model-rf-1.py

Initially, we ran a test according to the features above

```python
selected_features = [
    'length_url',          # Length of the URL
    'length_hostname',     # Length of the hostname
    'ip',                  # Contains IP address (0/1)
    'nb_dots',             # Number of dots
    'nb_hyphens',          # Number of hyphens
    'nb_at',               # Number of @ symbols
    'nb_qm',               # Number of question marks
    'nb_and',              # Number of & symbols
    'nb_or',               # Number of | symbols
    'nb_eq',               # Number of = symbols
    'nb_underscore',       # Number of underscores
    'nb_tilde',            # Number of tildes
    'nb_percent',          # Number of % symbols
    'nb_slash',            # Number of slashes
    'nb_star',             # Number of asterisks
    'nb_colon',            # Number of colons
    'nb_comma',            # Number of commas
    'nb_semicolumn',       # Number of semicolons
    'nb_dollar',           # Number of dollar signs
    'nb_space',            # Number of spaces
    'nb_www',              # Contains 'www' (0/1)
    'nb_com',              # Contains '.com' (0/1)
    'nb_dslash',           # Number of double slashes
]
```

```
Test Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.81      0.84      1157
           1       0.82      0.87      0.85      1129

    accuracy                           0.84      2286
   macro avg       0.84      0.84      0.84      2286
weighted avg       0.84      0.84      0.84      2286


Confusion Matrix:
[[941 216]
 [143 986]]

Feature Importance:
            feature  importance
20           nb_www    0.304286
0        length_url    0.096482
13         nb_slash    0.090965
1   length_hostname    0.086754
2                ip    0.085892
4        nb_hyphens    0.083416
3           nb_dots    0.083311
6             nb_qm    0.047540
10    nb_underscore    0.037261
9             nb_eq    0.032834
7            nb_and    0.016426
12       nb_percent    0.007040
5             nb_at    0.006925
21           nb_com    0.005931
17    nb_semicolumn    0.004224
19         nb_space    0.003923
11         nb_tilde    0.002639
15         nb_colon    0.002280
22        nb_dslash    0.001269
16         nb_comma    0.000445
```

So this model have a well-balanced structure and have high detection rate for phishing of 87% recall, low false positive rate (13%) and showed us which feature have a more significant impact.

So we decided to remove the features with <0.1% importance. The features are:

22	nb_dslash	0.001269
16	nb_comma	0.000445
14	nb_star	0.000130
18	nb_dollar	0.000027
8	nb_or	0.000000

model-rf-2.py

We then ran the test again and achieved even better results from 84% to 85%

```
Test Accuracy: 0.85

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.82      0.84      1157
           1       0.82      0.88      0.85      1129

    accuracy                           0.85      2286
   macro avg       0.85      0.85      0.85      2286
weighted avg       0.85      0.85      0.85      2286


Confusion Matrix:
[[946 211]
 [141 988]]

Feature Importance:
            feature  importance
16           nb_www    0.286546
0        length_url    0.101616
12         nb_slash    0.097016
1   length_hostname    0.096184
4         nb_hyphens    0.091830
3           nb_dots    0.086010
2                ip    0.084598
6             nb_qm    0.043196
9     nb_underscore    0.037041
8             nb_eq    0.030180
7            nb_and    0.014767
11       nb_percent    0.007814
17           nb_com    0.007669
5             nb_at    0.004343
15         nb_space    0.003963
10         nb_tilde    0.003339
13         nb_colon    0.002467
14    nb_semicolumn    0.001420
```

model-rf-3.py

So we decided to try the same strategy again and remove bottom 3 features <

10 nb_tilde 0.003339

13 nb_colon 0.002467

14 nb_semicolumn 0.001420

```
Test Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.81      0.84      1157
           1       0.82      0.87      0.85      1129

    accuracy                           0.84      2286
   macro avg       0.84      0.84      0.84      2286
weighted avg       0.84      0.84      0.84      2286


Confusion Matrix:
[[942 215]
 [145 984]]

Feature Importance:
            feature  importance
13           nb_www    0.270924
1   length_hostname    0.104890
11         nb_slash    0.101419
0        length_url    0.100507
3           nb_dots    0.092736
4        nb_hyphens    0.087175
2                ip    0.070841
8             nb_eq    0.048189
6             nb_qm    0.046224
9     nb_underscore    0.036134
7            nb_and    0.014645
14           nb_com    0.008810
10       nb_percent    0.007464
5             nb_at    0.005697
12         nb_space    0.004345
```

This resulted in worst accuracy, thus we added it back

model-rf-4.py

Two features caught our attention in the training dataset which was “page_rank” and “google_index”. Thus, we decided to train our model with these features. Which improves the accuracy to 94%

```
Test Accuracy: 0.94

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      1157
           1       0.94      0.94      0.94      1129

    accuracy                           0.94      2286
   macro avg       0.94      0.94      0.94      2286
weighted avg       0.94      0.94      0.94      2286


Confusion Matrix:
[[1092   65]
 [  68 1061]]

Feature Importance:
            feature  importance
19     google_index    0.385174
18        page_rank    0.220077
16           nb_www    0.115648
0        length_url    0.044539
12         nb_slash    0.034060
1   length_hostname    0.033760
3           nb_dots    0.032628
2                ip    0.029446
4        nb_hyphens    0.028459
6             nb_qm    0.025781
8             nb_eq    0.016681
9     nb_underscore    0.008780
7            nb_and    0.006306
11       nb_percent    0.005998
15         nb_space    0.003603
13         nb_colon    0.003003
17           nb_com    0.002365
5             nb_at    0.001818
14    nb_semicolumn    0.001189
10         nb_tilde    0.000685
```

model-rf-5.py

We will then drop the data with the lowest impact to simplify our model without significant impact. Test accuracy stays the same at 94%. Thus we will stick to this model

```
low_impact = [
   'nb_tilde',       # 0.07%
   'nb_semicolumn',  # 0.12%
   'nb_at',          # 0.18%
   'nb_com',         # 0.24%
   'nb_colon',       # 0.30%
   'nb_space'        # 0.36%
]
```

```
Test Accuracy: 0.94

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      1157
           1       0.94      0.94      0.94      1129

    accuracy                           0.94      2286
   macro avg       0.94      0.94      0.94      2286
weighted avg       0.94      0.94      0.94      2286


Confusion Matrix:
[[1092   65]
 [  68 1061]]

Feature Importance:
            feature  importance
19     google_index    0.385174
18        page_rank    0.220077
16           nb_www    0.115648
0        length_url    0.044539
12         nb_slash    0.034060
1   length_hostname    0.033760
3           nb_dots    0.032628
2                ip    0.029446
4        nb_hyphens    0.028459
6             nb_qm    0.025781
8             nb_eq    0.016681
9     nb_underscore    0.008780
7            nb_and    0.006306
11       nb_percent    0.005998
15         nb_space    0.003603
13         nb_colon    0.003003
17           nb_com    0.002365
5             nb_at    0.001818
14    nb_semicolumn    0.001189
10         nb_tilde    0.000685
```

We have to convert the trained model into onxx
Run: pip install skl2onnx onnx onnxruntime