Weights and Biases Assignment Expectations We are using the urlset.csv dataset where we try
identify URLs that are phished or not

- Go to the Weights and Biases website and Learn how to use weight and bias (https://wandb
  that's the website
- Then use a Decision Tree and Logistic Regression (To create a model that will help classify
  either phishing websites or not)
- However, your work should not be a Notebook
- Share the URL of your GitHub record and make sure your work is accessible
- Most importantly your work should be perfectly organized in terms of
- The Data
- Read me file
- Decision Tree
- Logistic Regression
- The result of the weights and biases Note:
- Submit the URL and the screenshots of your charts from the weights and bias
- The deadline has been pushed to the 25th of next week

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

In [ ]:
```
urlset = pd.read_csv('/content/drive/MyDrive/urlset.csv', encoding= 'ISO-
```

```
<ipython-input-4-800d359a132e>:1: DtypeWarning: Columns (1,2,3,11,12) have mi
types. Specify dtype option on import or set low_memory=False.
  urlset = pd.read_csv('/content/drive/MyDrive/urlset.csv', encoding= 'ISO-88
on_bad_lines='skip')
```

In [ ]:
```
urlset
```

Out[ ]:

	domain	ranking	mld_res	mld.ps_res	card_rem
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	10000000	1.0	0.0	18.0
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	10000000	0.0	0.0	11.0
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	10000000	0.0	0.0	14.0
3	mail.printakid.com/www.online.americanexpress....	10000000	0.0	0.0	6.0
4	thewhiskeydregs.com/wp-content/themes/widescre...	10000000	0.0	0.0	8.0
...	...	...	...	...	...
96000	xbox360.ign.com/objects/850/850402.html	339	1.0	1.0	2.0
96001	games.teamxbox.com/xbox-360/1860/Dead-Space/	63029	1.0	0.0	3.0
96002	www.gamespot.com/xbox360/action/deadspace/	753	1.0	1.0	3.0
96003	en.wikipedia.org/wiki/Dead_Space_(video_game)	6	1.0	1.0	4.0
96004	www.angelfire.com/goth/devilmaycrytonite/	2547	1.0	1.0	5.0

96005 rows × 14 columns

In [ ]:
```
urlset.head()
```

Out[ ]:

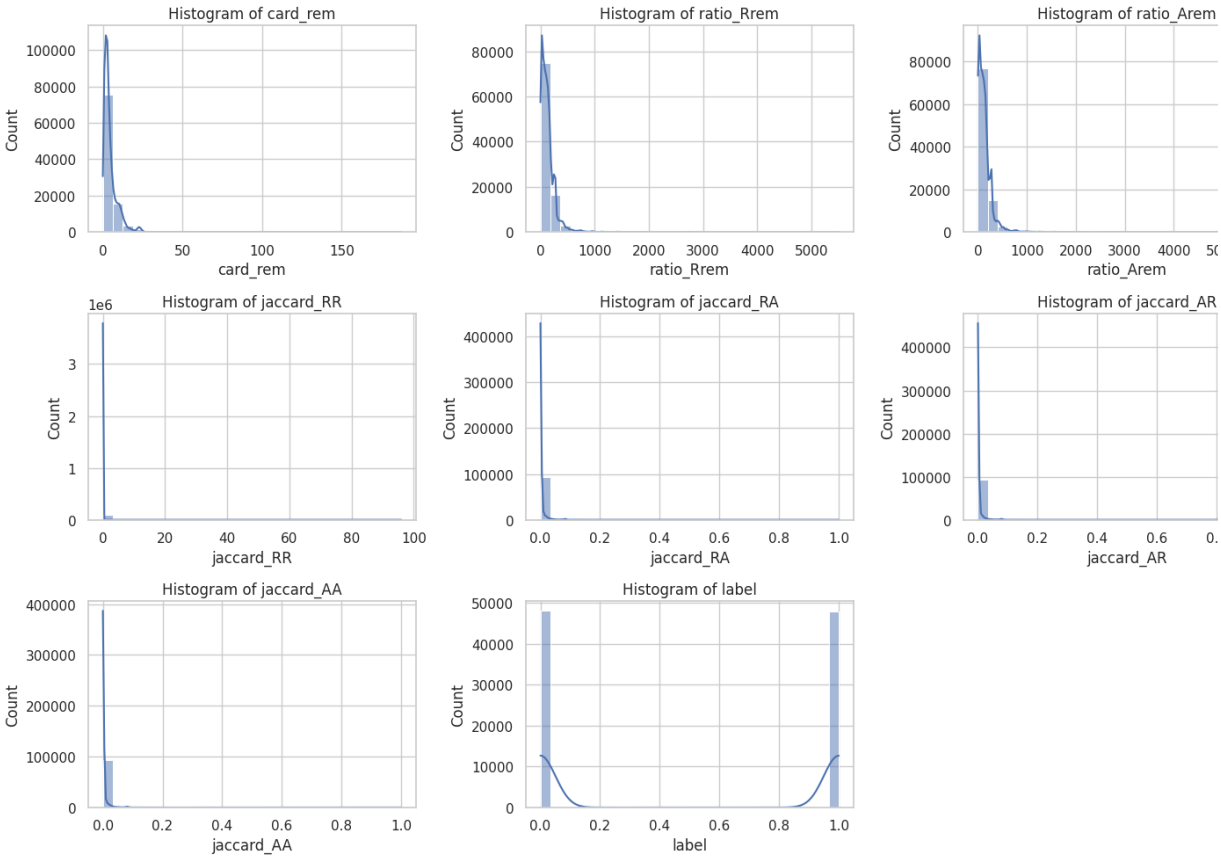
	domain	ranking	mld_res	mld.ps_res	card_rem	rati
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	10000000	1.0	0.0	18.0	107
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	10000000	0.0	0.0	11.0	150
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	10000000	0.0	0.0	14.0	73
3	mail.printakid.com/www.online.americanexpress....	10000000	0.0	0.0	6.0	562
4	thewhiskeydregs.com/wp-content/themes/widescre...	10000000	0.0	0.0	8.0	29

