Phishing Detection

Using Machine Learning Techniques

Team - 7

Team Members :

Vamsi Srinivas (015268279)

Suleyman Saib (013512850)

Sashank Pidur Kuppuswamy (015279498)

Sajal Gupta(015359643)

Abstract:

With the increase in number of users and user's personal data there has also been a rapid increase in online scams. With this increase of scams there is a sheer need of security and privacy of users credentials such as username and password. Usually, it is difficult to classify a website as malicious because of changing syntax of the URLs and because of use of redirection caused by the use of shortened URLs. Our project solves the problem using supervised learning model-Decision Tree Classifier generally used for classification analysis.

Introduction:

A URL can be classified as malicious when it is incorrectly identified as a popular URL already existing on web by user. A user cannot always verify authenticity of a website due to the fast pace of change happening in UIUX and content of website and also because of new techniques developed to target user's information. Hence, to browse safe on internet our solution can be used to verify the website's authenticity. The tools and technologies used are primarily Python along with other third party packages to facilitate the readability and efficiency of the solution. A naive approach to detect a malicious URL would be downloading the HTML content from website consisting of sub URLs and searching for it's presence in the database of known phishing websites. This naive approach neither accurate results nor delivers the results in expected short time frame as it involves a database to rely which should be continuously updated for new phishing websites, API's availability, latency etc. Our solution introduces a machine learning approach which solves all the issues accounted in naive approach and provides results within expected time.

Literature Review:

Generally three strategies are used to detect malicious or phishing websites.

1) Contextual Analysis of Source Code: Based on the textual content of website a strategy which uses TF-IDF algorithm accurately classifies website as malicious or phishing with coverage of 97% having 6% false positive.

2) Analysis based on URL: We can analyze the parameters from URL such as it's length, domain name etc. also with the search engine page rank to determine the popularity which ultimately can be used to judge the website as safe or unsafe to visit. It's coverage is 97%.

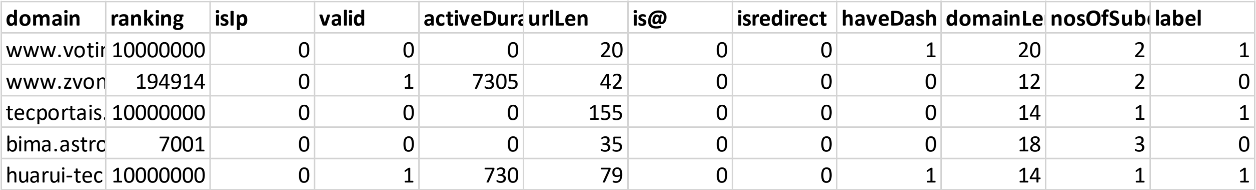
3) Analysis based on Machine Learning We can shortlist model based on the accuracy of training which uses different attributes derived from URL and page source code such as redirection, whether embedded URLs have an IP Address or domain name, URLs length etc. This analysis predicts 92% of websites correctly with 0.4% false positive rate.

Overview:

In this project we make initial exploratory data analysis and then depending on the target variable we list most suitable models to determine the suitability. After initial exploratory feature engineering and selection we fit the given data to train various models to extract predictions results and score. Based on the model's score and other parameters we finalize a model and for this particular model, then we determine the impact of different features based on correlations and feature importance technique. With the resultant analysis, few highly impacting features are shortlisted, and the selected model is tuned further for better scores. With this model and along with the third party API (Google Whois API) we determine if the site is phishing or legitimate.

Dataset :

The dataset considered for this project is taken from Kaggle website under the topic of phishing website dataset followed by the hyperlink clicking here. The dataset consists of total 12 columns including the target variable. Some of sample data from the given dataset is shown below.



domain	ranking	isIp	valid	activeDura	urlLen	is@	isredirect	haveDash	domainLe	nosOfSub	label
www.votir	10000000	0	0	0	20	0	0	1	20	2	1
www.zvon	194914	0	1	7305	42	0	0	0	12	2	0
tecportais.	10000000	0	0	0	155	0	0	0	14	1	1
bima.astro	7001	0	0	0	35	0	0	0	18	3	0
huarui-tec	10000000	0	1	730	79	0	0	1	14	1	1

All the features of this dataset are,

Domain: The URL of the website.

Ranking: the Ranking of the web page.

isIp: If an IP address is available in the weblink for the url.

valid: This data tells about the current status of the URL's registration. (fetched from google's whois API)

activeDuration: Gives the duration of the time since the registration up until now. (Also from whois API)

urlLen: The text length of the URL

is@: If the URL link has a '@' character present

isredirect: If the link has double dashes present together.

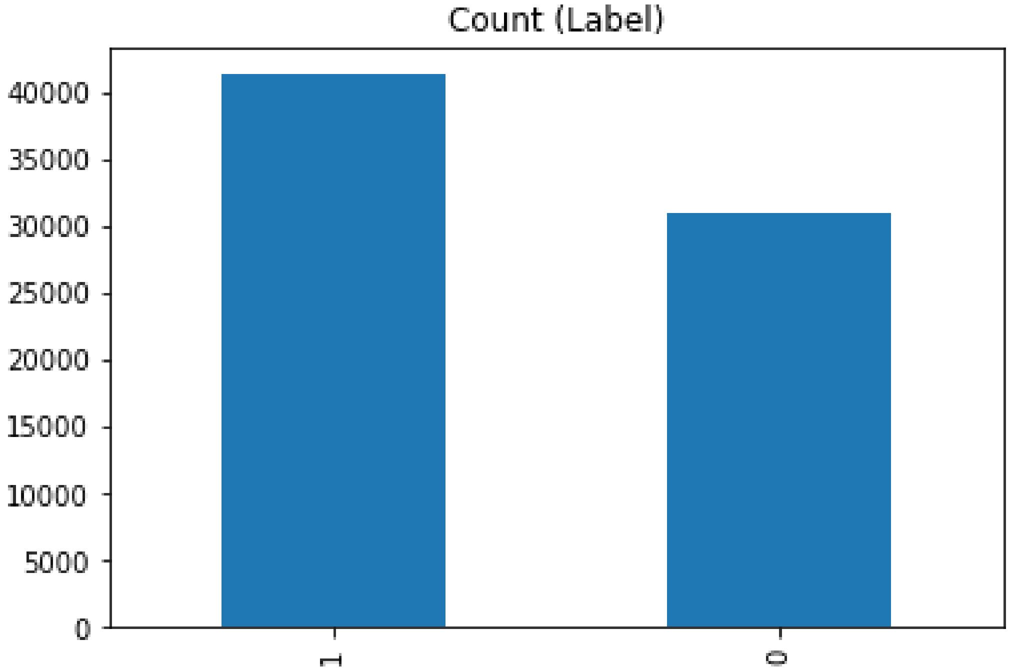
haveDash: If there are any dashes in the domain name.

