Cairo university  
Faculty of engineering  
Computer engineering department  
Machine Learning [**CMP4040**]   
Project Report

**Web page  
 Phishing Detection**

**Team 9**

|  |  |  |
| --- | --- | --- |
| Name | section | BN |
| احمد أسعد درويش محمد درويش | 1 | 1 |
| عمر فريد عبد العاطى لملوم | 2 | 4 |
| محمد نبيل عبد الفتاح فهمى | 2 | 19 |
| ممدوح احمد محمد محمد عطيه | 2 | 27 |

Presented to:

**Eng. Mohamed Shawky**

# Workload Division

|  |  |
| --- | --- |
| Name | Workload |
| احمد أسعد درويش محمد درويش | Data Preprocessing |
| عمر فريد عبد العاطى لملوم | Data Preprocessing |
| محمد نبيل عبد الفتاح فهمى | Models |
| ممدوح احمد محمد محمد عطيه | Models |

# Problem definition & Motivation

**Phishing** continue s to prove one of the most successful and effective ways for cybercriminals *to defraud us and steal our personal and financial information*.

**Our growing reliance** on the internet to conduct much of our day-to-day business has provided fraudsters with the perfect environment to launch targeted phishing attacks. The phishing attacks taking place today are sophisticated and increasingly more difficult to spot. A study conducted by Intel found that 97% of security experts fail at identifying phishing emails from genuine emails.

So in our ML project we would like to address this problem by training 3 phishing detection models and apply our knowledge to evaluate these using the following metrics for example:

# Evaluation metrics

Here are some of our proposed metrics (subject to add more of them – will be clarified in the final report إن شاء الله )

1. Accuracy
2. Confusion Matrix
   * which in turn include:
     1. TP : True positives
     2. TN : True Negatives
     3. FP : False positives
     4. FN : False negatives
3. F1 – Score
4. Precision
5. Recall

# Dataset Link

The dataset that we propose to use:

<https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset?resource=download>

# #1 : Dataset analysis

Let's talk about dataset analysis in the upcoming bullet-points

1. At first , we loaded the dataset from Kaggle site.
2. Explore the dataset : **info** – **description** – **shape.**
3. Data preprocessing : Drop duplicates – Drop nulls [There weren't any of these in our dataset]
4. Dataset visualization :
   1. **Histogram of features** : They gave me some insights about the feature values ranges and frequencies. Also you can notice that Many features are regex features 🡺 The majority of values are zero , and they take that values 0 or 1. At first I thought about dropping them , but said that they may turn to have useful information even if small.