In [ ]:
```
urlset.info()
```

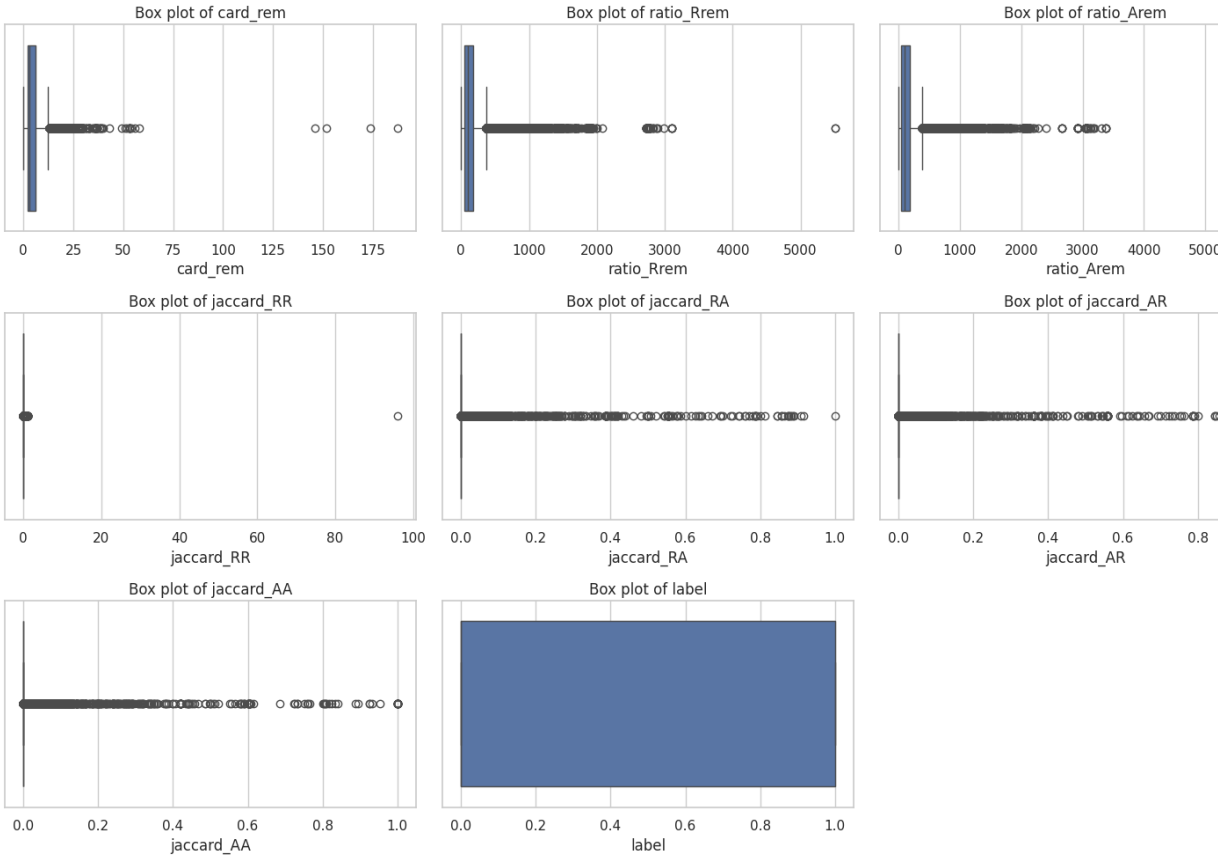
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96005 entries, 0 to 96004
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   domain         96005 non-null  object
 1   ranking        95953 non-null  object
 2   mld_res        95935 non-null  object
 3   mld.ps_res     95924 non-null  object
 4   card_rem       95923 non-null  float64
 5   ratio_Rrem     95923 non-null  float64
 6   ratio_Arem     95923 non-null  float64
