

```
#data preprocessing
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv("/content/detect_phishing_website.csv")
data.head()
```

	id	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double
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	

```
data.drop(["id"],axis=1,inplace=True)
data.columns
```

```
Index(['having_IP_Address', 'URL_Length', 'Shortining_Service',
       'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
       'having_Sub_Domain', 'SSLfinal_State', 'Domain_registration_length',
       'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
       'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL',
       'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe',
       'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
       'Google_Index', 'Links_pointing_to_page', 'Statistical_report',
       'Result'],
      dtype='object')
```

```
data.shape
```

```
(11055, 31)
```

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

```
False
```

```
#1
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('having_IP_Address',axis=1)
y=data['having_IP_Address']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 84.24920850293984
Test set accuracy : 84.84848484848484
```

	precision	recall	f1-score	support
-1	0.82	0.72	0.76	762
1	0.86	0.92	0.89	1449
accuracy			0.85	2211
macro avg	0.84	0.82	0.83	2211
weighted avg	0.85	0.85	0.85	2211

#2

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('URL_Length',axis=1)
y=data['URL_Length']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 87.46042514699232
Test set accuracy : 86.47670737222975
```

	precision	recall	f1-score	support
-1	0.90	0.95	0.92	1800
0	0.19	0.10	0.13	30
1	0.70	0.54	0.61	381
accuracy			0.86	2211
macro avg	0.60	0.53	0.55	2211
weighted avg	0.85	0.86	0.86	2211

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWarning: STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

#3

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Shortining_Service',axis=1)
y=data['Shortining_Service']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 97.87426503844414
Test set accuracy : 97.64812302125735
```

	precision	recall	f1-score	support
-1	0.96	0.86	0.90	287
1	0.98	0.99	0.99	1924
accuracy			0.98	2211
macro avg	0.97	0.93	0.95	2211
weighted avg	0.98	0.98	0.98	2211

#4

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('having_At_Symbol',axis=1)
y=data['having_At_Symbol']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 90.85255540479422
Test set accuracy : 90.41157847127997
```

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

-1	0.81	0.49	0.61	342
1	0.91	0.98	0.95	1869
accuracy			0.90	2211
macro avg	0.86	0.74	0.78	2211
weighted avg	0.90	0.90	0.89	2211

#5

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('double_slash_redirecting',axis=1)
y=data['double_slash_redirecting']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 97.98733604703754
Test set accuracy : 97.55766621438264

```

	precision	recall	f1-score	support
-1	0.93	0.87	0.90	278
1	0.98	0.99	0.99	1933
accuracy			0.98	2211
macro avg	0.95	0.93	0.94	2211
weighted avg	0.98	0.98	0.98	2211

#6

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Prefix_Suffix',axis=1)
y=data['Prefix_Suffix']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 87.2003618272275
Test set accuracy : 88.78335594753504
      precision    recall  f1-score   support

-1         0.89         0.99         0.94        1930
 1         0.71         0.20         0.31         281

 accuracy
macro avg         0.80         0.59         0.63        2211
weighted avg         0.87         0.89         0.86        2211

```

#7

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('having_Sub_Domain',axis=1)
y=data['having_Sub_Domain']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 53.38082315694256
Test set accuracy : 52.78154681139756
      precision    recall  f1-score   support

-1         0.40         0.35         0.37         650
 0         0.50         0.45         0.48         712
 1         0.62         0.73         0.67         849

 accuracy
macro avg         0.51         0.51         0.51        2211
weighted avg         0.52         0.53         0.52        2211

```

#8

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('SSLfinal_State',axis=1)
y=data['SSLfinal_State']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)

```

```
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 81.0719131614654
Test set accuracy : 80.1899592944369
```

	precision	recall	f1-score	support
-1	0.68	0.78	0.73	710
0	0.62	0.49	0.55	247
1	0.92	0.88	0.89	1254
accuracy			0.80	2211
macro avg	0.74	0.72	0.72	2211
weighted avg	0.81	0.80	0.80	2211

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWarning: STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

#9

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Domain_registration_length',axis=1)
y=data['Domain_registration_length']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 81.00407055630936
Test set accuracy : 80.86838534599728
```

	precision	recall	f1-score	support
-1	0.88	0.82	0.85	1436
1	0.70	0.79	0.74	775
accuracy			0.81	2211
macro avg	0.79	0.80	0.80	2211

weighted avg	0.82	0.81	0.81	2211
--------------	------	------	------	------

#10

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Favicon',axis=1)
y=data['Favicon']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 98.81275440976933
Test set accuracy : 98.95974672094076

```

	precision	recall	f1-score	support
-1	0.96	0.98	0.97	426
1	1.00	0.99	0.99	1785
accuracy			0.99	2211
macro avg	0.98	0.99	0.98	2211
weighted avg	0.99	0.99	0.99	2211

#11

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('port',axis=1)
y=data['port']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 98.79014020805066
Test set accuracy : 98.68837630031659