domainLen: The length of the domain name only of the given URL.

noOfSubdomain: The number of subdomains preset in the URL.

Labels: Determines if the website is Legitimate or phishing. 0 -> Legitimate website , 1 -> Phishing Link/ Spam Link.

The Total number of records present in the given dataset is 95910 records, out of which 72363 records were observed to be duplicates. From the total unique records dataset, the records marked legitimate are 31025 and records marked as phishing are 41338.



Data Exploration:

The goal is to use the given dataset to train a model to accurately predict a given url as legitimate or phishing. Before the data could be trained we make initial data exploration for cleaning, feature engineering, determining feature importance and finally to select features.

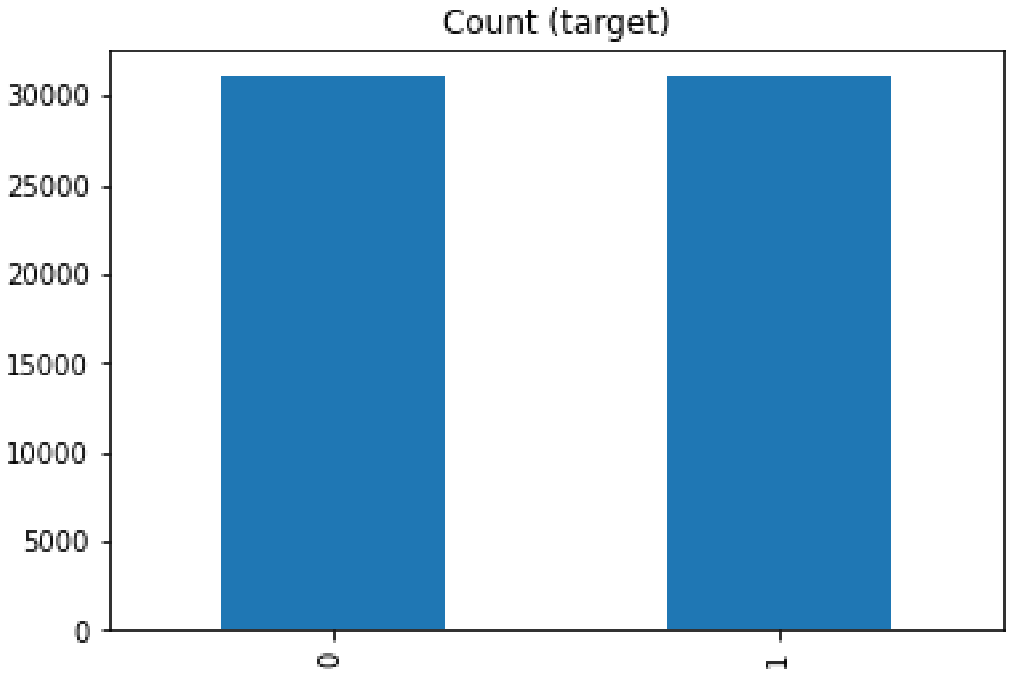
Data Cleaning:

After checking for unique records, we perform initial checks for any record with null values or empty values in the given dataset. The analysis resulted that no records are empty.

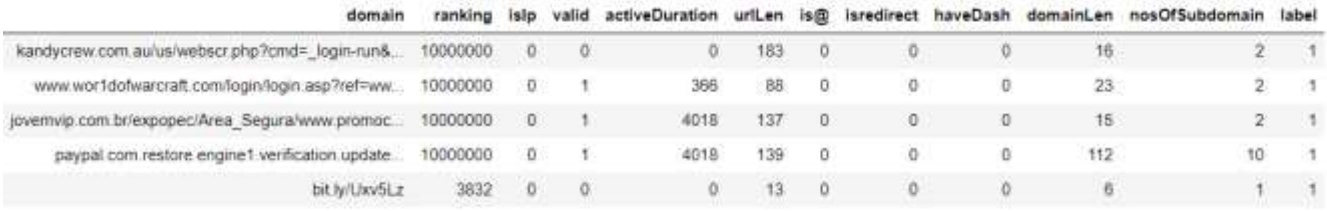
```
domain            0
ranking           0
isIp              0
valid             0
activeDuration    0
urlLen            0
is@               0
isredirect        0
haveDash          0
domainLen         0
nosOfSubdomain    0
label             0
```

Since the dataset has no null values, there is no requirement for any imputation techniques to fill the dataset. Further we analyse the other features for any outliers and errors. However, most of the features are binary interpretation of the given URL for special characters such as ‘@’ or ‘-‘. The features that consists of continuous numerical values are ranking, activeDuration, urlLen, domainLen and nosOfSubdomain. Out of these features, ranking and activeDuration are API based data and could not be cross verified for legitimacy while other URL based features are cross verified with the given URL. It was observed that there was no outliers observed, as most of the data are generated from the URL.