 7   jaccard_RR     95922 non-null  float64
 8   jaccard_RA     95921 non-null  float64
 9   jaccard_AR     95920 non-null  float64
 10  jaccard_AA     95919 non-null  float64
 11  jaccard_ARrd   95919 non-null  object
 12  jaccard_ARrem  95917 non-null  object
 13  label          95913 non-null  float64
dtypes: float64(8), object(6)
memory usage: 10.3+ MB
```

Some visualization of the data

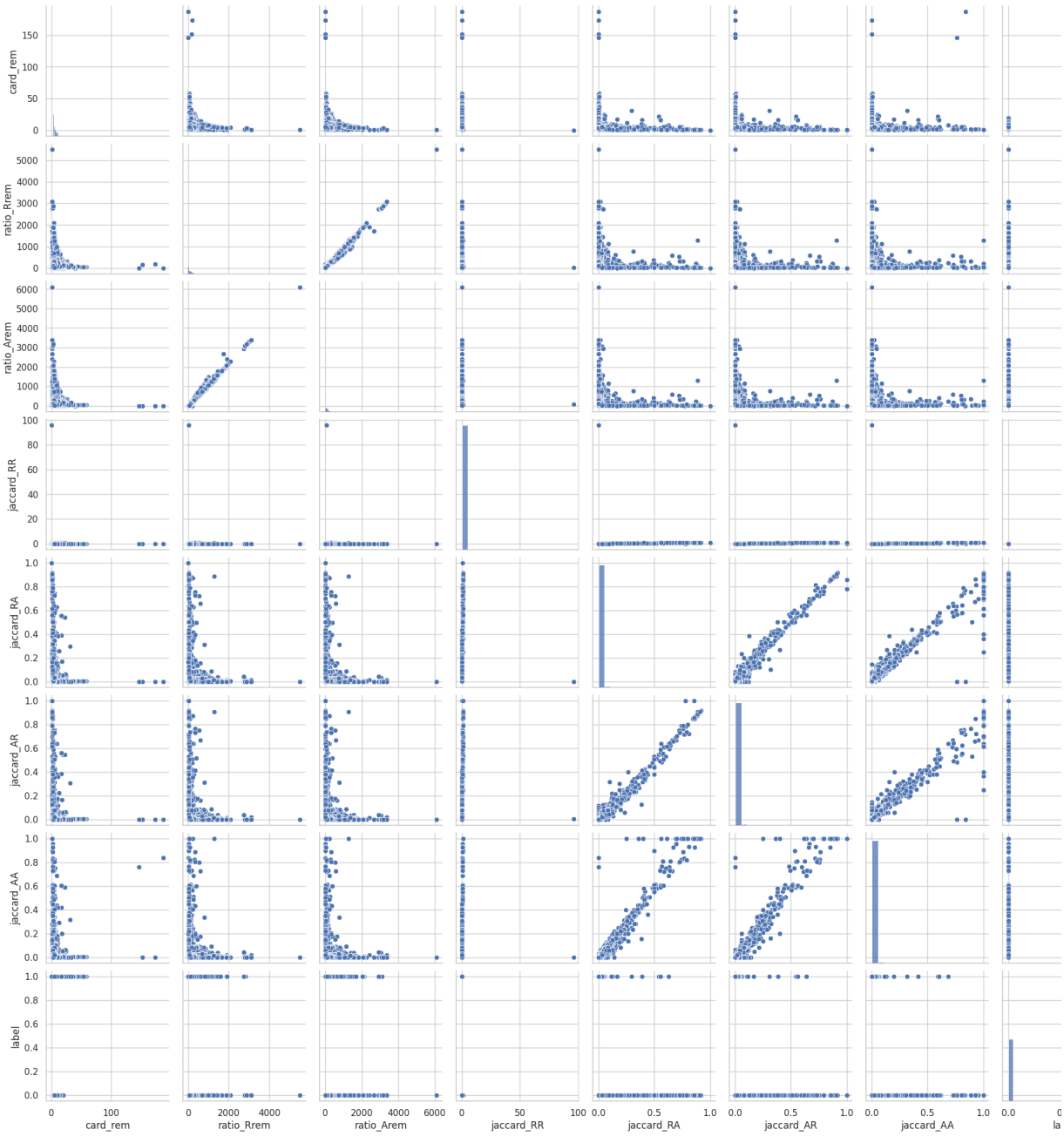
In [ ]:
```
sns.set(style='whitegrid')

