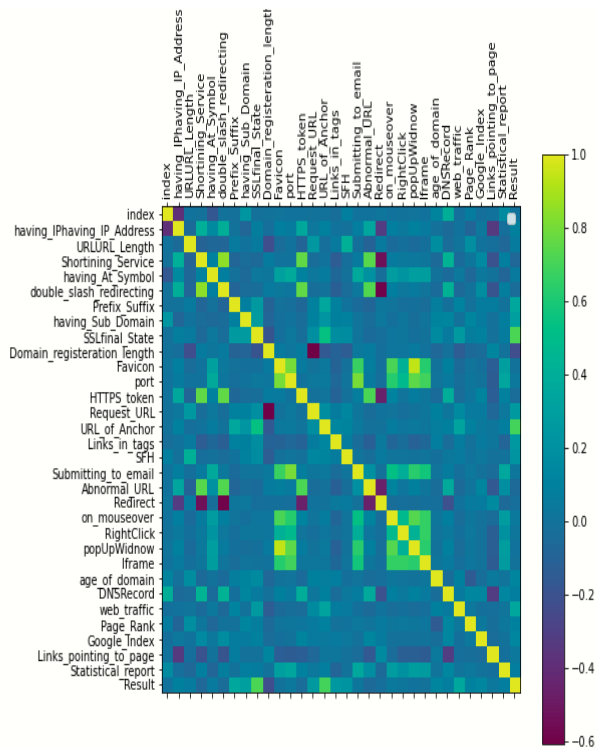
Data Visualization

```

In [25]: def plot_corr(df,size=8):
          corr=df.corr()
          fig,ax=plt.subplots(figsize=(size,size))
          ax.legend()
          cax=ax.matshow(corr)
          fig.colorbar(cax)
          plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical')
          plt.yticks(range(len(corr.columns)), corr.columns)
          plot_corr(data)

```

No handles with labels found to put in legend.



```
In [26]: with sns.color_palette('muted'):
sns.heatmap(v_data['Parul+'])
```