Since the given data set consists of uneven number of records for different labels, for the initial model selection we under sample the label with more records to same size of label with less records to maintain low bias.



This reduced dataset is then provided for the Feature engineering and transformation process to maintain a certain level of uniformity.



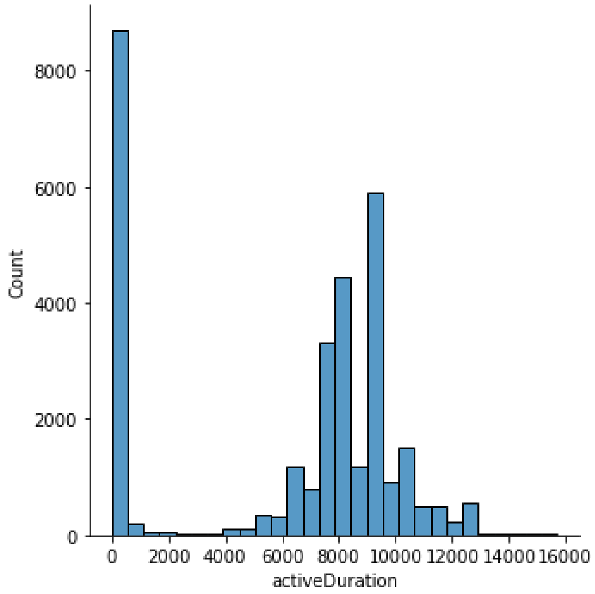
domain	ranking	isIp	valid	activeDuration	urlLen	is@	isredirect	haveDash	domainLen	nosOfSubdomain	label
kandycrew.com.au/us/webscr.php?cmd=_login-run&...	10000000	0	0	0	183	0	0	0	16	2	1
www.wor1dofwarcraft.com/login/login.asp?ref=ww...	10000000	0	1	366	88	0	0	0	23	2	1
jovemvip.com.br/expopec/Area_Segura/www.promoc...	10000000	0	1	4018	137	0	0	0	15	2	1
paypal.com.restore.engine1.verification.update...	10000000	0	1	4018	139	0	0	0	112	10	1
bit.ly/Uxv5Lz	3832	0	0	0	13	0	0	0	6	1	1

Feature Engineering and Transformation:

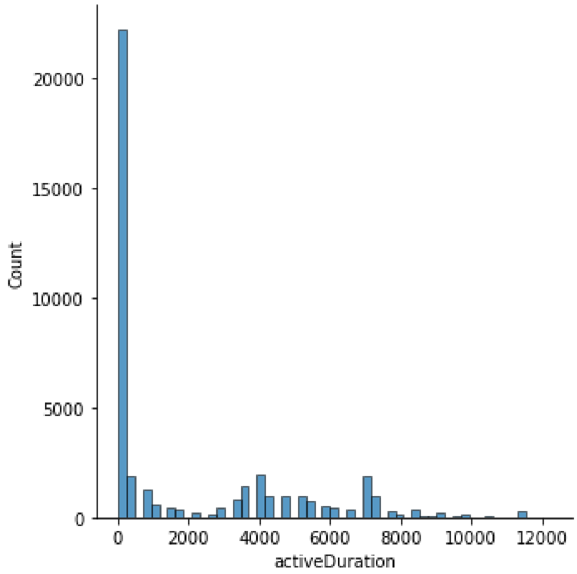
Since most of the given data is in binary format, for the initial model selection process we convert the continuous features to binary format using the threshold. We are converting the rank of the URL, active duration of the URL, and number of sub domains of the URL to binary format based on the below distribution observation.

Active Duration :

For Legitimate Sites



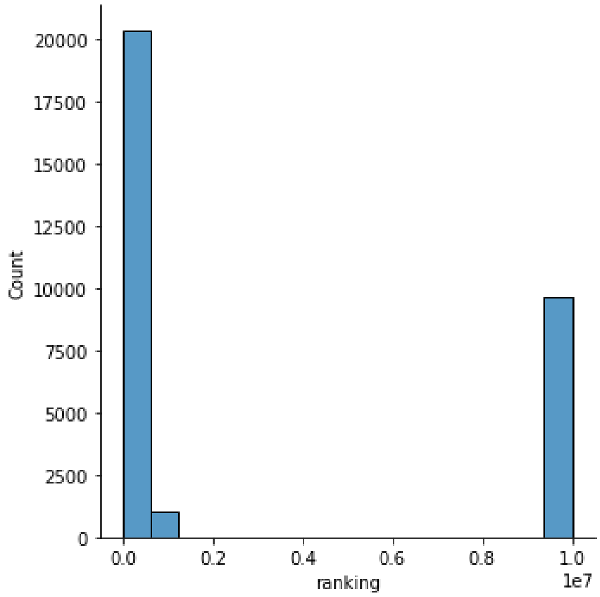
For Phishing sites



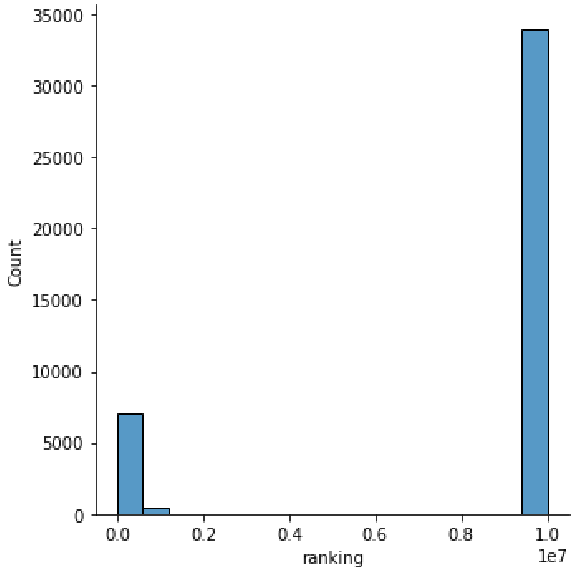
From the above observed graphs we can see that the max of 10000 covers almost 90% of the phishing websites.