# 1. Histograms for numerical features
numerical_cols = urlset.select_dtypes(include=['float64']).columns

plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_cols):
    plt.subplot(3, 3, i + 1)
    sns.histplot(urlset[col], kde=True, bins=30)
    plt.title(f'Histogram of {col}')
plt.tight_layout()
plt.show()
```



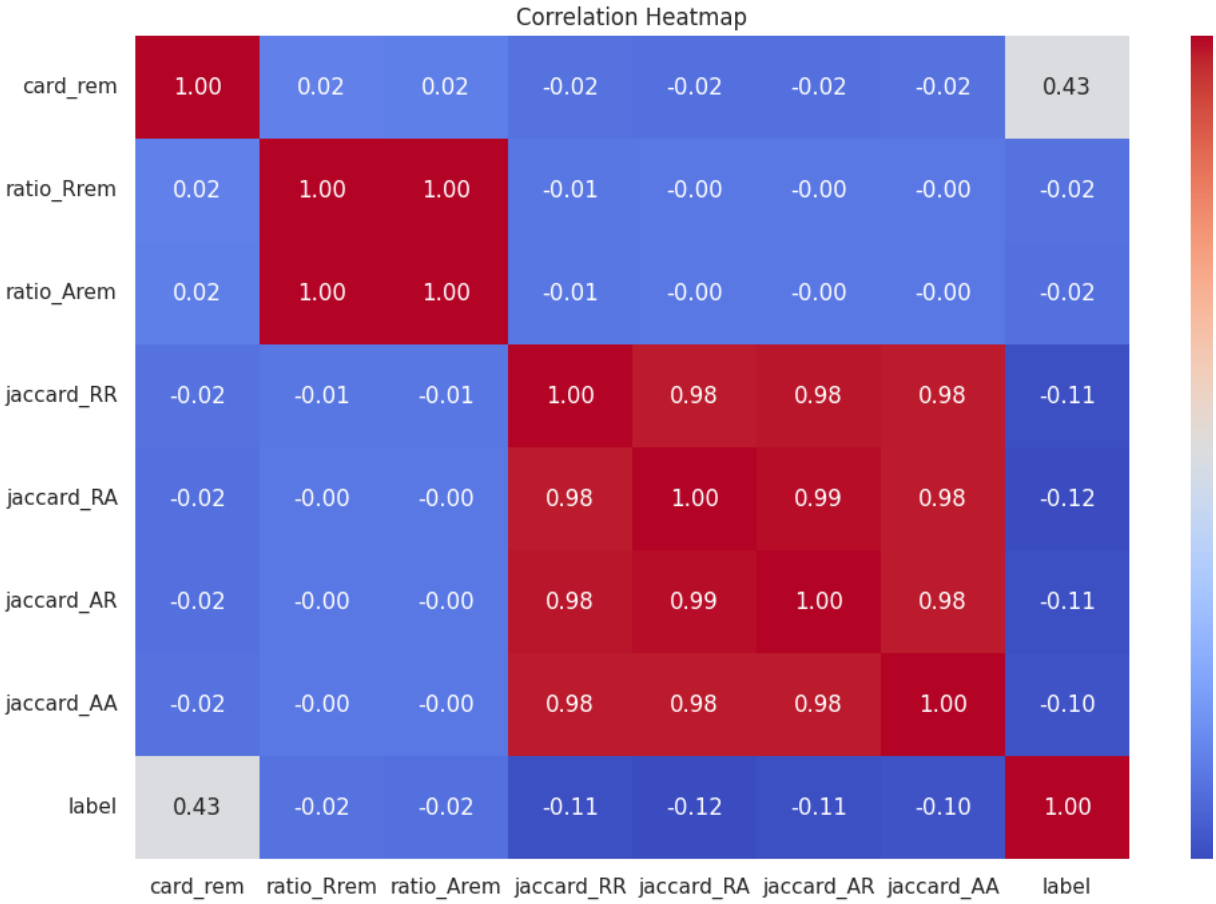
In [ ]:
```
plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_cols):
    plt.subplot(3, 3, i + 1)
    sns.boxplot(x=urlset[col])
    plt.title(f'Box plot of {col}')
plt.tight_layout()
plt.show()
```



In [ ]:
```
sns.pairplot(urlset[numerical_cols])
plt.show()
```



In [ ]:
```
plt.figure(figsize=(12, 8))
correlation_matrix = urlset[numerical_cols].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```



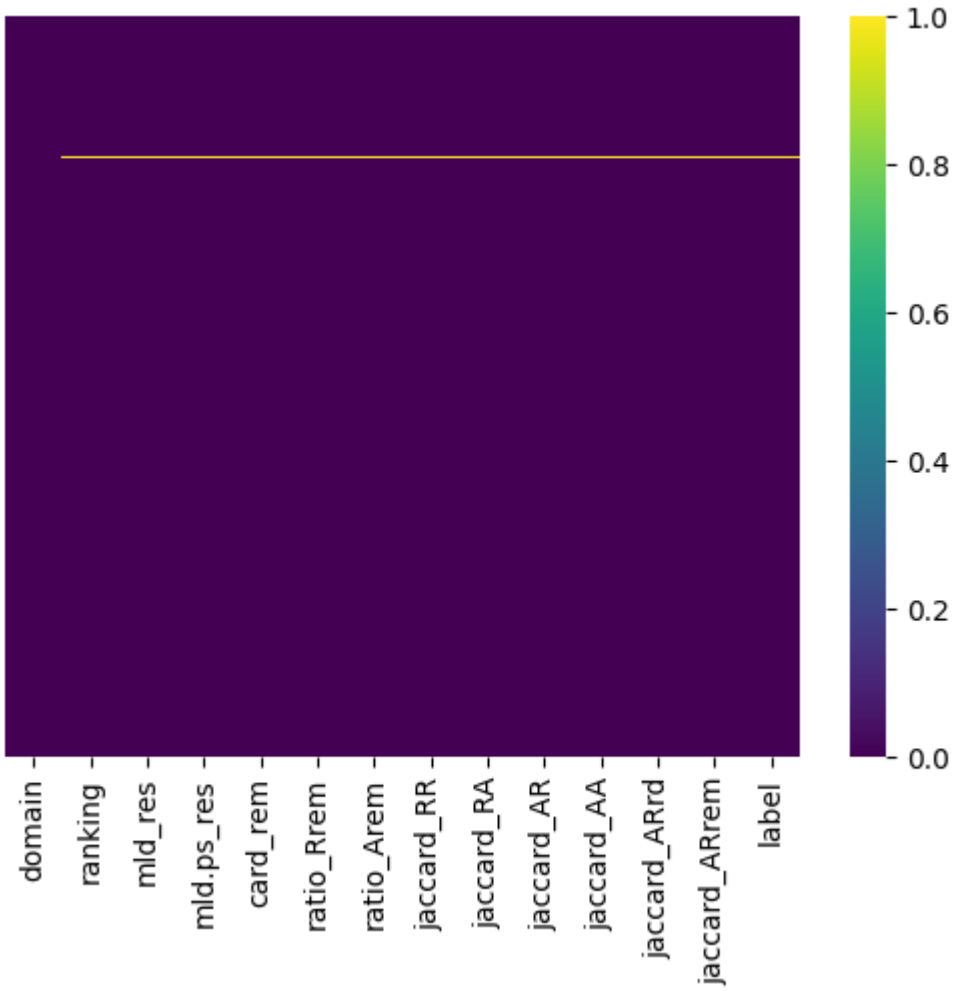
Missing data

In [ ]:
```
urlset.isnull().sum()
```

Out[ ]:
```
domain           0
ranking         52
mld_res         70
mld.ps_res      81
card_rem        82
ratio_Rrem      82
ratio_Arem      82
jaccard_RR      83
jaccard_RA      84
jaccard_AR      85
jaccard_AA      86
jaccard_ARrd    86
jaccard_ARrem   88
label           92
dtype: int64
```

