```
import pandas as pd
import numpy as np
df=pd.read csv('dataset.csv')
df
       index having IPhaving IP Address URLURL Length
Shortining_Service √
                                           - 1
                                                             1
1
            2
                                            1
1
                                                             1
1
2
            3
                                            1
                                                             0
1
3
            4
                                            1
1
4
            5
                                            1
                                                             0
- 1
. . .
                                          . . .
. . .
       11051
11050
                                            1
                                                            - 1
1
11051
       11052
                                           - 1
                                                             1
11052
                                            1
      11053
                                                            - 1
11053
                                           - 1
       11054
                                                            - 1
1
11054
                                           - 1
       11055
                                                            - 1
       having_At_Symbol double_slash_redirecting Prefix_Suffix \
0
1
                         1
                                                      1
                                                                      - 1
2
                         1
                                                      1
                                                                      - 1
3
                         1
                                                      1
                                                                      - 1
4
                         1
                                                      1
                                                                      - 1
                                                                     . . .
11050
                        - 1
                                                      1
                                                                       1
11051
                        -1
                                                     - 1
                                                                      - 1
11052
                         1
                                                      1
                                                                      - 1
                         1
                                                      1
                                                                      - 1
11053
11054
                         1
                                                      1
                                                                      - 1
       having_Sub_Domain SSLfinal_State Domain_registeration_length
. . .
0
                         - 1
                                           - 1
                                                                            - 1
                          0
                                            1
                                                                            -1
1
```

2		-1	-1			-1	
3		-1	-1			1	
4		1	1			-1	
 11050		1	1			-1	
 11051		1	-1			-1	
 11052		1	-1			-1	
11053		-1	-1			1	
11054		-1	-1			1	
			-1			•	
D D	popUpWidnow	Iframe	age_of_domain	DNSRecord	web_traffic		
Page_R	ank \ 1	1	-1	-1	-1		
-1 1	1	1	-1	-1	0		
-1 2	1	1	1	-1	1		
-1 3	1	1	-1	-1	1		
- 1 4	-1	1	-1	-1	0		
-1							
 11050	-1	-1	1	1	-1		
-1 11051	-1	1	1	1	1		
1 1 11052	1	1	1	1	1		
-1							
11053 -1	-1	1	1	1	1		
11054 -1	1	1	-1	1	-1		
	Google_Index	Links_	Links_pointing_to_page Statistical_report				
Result 0	1			1	-1	-	
1 1	1			1	1	_	
1	-				_		

