

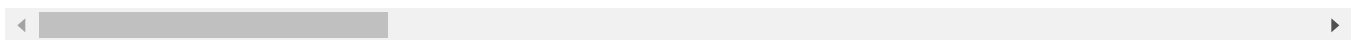
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import r2_score, accuracy_score
from sklearn.model_selection import train_test_split
```

```
df_test=pd.read_csv("Phising_Testing_Dataset.csv")
df_train=pd.read_csv("Phising_Training_Dataset.csv")
```

```
df_test.head(100)
```

	key	having_IP	URL_Length	Shortining_Service	having_At_Symbol	double_slash_re
0	21338	1	1	1	1	
1	21339	1	-1	1	1	
2	21340	1	-1	1	1	
3	21341	-1	-1	-1	1	
4	21342	1	-1	1	1	
...
95	21433	1	-1	1	1	
96	21434	1	-1	1	1	
97	21435	1	-1	1	1	
98	21436	1	-1	1	1	
99	21437	-1	-1	-1	1	

100 rows × 31 columns



```
x=df_train.drop('key', axis=1)
```

```
df_train.isnull().sum()
```

```
key          0
having_IP    0
URL_Length   0
Shortining_Service  0
having_At_Symbol  0
double_slash_redirecting  0
```

```

Prefix_Suffix      0
having_Sub_Domain  0
SSLfinal_State     0
Domain_registration_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

```

```

x=df_train.drop('Result',axis=1)
y=df_train['Result']

```

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.5,random_state=10)
```

```
print(xtrain.shape,xtest.shape,ytrain.shape,ytest.shape)
```

```
(4477, 31) (4478, 31) (4477,) (4478,)
```

```

from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(bootstrap=True,n_estimators=20, criterion="entropy")
classifier.fit(xtrain, ytrain)
pred=classifier.predict(xtest)

```

```

print("accuracy score on testing:",accuracy_score(ytest,pred)*100)
print('accuracy score on training:-',(accuracy_score(classifier.predict(xtrain),ytrain))*100)

```

```

accuracy score on testing: 96.13666815542653
accuracy score on training:- 99.97766361402725

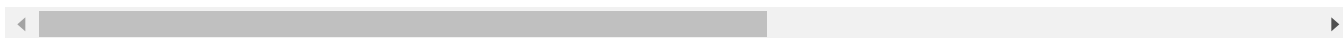
```

```

prediction=classifier.predict(df_test)
prediction= pd.DataFrame(prediction,columns=['Result'])

```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:444: UserWarning: X has feature
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```



```
sub=pd.concat([df_test.key,prediction],axis=1)
```

```
sub.to_csv(f"Submission.csv")
```

✓ 0s completed at 9:57 AM