In [ ]:
```
sns.heatmap(urlset.isnull(),yticklabels=False,cbar=True,cmap='viridis')
```

Out[ ]:
```
<Axes: >
```



In [ ]:
```
urlset.nunique()
```

Out[ ]:
```
domain           96003
ranking           8206
mld_res             21
mld.ps_res          10
card_rem            53
ratio_Rrem       10042
ratio_Arem       10231
jaccard_RR        5446
jaccard_RA        5628
jaccard_AR        5071
jaccard_AA        5313
jaccard_ARrd      1072
jaccard_ARrem    30752
label                2
dtype: int64
```

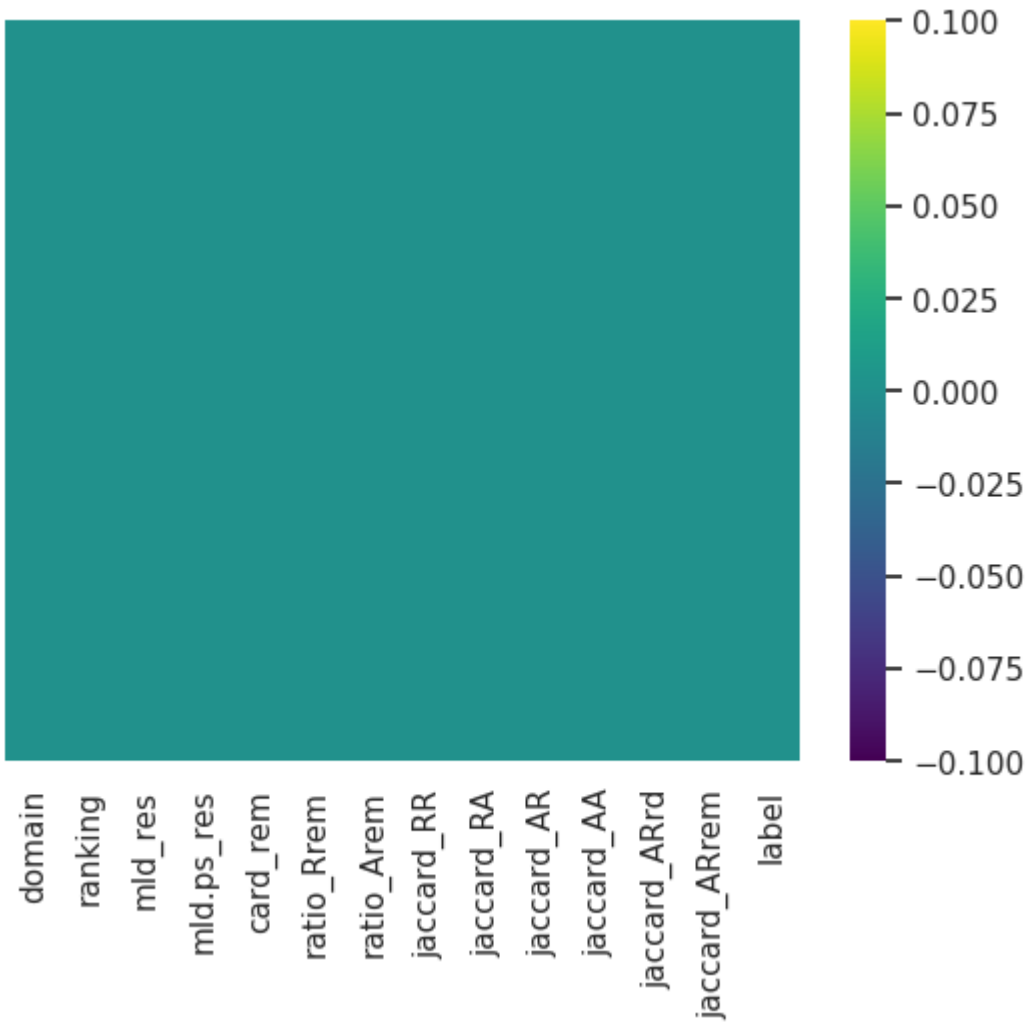
In [ ]:
```
urlset = urlset.dropna(subset=['label'])
#urlset = urlset.fillna(0)
```

In [ ]:
```
urlset.isnull().sum()
```

Out[ ]:
```
domain           0
ranking          0
mld_res          0
mld.ps_res       0
card_rem         0
ratio_Rrem       0
ratio_Arem       0
jaccard_RR       0
jaccard_RA       0
jaccard_AR       0
jaccard_AA       0
jaccard_ARrd     0
jaccard_ARrem    0
label            0
dtype: int64
```

In [ ]:
```
sns.heatmap(urlset.isnull(),yticklabels=False,cbar=True,cmap='viridis')
```

Out[ ]:
```
<Axes: >
```



Logistic regression

In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classificat
```

In [ ]:
```
features = [col for col in numerical_cols if col != 'label']
```

In [ ]:
```
X = urlset[features]
y = urlset['label']
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

In [ ]:
```
X_train.shape
```

Out[ ]:
```
(76730, 7)
```

In [ ]:
```
X_train.head()
```

Out[ ]:

	card_rem	ratio_Rrem
65406	3.0	40.666667
21429	14.0	116.571429
67697	1.0	265.000000
88510	2.0	55.500000
31344	1.0	24.000000