Ranking :

Ranks for Legitimate Site



Ranks of Phishing sites

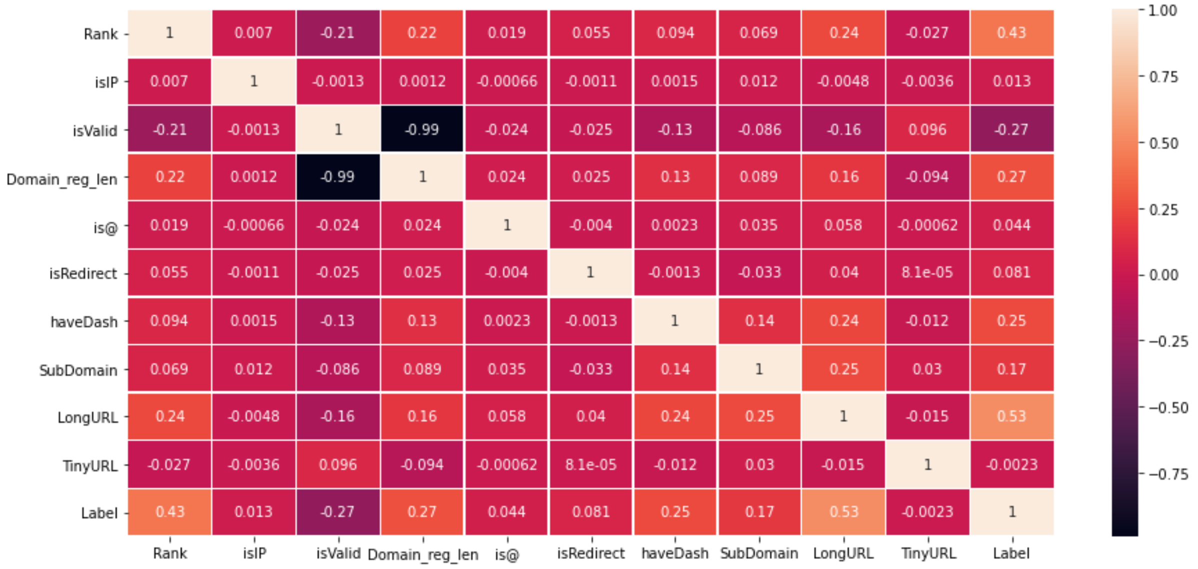


From the above distribution graphs we can observe that most of the phishing websites have the rank more than 10^7. So we mark the records with rankings grater than 100000 as 1 and others as 0.

Similarly we mark the records with more than 1 subdomains as 1 and others as 0.

Apart from these features we engineering new features such as determining if the url is tiny url in nature such as bit.ly or goo.gl and mark them as 1. Furthermore we mark the URLs with unsecure http and mark them as 1. Also we determine a new feature based on the URL length such that if the URL is of length greater than 54 characters we then mark it as longer URL with 1.

With these new engineered features along with the converted features, we determine the correlation between each features. This Correlation is observed using the help of a heatmap graph which is observed below.