2	1		0	-1	-			
1 3 1	1	-	.1	1	-			
	1		1	1				
			•					
1050	1		1	1				
1051	1	-	-1	1	_			
.1052	1		0	1	_			
.1053	1		1	1	_			
1054	-1		1	-1	_			
[11055 rows	x 32 columns]							
df.head()								
	naving IPhaving	_IP_Address URLU	JRL Length					
Shortining_ 0        1	Service \		1					
1 1 2		1	1					
		1	0					
			0					
3 4 L		1						
4 5 L		1	0		-			
	having_At_Symbol double_slash_redirecting Prefix_Suffix \ 1							
9 1 2 3 4	1 1 1		1 1 1	-1 -1 -1				
	Sub_Domain SSL	final_State Doma	nin_registerat	ion_length				
\ 0	-1	-1		-1				
1	0	1		-1				
2	-1	-1		-1				

```
4
                    1
                                     1
                                                                    -1 ...
   popUpWidnow Iframe
                         age of domain DNSRecord web traffic
Page_Rank \
                      1
0
              1
                                     -1
                                                 - 1
                                                               - 1
- 1
1
              1
                      1
                                     - 1
                                                 - 1
                                                                0
- 1
2
              1
                      1
                                      1
                                                 -1
                                                                1
- 1
3
              1
                      1
                                     -1
                                                 - 1
                                                                1
- 1
             - 1
                      1
                                     -1
                                                 - 1
                                                                0
4
- 1
   Google_Index Links_pointing_to_page Statistical_report
0
               1
                                        1
                                                             - 1
                                                                      -1
1
               1
                                        1
                                                              1
                                                                      - 1
2
               1
                                                             - 1
                                        0
                                                                      - 1
3
               1
                                        - 1
                                                              1
                                                                      - 1
4
                                        1
               1
                                                              1
                                                                      1
[5 rows x 32 columns]
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read csv('dataset.csv')
numeric features = df.select dtypes(include='number')
numeric features.hist(figsize=(12, 8))
plt.tight layout()
plt.show()
corr matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

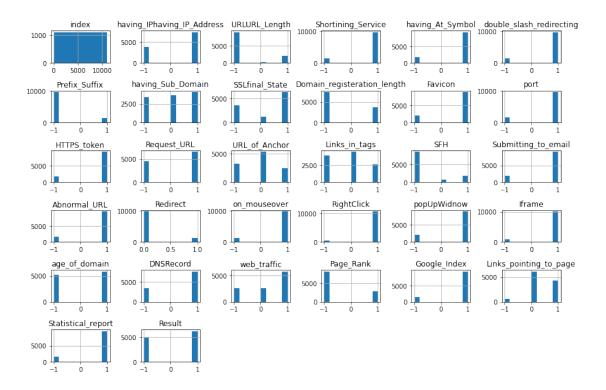
- 1

1

. . .

3

- 1



```
Correlation Heatmap
                            1.0
                                    391 .0524.1 604 00508 07 102 08 7 60136 0 3010 005 01 0 783 4.0 08 40 8 70 70 50 10 50 10 20 10 99 4
     having IPhaving IP Address
                  URLURL Length/-040505 1 .99.80.50805 0.04492/20420308925 0235241.0449104.04100 704.501.4449 040001091842 90 806057
        Shortining Service0-0063.09 1 0.1 20.0804.26050000 /20.00305630264 2.4.5 063988 70.05344040 751 60.2085068 having At Symbol-0.1071560 7551 1 08.7012508 D16 30.36.0 02.85567006 3 7-2.0 0286 20.29-200553468 30568 006 D855 3 double_slash_redirecting0-00840.0436 08 1 .0864.30868 78.0 5250 30403 372.5 08 70.0 54400.0548 062008 16.1947039
                                                                                                                                          - 0.8
             Prefix_Suffix0-9017053B150830_2831_08822509907529307399350104018450780.60392501583044017010066586500285
having_Sub_Domain 9.280860046425594883<mark>17.27.98.811704893717.29.99496688668989101808026</mark>10112-0390558268811082_3
                                                                                                                                           0.6
                    SSLfinal State0-0060710490603108662110.09016270031954180.008046202406013002860501260759601Q63
    Domain registeration length0-0012232206.00.6479.7631 1 05.40275 3 1.6. D. 0.408.98801.6240245004466301 10:96.0940200223
                    - 0.4
                   Request URO 900808.250300 802699 D.1 1050040260 1) D8067 D304 8030 2086 D204 D30400 D50505 604 6057 3525 URL of Anchor-0-00510 0000 506 80 536 20 50 103 38040 121 1) 16 D10640 D106660 204 D4 D4 D4 00 0036 20 375
                    Links_in_tags-0,0239562535305011 8 0 0941 8 10-0 0670 0677 1 0670430204 197 8 3710 07048036560604 90 40 872 5 
SFHC-08 50 047 02366604337136951 4 10-0 056706 7 0 0006 1 00 10 50 50 0 6870 0670 10 68 46 50 0 20 80 90 53 2
                                                                                                                                           - 0.2
             Submitting to email.-0.0566.01.0493708.20430.88813.357.0 07.50.88.30431 1 00200753.4 0.68.507486401.02608.404.35018
                   Abnormal_URL0-00824.1 770.2 76.07805946590 2540.0300102030 2 10 0 120249 0030 8 87 0500 3 2 16 19 06 Redirect0-0170 04 15 0.785 0168 02 01616 02 2 0000 04 10 50 12 51 03 50 42 61 8 27 00 00 0 57 16 0 990 2
                                                                                                                                          -00
                   on_mouseover0.00388408682280874311624270150100806872905312031144761610128673840160658928042
RightClickO-0053931193822026026080162444880093029203088402402441114200068814228081228013
                   popUpWidnow0.005397049377290940130251195<mark>293 0</mark> 0500405010.0046 0910 6 6 4 100 000990491701.10296-05