In [ ]:
```
X_train.columns
```

Out[ ]:
```
Index(['card_rem', 'ratio_Rrem', 'ratio_Arem', 'jaccard_RR', 'jaccard_RA',
       'jaccard_AR', 'jaccard_AA'],
      dtype='object')
```

In [ ]:
```
X_test.shape
```

Out[ ]:
```
(19183, 7)
```

In [ ]:
```
y_train.shape
```

Out[ ]:
```
(76730,)
```

In [ ]:
```
y_train.head()
```

Out[ ]:
```
65406    0.0
21429    1.0
67697    0.0
88510    0.0
31344    1.0
Name: label, dtype: float64
```

Training and predicting

In [ ]:
```
urlmodel = LogisticRegression()
```

In [ ]:
```
urlmodel
```

Out[ ]:
```
LogisticRegression
LogisticRegression()
```

In [ ]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 76730 entries, 65406 to 15795
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   card_rem    76730 non-null  float64
 1   ratio_Rrem  76730 non-null  float64
 2   ratio_Arem  76730 non-null  float64
 3   jaccard_RR  76730 non-null  float64
 4   jaccard_RA  76730 non-null  float64
 5   jaccard_AR  76730 non-null  float64
 6   jaccard_AA  76730 non-null  float64
dtypes: float64(7)
memory usage: 4.7 MB
```

In [ ]:
```
urlmodel.fit(X_train, y_train)
```

Out[ ]:
```
LogisticRegression
LogisticRegression()
```

In [ ]:
```
y_pred = urlmodel.predict(X_test)
```

In [ ]:
```
y_pred
```

Out[ ]:
```
array([1., 0., 1., ..., 1., 0., 0.])
```

Evaluation

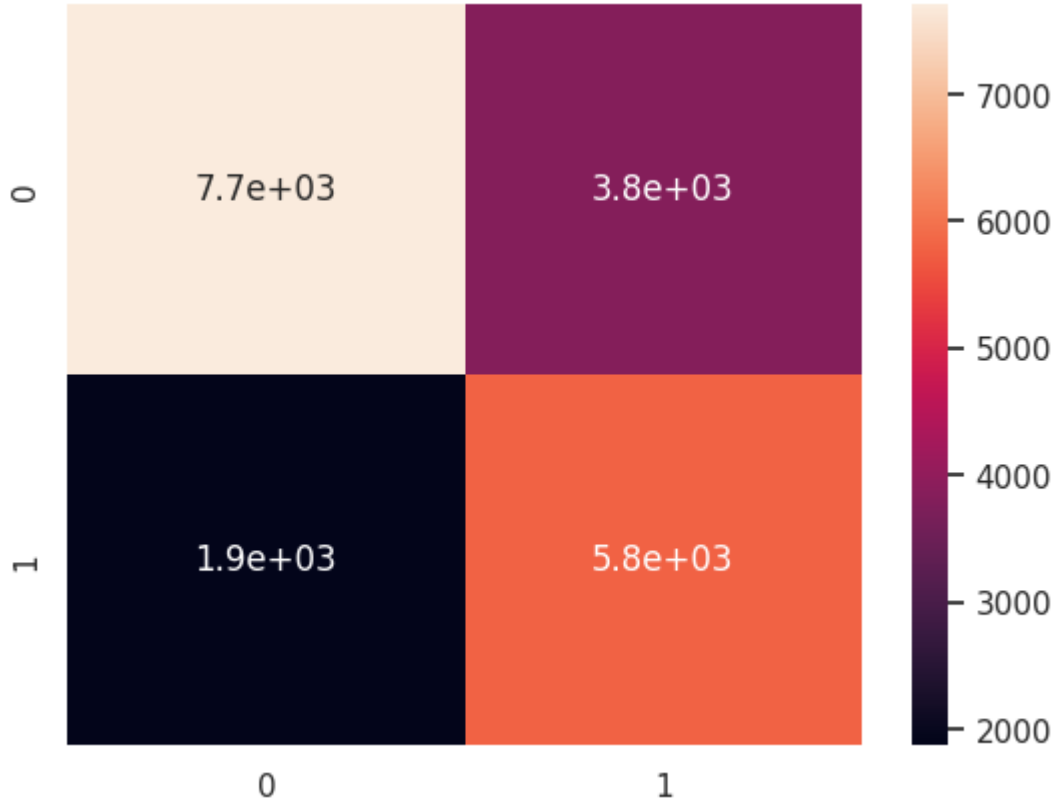
In [ ]:
```
print(accuracy = accuracy_score(y_test, y_pred))
```

In [ ]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.67      0.80      0.73      9579
         1.0       0.76      0.60      0.67      9604

    accuracy                           0.70     19183
   macro avg       0.71      0.70      0.70     19183
weighted avg       0.71      0.70      0.70     19183
```

