

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, accuracy_score
```

## Reading the dataset

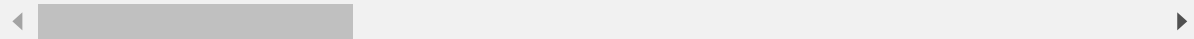
In [8]:

```
# Importing Dataset
ds = pd.read_csv("data_website.csv")
ds.head()
```

Out[8]:

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol
0	1	-1	1	1	1
1	2	1	1	1	1
2	3	1	0	1	1
3	4	1	0	1	1
4	5	1	0	-1	1

5 rows × 32 columns



## Handling null values

In [4]:

```
ds.info()
ds.isnull().any()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
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_registration_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
```

Out[4]:

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
port	False
HTTPS_token	False

Preprocessing math: 100%

```
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
```

## Splitting the data

In [5]:

```
# removing index column in independent dataset
x = ds.iloc[:,1:31].values
y = ds.iloc[:, -1].values
print(x,y)
```

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

In [7]:

```
# 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

In [9]:

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

Out[9]:

```
LogisticRegression()
```

In [10]:

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

Out[10]:

0.9167797376752601

In [11]:

```
import pickle
pickle.dump(lr,open('Phishing_Website.pkl','wb'))
```