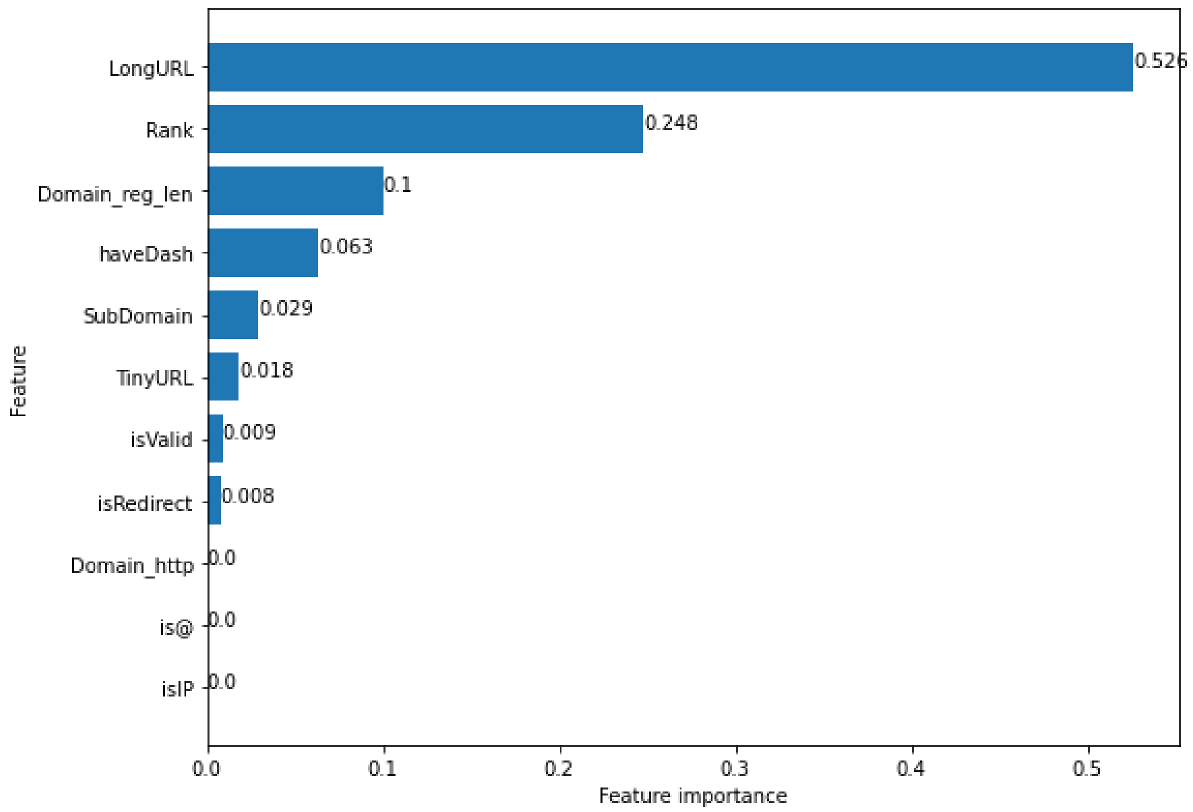


	Rank	isIP	isValid	Domain_reg_len	is@	isRedirect	haveDash	SubDomain	LongURL	TinyURL	Label
Rank	1	0.007	-0.21	0.22	0.019	0.055	0.094	0.069	0.24	-0.027	0.43
isIP	0.007	1	-0.0013	0.0012	-0.00066	-0.0011	0.0015	0.012	-0.0048	-0.0036	0.013
isValid	-0.21	-0.0013	1	-0.99	-0.024	-0.025	-0.13	-0.086	-0.16	0.096	-0.27
Domain_reg_len	0.22	0.0012	-0.99	1	0.024	0.025	0.13	0.089	0.16	-0.094	0.27
is@	0.019	-0.00066	-0.024	0.024	1	-0.004	0.0023	0.035	0.058	-0.00062	0.044
isRedirect	0.055	-0.0011	-0.025	0.025	-0.004	1	-0.0013	-0.033	0.04	8.1e-05	0.081
haveDash	0.094	0.0015	-0.13	0.13	0.0023	-0.0013	1	0.14	0.24	-0.012	0.25
SubDomain	0.069	0.012	-0.086	0.089	0.035	-0.033	0.14	1	0.25	0.03	0.17
LongURL	0.24	-0.0048	-0.16	0.16	0.058	0.04	0.24	0.25	1	-0.015	0.53
TinyURL	-0.027	-0.0036	0.096	-0.094	-0.00062	8.1e-05	-0.012	0.03	-0.015	1	-0.0023
Label	0.43	0.013	-0.27	0.27	0.044	0.081	0.25	0.17	0.53	-0.0023	1

From the heatmap graph, we can observe that the features rank, longURL, have dash and domain len has higher correlation values compared to others with the target variable. Furthermore, we determine the Feature importance to determine the exact features to be selected for the model and get in depth insight on the newly created features.

Feature Importance:

From the above heatmap graph it is clear that the above mentioned features are having higher importance and impact on the target variable than other features. However to certain this as a fact we are using feature importance technique. From this technique the observed features sorted by their importance values are as shown below.



The feature importance graph also matches with the heatmap graph ascertaining the fact that the features LongURL, rank, Domain len and have dash having higher impact than other features. Since other features such as is@ and isIP are of with 0 importance, we are dropping these features and proceeding to Model selection for the given Data.

Model Selection:

Since the target variable is of the type classification categories and binary in nature, The models considered are,

- Decision Trees
- Random Forest Classification
- XGBoost Algorithms
- SVM Models
- Naive Bayesian Classification
- Logistic Regression
- Neural Networks

The major reason to consider the following models are,

1) The problem is classification in nature :
   Decision trees, Random forest, XGBoost, Naive Bayesian classifiers

2) The problem is binary in classification nature:
   Logistic regression
3) The given Data features are binary in nature:
   SVM, NeuralNetwork

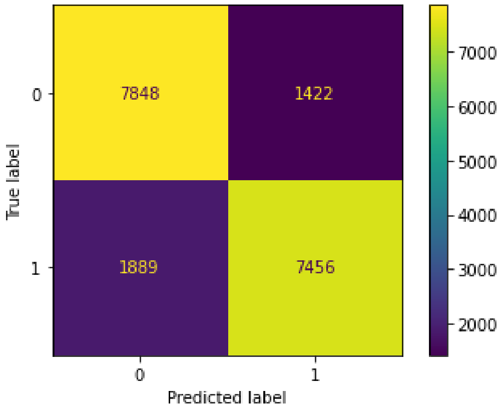
All the models are trained and tested with the train-test ratio of 70:30, and the results of each models are observed by the classification report, Confusion matrix and precision-recall graphs.

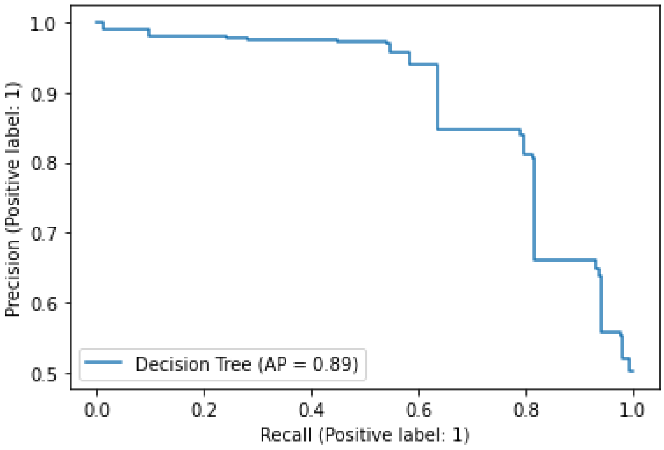
Decision tree :

Decision tree classifier are a non-parametric supervised learning method used for classification. The Decision tree model predicts the value of a target variable by learning simple decision rules inferred from the data features. The features are grouped based on their purity and the tree is divided into nodes until the data in every node is completely pure. The results observed for the Decision tree model is as shown below.