Iframe0.0023950191728 91 0391 00084-6: 0 00 80 0 100070 150080 156 66 6 1 00 94 702 02 00 0394 03 00 34
                                                                                                                                          - -0.2
                   age_of_domain 9.14.01.18405035.095.0957412.105060802689350997.6448010807.83380200.900680095_1_08449.05028040912
                      DNSRecord -0.4.95.D41040484B0D708050L0108856.4.00.6930B9346434.21087B899447031 04914.14.0.104076
                      web_traffic-0.01x10290904406306201005286.0395.04900416.0306596330.05x404633.00.444.0002090411.0610-03000925
                       -0.4
                    Google Index-0.016090029503718068658956040170054208-60398-60280084295-6006568-0.003502804010331_-03900513
          Statistical report 9.1690.9608608070027836680223.3409.6865070870635519069280.20.29 0.700911.40992810050111.08
                            Result 900 99 49 706 95 30 89 50. 3 7 002 8000 2 86 004 2 7 00. 26. 20 20 108 006 0 20 9 28 1329 .0503 4020 7053 50. 10. 1080 3 800 11
                                                                                                                                          - -0.6
                                            Shortining Service –
having At Symbol
double_slash_redirecting –
Prefix_Suffix –
Prefix_Suffix –
having Sub Domain –
SSLfinal State –
Domain_registeration_ength –
Favicon –
Prefix Favicon –
Request_token –
Links_in_tags –
Links_in_tags –
                                                                                                        age_of_domain -
DNSRecord -
web_traffic -
Page_Rank -
Google_Index -
                                                                                    Submitting to email
Abnormal_URL
import pandas as pd
df = pd.read csv('dataset.csv')
num samples = len(df)
print("Number of samples:", num samples)
for column in df.columns:
         unique elements = df[column].nunique()
        print("Unique elements in", column, ":", unique elements)
Number of samples: 11055
Unique elements in index: 11055
Unique elements in having IPhaving IP Address : 2
Unique elements in URLURL Length: 3
Unique elements in Shortining_Service : 2
Unique elements in having At Symbol : 2
Unique elements in double slash redirecting : 2
Unique elements in Prefix Suffix : 2
Unique elements in having Sub Domain : 3
Unique elements in SSLfinal_State : 3
```

Unique elements in Domain registeration length: 2

Unique elements in Favicon : 2

```
Unique elements in port : 2
Unique elements in HTTPS token : 2
Unique elements in Request URL : 2
Unique elements in URL of Anchor: 3
Unique elements in Links in tags : 3
Unique elements in SFH : 3
Unique elements in Submitting to email: 2
Unique elements in Abnormal URL: 2
Unique elements in Redirect : 2
Unique elements in on mouseover : 2
Unique elements in RightClick: 2
Unique elements in popUpWidnow : 2
Unique elements in Iframe : 2
Unique elements in age of domain: 2
Unique elements in DNSRecord : 2
Unique elements in web traffic : 3
Unique elements in Page Rank : 2
Unique elements in Google_Index : 2
Unique elements in Links pointing to page : 3
Unique elements in Statistical report : 2
Unique elements in Result : 2
import pandas as pd
df = pd.read csv('dataset.csv')
null values = df.isnull().sum()
if null values.any():
    print("Null values found in the following features:")
    print(null values[null values > 0])
else:
    print("No null values found in any features.")
No null values found in any features.
import pandas as pd
import numpy as np
df = pd.read csv('dataset.csv')
corr matrix = df.corr().abs()
correlation threshold = 0.7
correlated features = []
for i in range(len(corr matrix.columns)):
    for j in range(i):
        if corr matrix.iloc[i, j] > correlation threshold:
            correlated feature = corr matrix.columns[i]
            correlated features.append(correlated feature)
df filtered = df.drop(columns=correlated_features)
remaining features = df filtered.columns.tolist()
print("Remaining features after removing correlated features:")
print(remaining features)
Remaining features after removing correlated features:
['index', 'having IPhaving IP Address', 'URLURL Length',
```