In [ ]:
```
print(sns.heatmap(confusion_matrix(y_pred, y_test), annot=True))
```

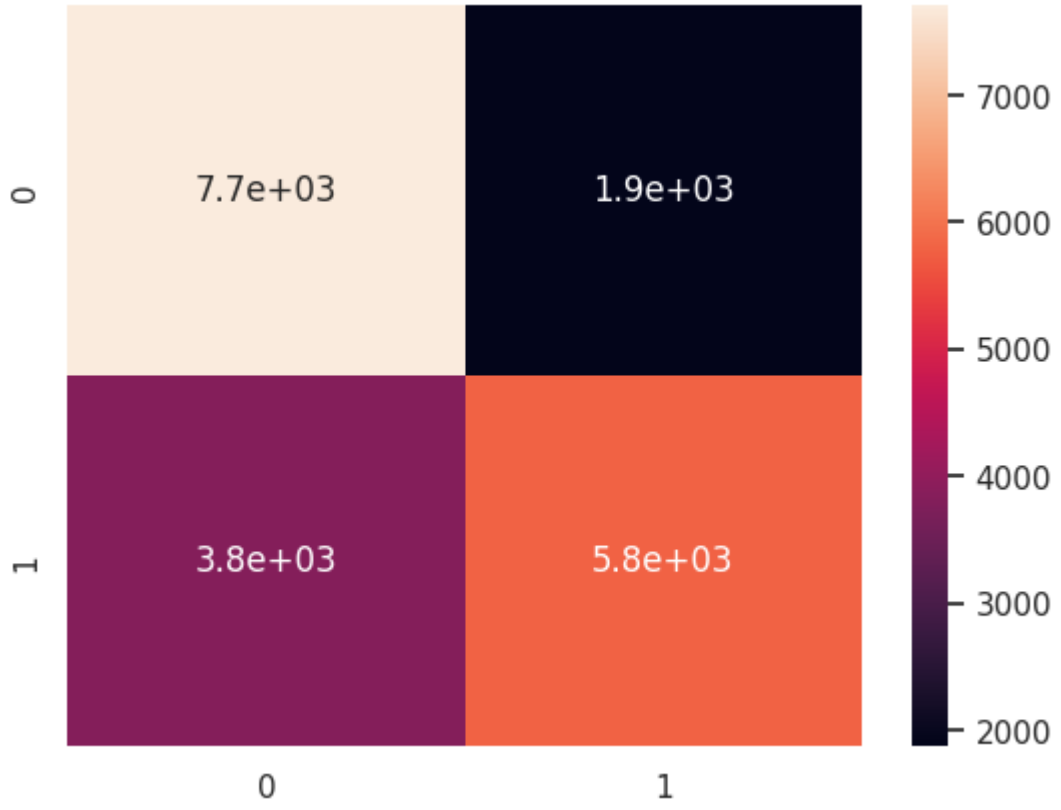
```
Axes(0.125,0.11;0.62x0.77)
```



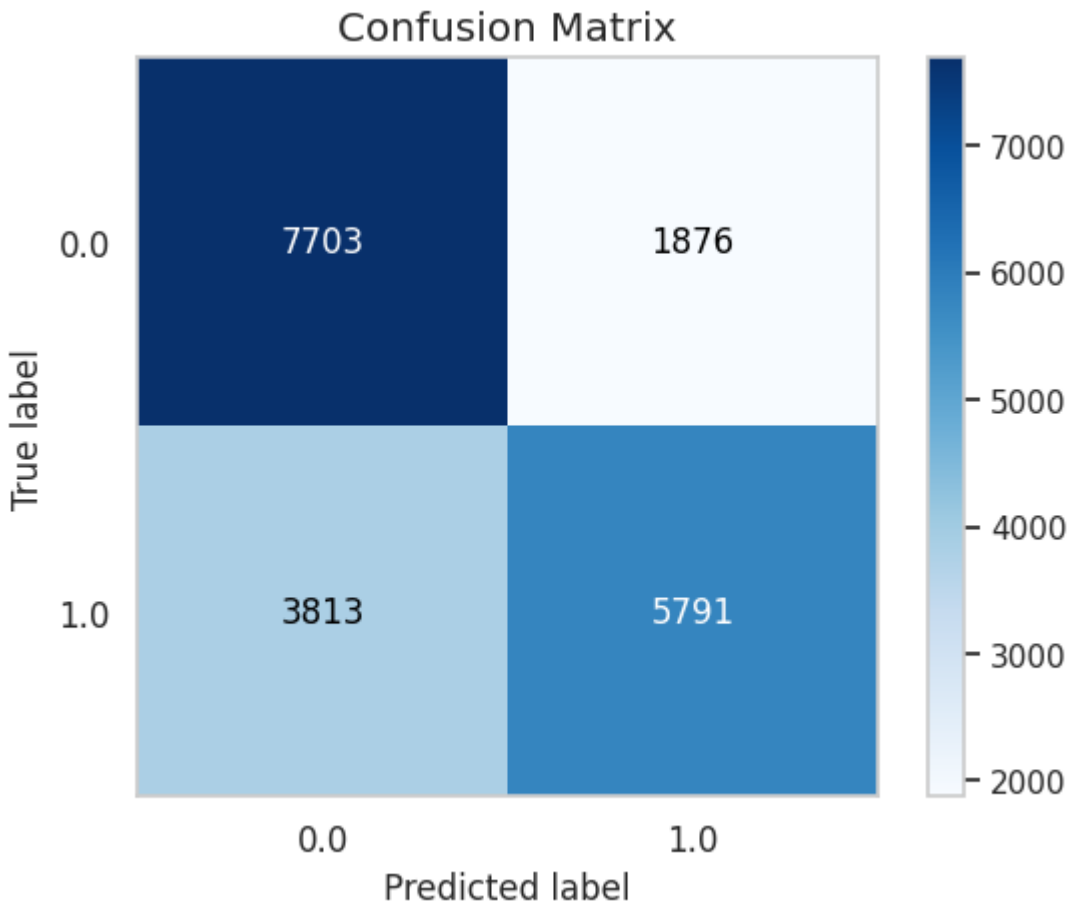
In [ ]:
```
#Evaluate the model
print(classification_report(y_test, y_pred))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True)
```

```
              precision    recall  f1-score   support

         0.0       0.67      0.80      0.73      9579
         1.0       0.76      0.60      0.67      9604

    accuracy                           0.70     19183
   macro avg       0.71      0.70      0.70     19183
weighted avg       0.71      0.70      0.70     19183
```