Classification Report :

```
                   precision    recall  f1-score   support

             0       0.81      0.85      0.83      9270
             1       0.84      0.80      0.82      9345

      accuracy                           0.82     18615
     macro avg       0.82      0.82      0.82     18615
weighted avg       0.82      0.82      0.82     18615
```

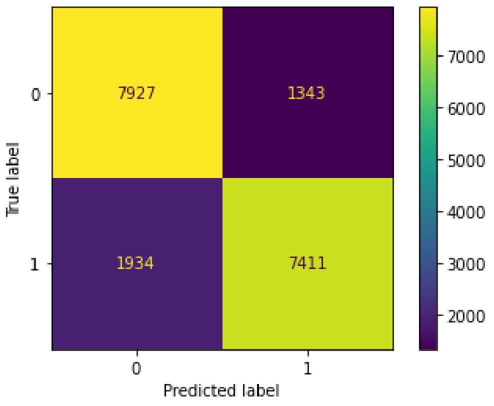


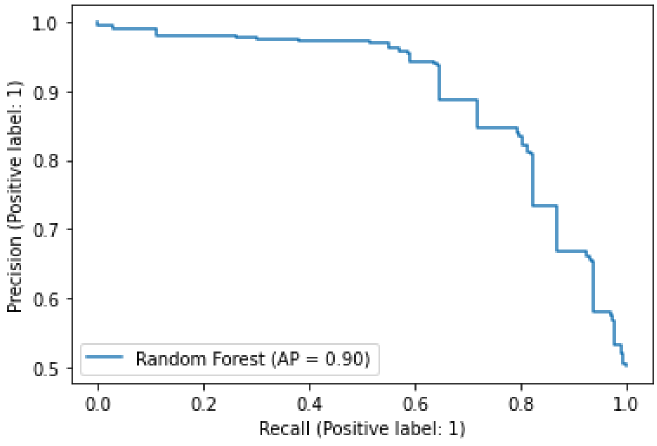


Random Forest classifier:

It is a Decision Tree based classifier. Random Forest creates multiple trees based on a random value of vector. The strategy for Random Forest is to combine multiple ’weak learners’ in to one ’strong learner’. Random Forest overcomes the problems of single Decision Tree based algorithms which are over fitting and issues related to high variance and bias. The results observed for the Random forest model is as shown below.

```
        Classification Report:
                   precision    recall  f1-score   support


                0      0.80      0.86      0.83      9270
                1      0.85      0.79      0.82      9345

          accuracy                         0.82     18615
         macro avg      0.83      0.82      0.82     18615
weighted avg      0.83      0.82      0.82     18615
```

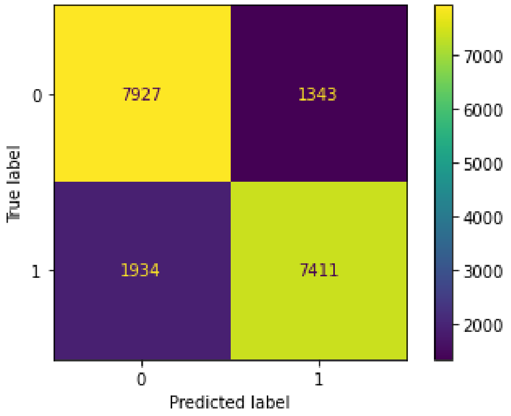


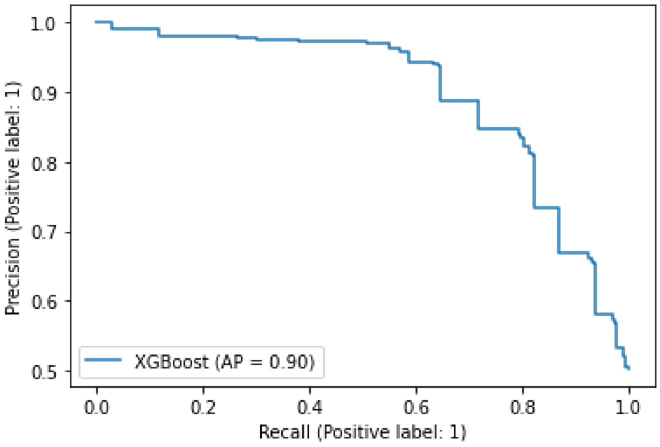


XGBoost Classifier :

XGboost classifier is also a tree based classifier model. However, Rather than training all of the models in isolation of one another, boosting trains models in succession, with each new model being trained to correct the errors made by the previous ones. The results observed for XGBoost model are,

```
          Classification Report:
                     precision    recall  f1-score   support

                  0       0.80      0.86      0.83      9270
                  1       0.85      0.79      0.82      9345

             accuracy                           0.82     18615
            macro avg       0.83      0.82      0.82     18615
weighted avg       0.83      0.82      0.82     18615
```

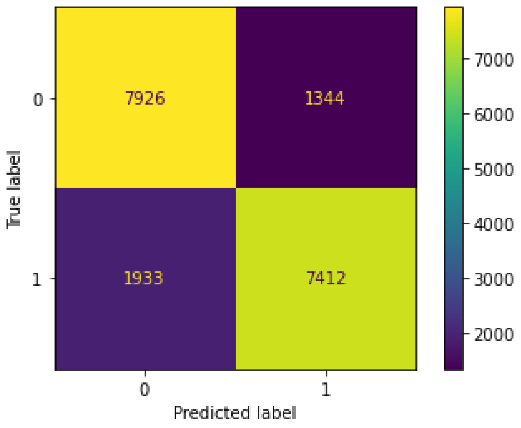


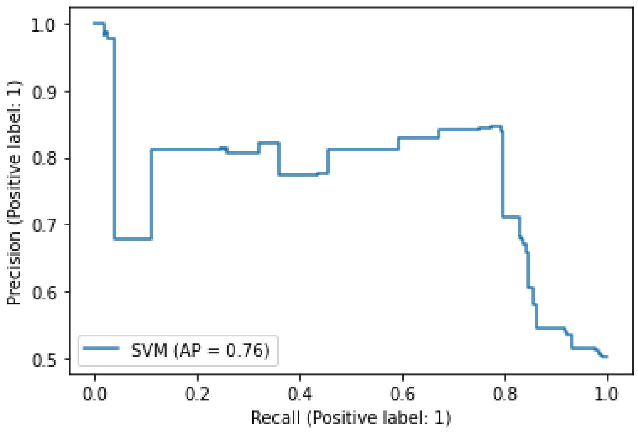


SVM :