صورة تحتوي على لقطة شاشة, رسم بياني, دائرة, الخط

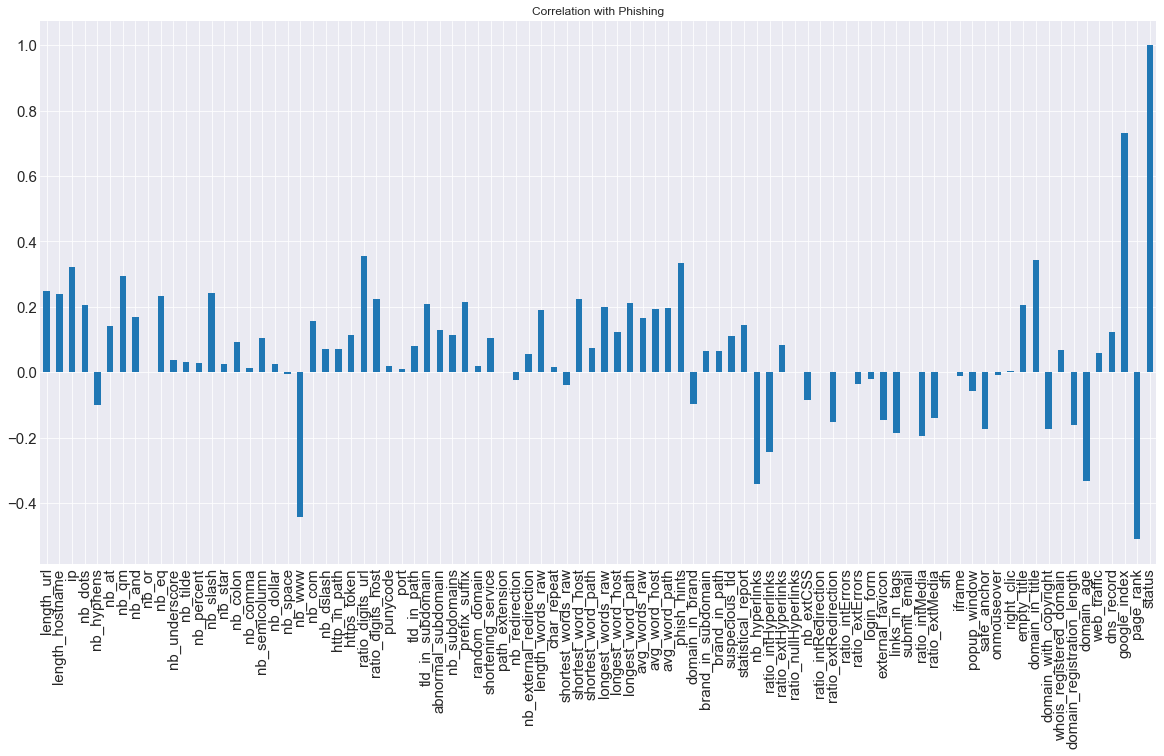
تم إنشاء الوصف تلقائياً

* 1. Pie chart of the output variable
     1. Concluded that **the dataset is balanced.**
  2. Correlation matrix

صورة تحتوي على نص, خط, لقطة شاشة, شكل

تم إنشاء الوصف تلقائياً

* 1. Correlation with the output variable yielded the following graph:



There were 2 experiments made , we will show the results before and after dropping the lowly correlated features (with target correlation < 0.1) in the experiments section below.

Anyway , after dropping the columns with correlation in range [ -1 : 1 ], here are the rest of the features after dropping these columns:

صورة تحتوي على نص, لقطة شاشة, تخطيط, الخط

تم إنشاء الوصف تلقائياً

Top 5 features with the highest correlation with the output variable

google\_index 0.731171

page\_rank 0.511137

nb\_www 0.443468

ratio\_digits\_url 0.356395

domain\_in\_title 0.342807

* 1. Box Plot (To analyze outliers)

صورة تحتوي على لقطة شاشة, مستطيل, نص, ميدان/ مربع

تم إنشاء الوصف تلقائياً

Woah ! umm well this is hard to view :)

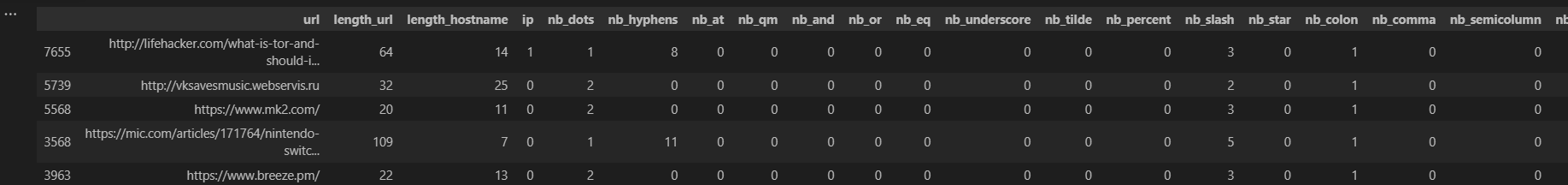
Some important notes from the box plot are:

1. The feature : "web\_traffic" has a lot of outliers.=> to solve this we can use log transformation.

2. Features ranges are different.=> to solve this we can use standardization.

Also , looks like the features needs scaling. :)