```

	precision	recall	f1-score	support
-1	0.95	0.96	0.95	295
1	0.99	0.99	0.99	1916

accuracy			0.99	2211
macro avg	0.97	0.97	0.97	2211
weighted avg	0.99	0.99	0.99	2211

#12

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('HTTPS_token',axis=1)
y=data['HTTPS_token']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 94.54997738579827
Test set accuracy : 93.39665309814563

```

	precision	recall	f1-score	support
-1	0.89	0.69	0.78	368
1	0.94	0.98	0.96	1843

accuracy			0.93	2211
macro avg	0.91	0.84	0.87	2211
weighted avg	0.93	0.93	0.93	2211

#13

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Request_URL',axis=1)
y=data['Request_URL']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 81.54681139755766
Test set accuracy : 81.00407055630936

```


	precision	recall	f1-score	support
-1	0.84	0.69	0.76	950
1	0.80	0.90	0.84	1261
accuracy			0.81	2211
macro avg	0.82	0.80	0.80	2211
weighted avg	0.81	0.81	0.81	2211

#14

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('URL_of_Anchor',axis=1)
y=data['URL_of_Anchor']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 69.79873360470376
Test set accuracy : 70.4658525554048

```

	precision	recall	f1-score	support
-1	0.77	0.87	0.82	694
0	0.67	0.77	0.72	1043
1	0.63	0.31	0.42	474
accuracy			0.70	2211
macro avg	0.69	0.65	0.65	2211
weighted avg	0.70	0.70	0.69	2211

#15

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Links_in_tags',axis=1)
y=data['Links_in_tags']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))

```

```
print(metrics.classification_report(y_test,y_predict))
```

```

train set accuracy: 50.93848937132519
Test set accuracy : 49.43464495703302
      precision    recall  f1-score   support

-1         0.52         0.52         0.52         785
 0         0.49         0.62         0.55         869
 1         0.46         0.27         0.34         557

 accuracy                   0.49         2211
 macro avg              0.49         0.47         0.47         2211
 weighted avg           0.49         0.49         0.48         2211

```

#16

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('SFH',axis=1)
y=data['SFH']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 80.0429669832655
Test set accuracy : 80.1899592944369
      precision    recall  f1-score   support

-1         0.83         0.96         0.89         1712
 0         0.00         0.00         0.00          151
 1         0.58         0.37         0.45          348

 accuracy                   0.80         2211
 macro avg              0.47         0.44         0.45         2211
 weighted avg           0.73         0.80         0.76         2211

```

#17

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Submitting_to_email',axis=1)
y=data['Submitting_to_email']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)

```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 95.7259158751696
Test set accuracy : 93.93939393939394
```

	precision	recall	f1-score	support
-1	0.92	0.75	0.82	420
1	0.94	0.98	0.96	1791
accuracy			0.94	2211
macro avg	0.93	0.87	0.89	2211
weighted avg	0.94	0.94	0.94	2211

#18

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Abnormal_URL',axis=1)
y=data['Abnormal_URL']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 96.24604251469923
Test set accuracy : 95.61284486657621
```

	precision	recall	f1-score	support
-1	0.87	0.83	0.85	327
1	0.97	0.98	0.97	1884
accuracy			0.96	2211
macro avg	0.92	0.90	0.91	2211
weighted avg	0.96	0.96	0.96	2211

#19

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
```

```

from sklearn.model_selection import train_test_split
x=data.drop('Redirect',axis=1)
y=data['Redirect']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 93.61148801447308
Test set accuracy : 92.80868385345997

```

	precision	recall	f1-score	support
0	0.94	0.98	0.96	1951
1	0.76	0.56	0.65	260
accuracy			0.93	2211
macro avg	0.85	0.77	0.80	2211
weighted avg	0.92	0.93	0.92	2211

#20

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('on_mouseover',axis=1)
y=data['on_mouseover']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 95.60153776571687
Test set accuracy : 94.61781999095432

```

	precision	recall	f1-score	support
-1	0.76	0.79	0.78	260
1	0.97	0.97	0.97	1951
accuracy			0.95	2211
macro avg	0.87	0.88	0.87	2211
weighted avg	0.95	0.95	0.95	2211

#21

```

from sklearn.linear_model import LogisticRegression

```

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('RightClick',axis=1)
y=data['RightClick']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 96.80009045680687
Test set accuracy : 96.69832654907282

```

	precision	recall	f1-score	support
-1	0.65	0.53	0.58	96
1	0.98	0.99	0.98	2115
accuracy			0.97	2211
macro avg	0.81	0.76	0.78	2211
weighted avg	0.96	0.97	0.97	2211

#22

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('popUpWidnow',axis=1)
y=data['popUpWidnow']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 98.68837630031659
Test set accuracy : 98.41700587969244