A Support Vector Machines are also used for classification as well as regression. SVM uses a line or plane depending on number of dimensions and creates a hyperplane to classify the data points. The Observed results are,

```
          Classification Report:
                     precision    recall  f1-score   support

                  0       0.80      0.86      0.83      9270
                  1       0.85      0.79      0.82      9345

             accuracy                           0.82     18615
            macro avg       0.83      0.82      0.82     18615
          weighted avg       0.83      0.82      0.82     18615
```



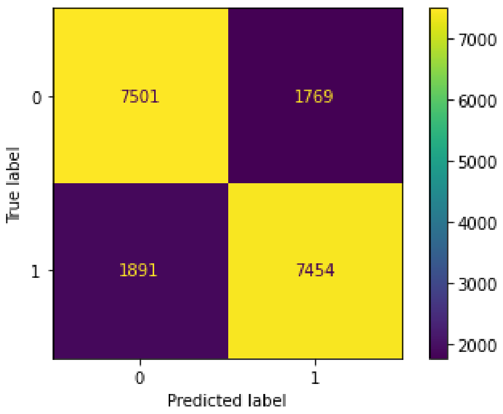


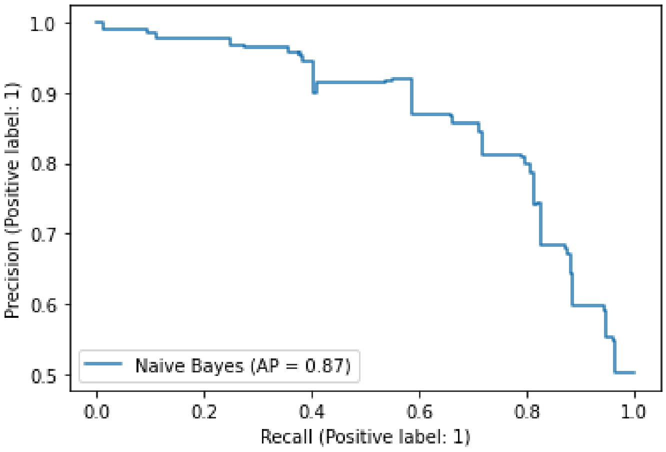
Naive Bayes:

Naïve Bayes classifiers are a class of simple classifiers which applies strong independence between features. This model is the simplest among the Bayesian models. The advantage of this model is that it is highly scalable. The results observed for this model are.

Classification Report:

```
              precision    recall  f1-score   support

           0       0.80      0.81      0.80      9270
           1       0.81      0.80      0.80      9345

    accuracy                           0.80     18615
   macro avg       0.80      0.80      0.80     18615
weighted avg       0.80      0.80      0.80     18615
```



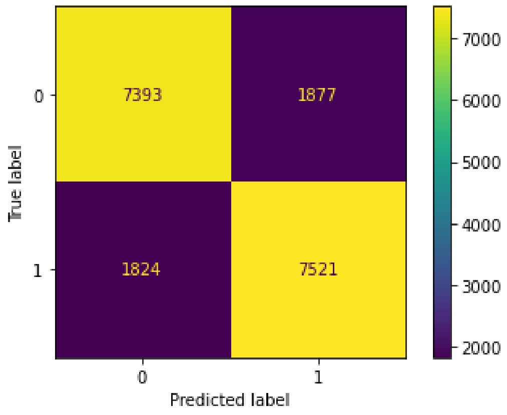


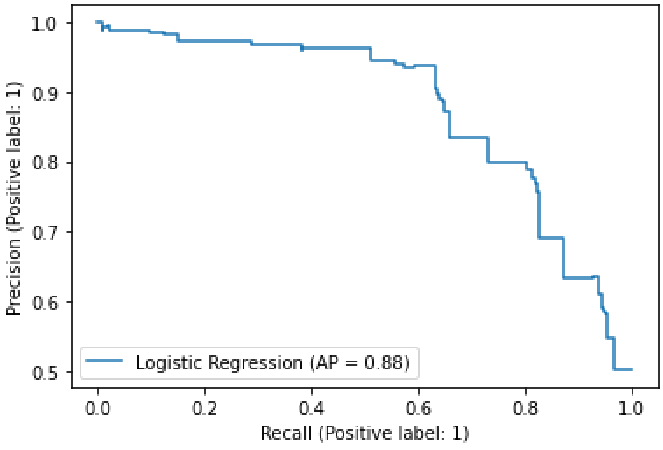
Logistic Regression :

Logistic regression model is the appropriate regression analysis to conduct when the dependent variable is binary in nature. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. The observed results are,

Classification Report:

```
              precision    recall  f1-score   support

           0       0.80      0.80      0.80      9270
           1       0.80      0.80      0.80      9345

    accuracy                           0.80     18615
   macro avg       0.80      0.80      0.80     18615
weighted avg       0.80      0.80      0.80     18615
```