* 1. Took random data sample to view (please refer to notebook for full row view:



1. Data preprocessing:
   1. We have to convert the categorical data into numerical data

## the only categorical data are the target column and the url column

# we will convert the target column to numerical data

#by mapping the values : 1 for phishing and 0 for legitimate

* 1. The url column is not useful for the model so we will drop it
  2. Scaling the features using a StandardScaler.
  3. Fix the web\_traffic column values
     1. we will use the median value to replace the negative values

now it looks like this:

صورة تحتوي على لقطة شاشة, مستطيل, خط, نص

تم إنشاء الوصف تلقائياً

* 1. Redrawing BoxPlot after the scaling and fixing

صورة تحتوي على لقطة شاشة, خط

تم إنشاء الوصف تلقائياً

A bit better and the boxes are more apparent . For Better visualization kindly run the corresponding cell and open the plot from the cell and zoom in like this:

صورة تحتوي على لقطة شاشة, نص, رسم بياني, خط

تم إنشاء الوصف تلقائياً

At first it came to my mind to remove the remaining outliers. But after searching I decided to keep them because they are important for the model to learn the patterns, and gain insights from the data.

# #2 Experiments & Results

Experiment #1 : Without dropping low correlated features:

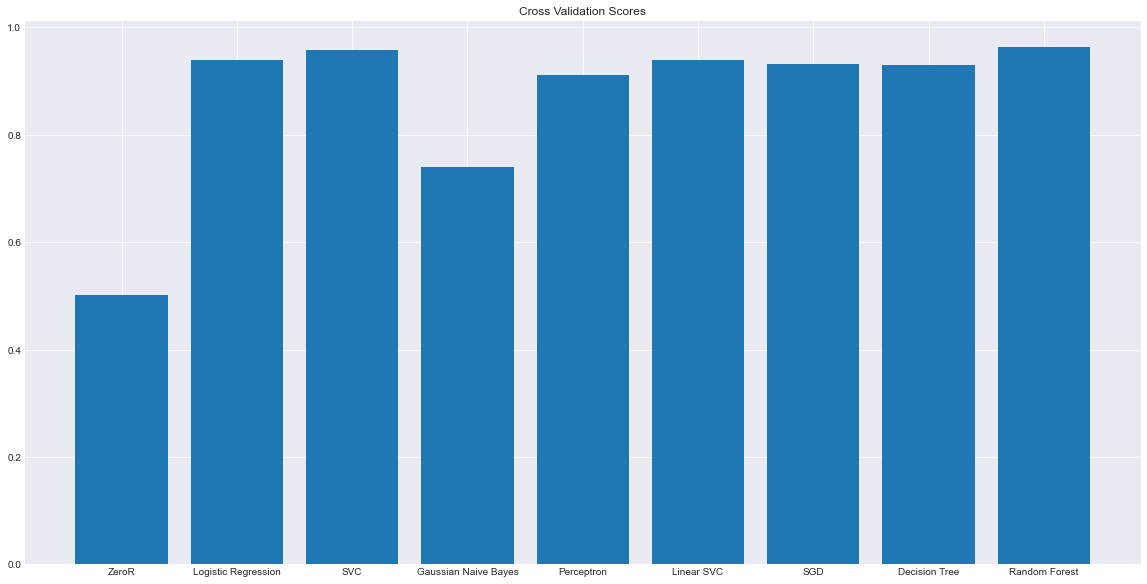
Experiment #2 : After dropping low correlated features (no hyperparameter tuning experiment):

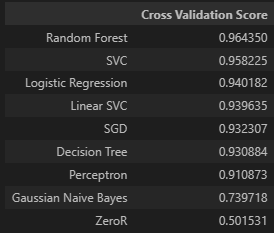
Models accuracies:

صورة تحتوي على نص, لقطة شاشة, الخط, رقم

تم إنشاء الوصف تلقائياً

Cross validation scores:





# Our conclusion