```
'Shortining_Service', 'having_At_Symbol', 'Prefix_Suffix', 'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
'Favicon', 'Request_URL', 'URL_of_Anchor', 'Links_in_tags', 'SFH', 'Redirect', 'RightClick', 'Iframe', 'age_of_domain', 'DNSRecord',
'web traffic', 'Page Rank', 'Google Index', 'Links pointing to page',
'Statistical report']
import pandas as pd
from sklearn.model selection import train test split
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy score, precision_score,
recall score, f1 score
df = pd.read csv('dataset.csv')
url text = df['URLURL Length'].astype(str)
vectorizer = TfidfVectorizer()
url features = vectorizer.fit transform(url text)
X = url features.toarray()
y = df[ \( \frac{1}{2} \) Shortining_Service' ]
X train, X test, y train, y test = train test split(X, y,
test size=0.2, random state=42)
model = RandomForestClassifier()
model.fit(X_train, y train)
y pred = model.predict(X test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
import pandas as pd
from sklearn.model_selection import train test split
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc curve, auc
import matplotlib.pyplot as plt
df = pd.read csv('dataset.csv')
url text = df['URLURL Length'].astype(str)
vectorizer = TfidfVectorizer()
url features = vectorizer.fit transform(url text)
X = url features.toarray()
y = df['Shortining Service']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test size=0.2, random state=42)
model = RandomForestClassifier()
model.fit(X train, y train)
y pred prob = model.predict proba(X test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
```

```
auc score = auc(fpr, tpr)
plt.plot(fpr, tpr, label='ROC Curve (AUC = {:.2f})'.format(auc score))
plt.plot([0, 1], [0, 1], 'k--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
import pandas as pd
from sklearn.model selection import cross val score
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
df = pd.read csv('dataset.csv')
url text = df['URLURL Length'].astype(str)
vectorizer = TfidfVectorizer()
url features = vectorizer.fit transform(url text)
X = url features.toarray()
y = df['Shortining Service']
model = RandomForestClassifier()
accuracy scores = cross val score(model, X, y, cv=k,
scoring='accuracy')
print("Accuracy scores for each fold:", accuracy scores)
mean accuracy = accuracy scores.mean()
print("Mean accuracy:", mean_accuracy)
import pandas as pd
from sklearn.model selection import train test split
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy score
df = pd.read csv('dataset.csv')
selected attributes = ['URLURL Length', 'Feature1', 'Feature2']
df_selected = df[selected attributes]
url text = df selected['URLURL Length'].astype(str)
vectorizer = TfidfVectorizer()
url features = vectorizer.fit transform(url text)
X = url features.toarray()
y = df['Shortining service
X_train, X_val, y_train, y_val = train_test_split(X, y, test size=0.2,
random state=42)
models = [
    RandomForestClassifier().
1
best model = None
best accuracy = 0.0
for model in models:
```

```
model.fit(X_train, y_train)
  y_pred = model.predict(X_val)
  accuracy = accuracy_score(y_val, y_pred)
  if accuracy > best_accuracy:
      best_accuracy = accuracy
      best_model = model
print("Best Model:", best_model)
print("Accuracy on Validation Dataset:", best_accuracy)
```