```

	precision	recall	f1-score	support
-1	0.96	0.96	0.96	439
1	0.99	0.99	0.99	1772
accuracy			0.98	2211
macro avg	0.97	0.98	0.98	2211
weighted avg	0.98	0.98	0.98	2211

#23

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Iframe',axis=1)
y=data['Iframe']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 96.969696969697
Test set accuracy : 96.92446856625962

```

	precision	recall	f1-score	support
-1	0.86	0.77	0.82	194
1	0.98	0.99	0.98	2017
accuracy			0.97	2211
macro avg	0.92	0.88	0.90	2211
weighted avg	0.97	0.97	0.97	2211

#24

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('age_of_domain',axis=1)
y=data['age_of_domain']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 65.76209859791949
Test set accuracy : 64.49570330167344

```

	precision	recall	f1-score	support
-1	0.62	0.59	0.60	1010
1	0.67	0.69	0.68	1201

accuracy			0.64	2211
macro avg	0.64	0.64	0.64	2211
weighted avg	0.64	0.64	0.64	2211

#25

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('DNSRecord',axis=1)
y=data['DNSRecord']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 85.68521031207598
Test set accuracy : 86.34102216191768

```

	precision	recall	f1-score	support
-1	0.81	0.72	0.76	674
1	0.88	0.93	0.90	1537

accuracy			0.86	2211
macro avg	0.85	0.82	0.83	2211
weighted avg	0.86	0.86	0.86	2211

#26

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('web_traffic',axis=1)
y=data['web_traffic']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 60.10854816824966
Test set accuracy : 60.651289009497965

```

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

-1	0.52	0.38	0.44	553
0	0.47	0.35	0.40	501
1	0.67	0.82	0.74	1157
accuracy			0.61	2211
macro avg	0.55	0.52	0.53	2211
weighted avg	0.59	0.61	0.59	2211

#27

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Page_Rank',axis=1)
y=data['Page_Rank']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 75.94979647218453
Test set accuracy : 76.20985979194934

```

	precision	recall	f1-score	support
-1	0.78	0.94	0.85	1636
1	0.60	0.26	0.36	575
accuracy			0.76	2211
macro avg	0.69	0.60	0.61	2211
weighted avg	0.73	0.76	0.73	2211

#28

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Google_Index',axis=1)
y=data['Google_Index']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))

```



```
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 87.14382632293079
Test set accuracy : 86.79330619629127
```

	precision	recall	f1-score	support
-1	0.62	0.15	0.25	310
1	0.88	0.98	0.93	1901
accuracy			0.87	2211
macro avg	0.75	0.57	0.59	2211
weighted avg	0.84	0.87	0.83	2211

#29

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Links_pointing_to_page',axis=1)
y=data['Links_pointing_to_page']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))
```

```
train set accuracy: 76.24378109452736
Test set accuracy : 76.4812302125735
```

	precision	recall	f1-score	support
-1	0.29	0.12	0.17	107
0	0.80	0.82	0.81	1238
1	0.74	0.76	0.75	866
accuracy			0.76	2211
macro avg	0.61	0.57	0.58	2211
weighted avg	0.75	0.76	0.76	2211

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: Convergence
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

#30

```
from sklearn.linear_model import LogisticRegression
```

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Statistical_report',axis=1)
y=data['Statistical_report']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 90.94301221166893
Test set accuracy : 90.86386250565354

```

	precision	recall	f1-score	support
-1	0.79	0.45	0.57	300
1	0.92	0.98	0.95	1911
accuracy			0.91	2211
macro avg	0.85	0.71	0.76	2211
weighted avg	0.90	0.91	0.90	2211

#31

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.model_selection import train_test_split
x=data.drop('Result',axis=1)
y=data['Result']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
lr=LogisticRegression(random_state=0)
lr.fit(x_train,y_train)
y_predict=lr.predict(x_test)
print("train set accuracy: ",100*lr.score(x_train,y_train))
print("Test set accuracy : ",100*lr.score(x_test,y_test))
print(metrics.classification_report(y_test,y_predict))

```

```

train set accuracy: 93.19312528267753
Test set accuracy : 91.67797376752601

```

	precision	recall	f1-score	support
-1	0.92	0.89	0.91	1014
1	0.91	0.94	0.92	1197
accuracy			0.92	2211
macro avg	0.92	0.91	0.92	2211
weighted avg	0.92	0.92	0.92	2211

✓ 0s completed at 10:20 AM

● ✕