Out[ ]: <Axes: >



In [ ]:
```
# confusion matrix
skplt.metrics.plot_confusion_matrix(y_test, y_pred)
plt.show()

train_accuracy = accuracy_score(y_train, urlmodel.predict(X_train))
test_accuracy = accuracy_score(y_test, y_pred)
print("Train Accuracy:", train_accuracy)
print("Test Accuracy:", test_accuracy)

test_auc = accuracy_score(y_test, y_pred)
print("Test AUC:", test_auc)
```




```
Train Accuracy: 0.7043398931317607
Test Accuracy: 0.7034353333680863
Test AUC: 0.7034353333680863
```

Decision Tree

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [ ]:
```
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(X_train, y_train)
```

Out[ ]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [ ]:
```
y_pred_tree = decision_tree_model.predict(X_test)
```

In [ ]:
```
print("Decision Tree")
print("Accuracy:", accuracy_score(y_test, y_pred_tree))
print("Classification Report:\n", classification_report(y_test, y_pred_tr
```

```
Decision Tree
Accuracy: 0.8794244904342386
Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      0.87      0.88      9579
         1.0       0.87      0.89      0.88      9604

    accuracy                           0.88     19183
   macro avg       0.88      0.88      0.88     19183
weighted avg       0.88      0.88      0.88     19183
```

W&B

In [ ]:
```
!pip install wandb -qU
```

```
     6.8/6.8 MB 35.3 MB/s eta 0:00:00
     207.3/207.3 kB 9.3 MB/s eta 0:00
     303.6/303.6 kB 12.4 MB/s eta 0:0
     62.7/62.7 kB 3.9 MB/s eta 0:00:0
```

In [ ]:
```
import wandb
```

In [ ]:
```
wandb.login()
```

```
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https
wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/autho
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to qu
 ..........
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
```

Out[ ]:
```
True
```

In [ ]:
```
wandb.init(project='url-assignment')
```

```
Tracking run with wandb version 0.17.5
Run data is saved locally in /content/wandb/run-20240725_145522-czdjs4q9
Syncing run jolly-lion-2 to Weights & Biases (docs)
View project at https://wandb.ai/shamywams-usiu/url-assignment
View run at https://wandb.ai/shamywams-usiu/url-assignment/runs/czdjs4q9
```

Out[ ]: Display W&B run

In [ ]:
```
!pip install scikit-plot
import scikitplot as skplt
```

```
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: matplotlib>=1.4.0 in /usr/local/lib/python3.10/
packages (from scikit-plot) (3.7.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.10
dist-packages (from scikit-plot) (1.2.2)
Requirement already satisfied: scipy>=0.9 in /usr/local/lib/python3.10/dist-
packages (from scikit-plot) (1.11.4)
Requirement already satisfied: joblib>=0.10 in /usr/local/lib/python3.10/dist
packages (from scikit-plot) (1.4.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/
packages (from matplotlib>=1.4.0->scikit-plot) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist
packages (from matplotlib>=1.4.0->scikit-plot) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/
packages (from matplotlib>=1.4.0->scikit-plot) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/
packages (from matplotlib>=1.4.0->scikit-plot) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-
packages (from matplotlib>=1.4.0->scikit-plot) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/di
packages (from matplotlib>=1.4.0->scikit-plot) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist
packages (from matplotlib>=1.4.0->scikit-plot) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/
packages (from matplotlib>=1.4.0->scikit-plot) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3
dist-packages (from matplotlib>=1.4.0->scikit-plot) (2.8.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3
dist-packages (from scikit-learn>=0.18->scikit-plot) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-pacl
(from python-dateutil>=2.7->matplotlib>=1.4.0->scikit-plot) (1.16.0)
Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Installing collected packages: scikit-plot
Successfully installed scikit-plot-0.3.7
```

In [ ]:
```python
# Log metrics to wandb
epoch = 10
train_accuracy = accuracy_score(y_train, urlmodel.predict(X_train))
test_accuracy = accuracy_score(y_test, y_pred)
test_auc = accuracy_score(y_test, y_pred)
wandb.log({
      'epoch': epoch + 1,
        'train_accuracy': train_accuracy,
        'test_accuracy': test_accuracy,
        'test_auc': test_auc
    })
```

In [ ]:
```python
wandb.finish()
```

```
VBox(children=(Label(value='0.011 MB of 0.011 MB uploaded\r'),
FloatProgress(value=1.0, max=1.0)))
```

Run history:

epoch	
test_accuracy	
test_auc	
train_accuracy	

Run summary:

epoch	
test_accuracy	0.70
test_auc	0.70
train_accuracy	0.70

View run **jolly-lion-2** at: https://wandb.ai/shamywams-usiu/url-assignment/runs/czdjs4q9
View project at: https://wandb.ai/shamywams-usiu/url-assignment
Synced 4 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
Find logs at: `./wandb/run-20240725_145522-czdjs4q9/logs`
The new W&B backend becomes opt-out in version 0.18.0; try it out with `wandb.require("core")`! S
https://wandb.me/wandb-core for more information.