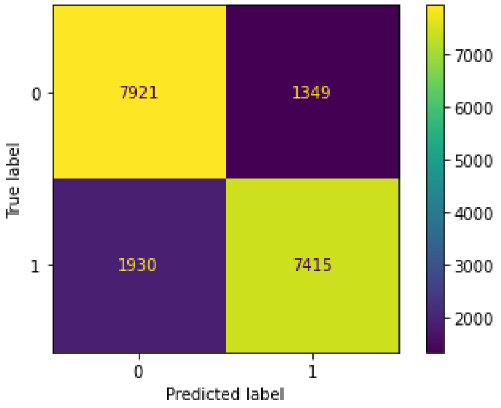


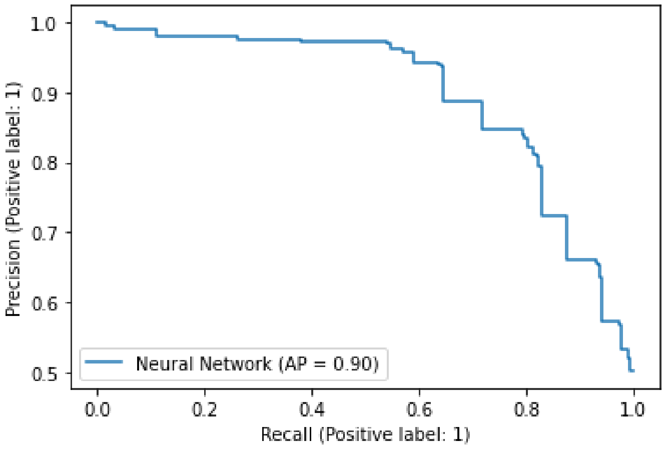


Neural Networks:

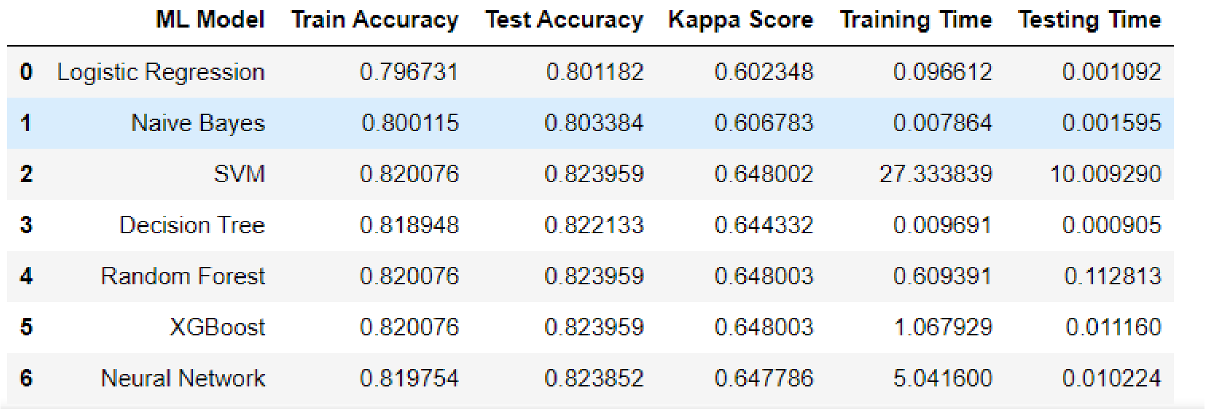
Neural Networks classifier model considered is MLP classifier. The model includes an input layer, an output (or target) layer and a hidden layers. The Activation function considered is Relu and the solver considered in this model is adam solver. The results observed are,

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.85      0.83      9270
           1       0.85      0.79      0.82      9345

    accuracy                           0.82     18615
   macro avg       0.83      0.82      0.82     18615
weighted avg       0.83      0.82      0.82     18615
```



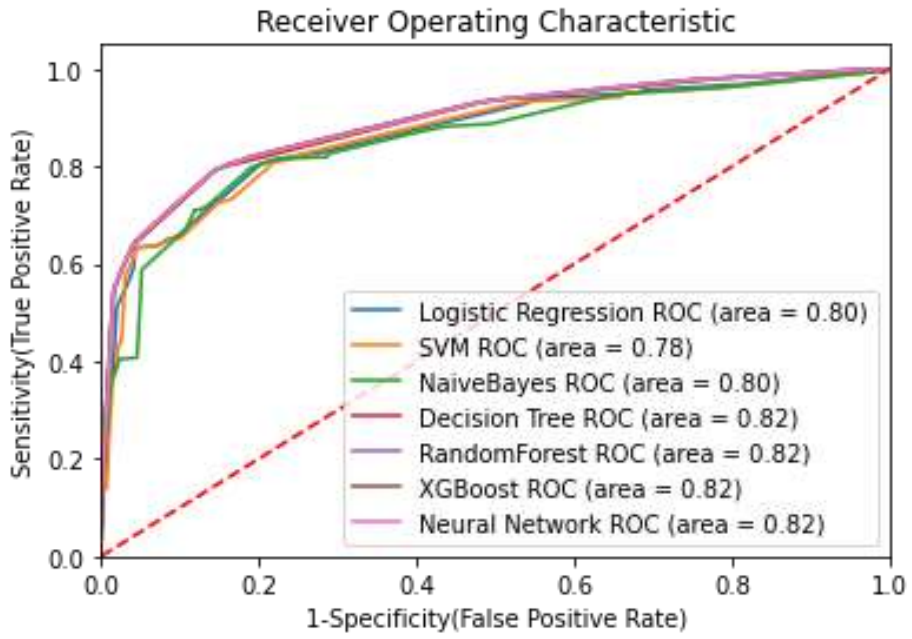


While running the above models, the training, testing accuracy scores and the respective time taken ar e also noted and the observations are shown below.



	ML Model	Train Accuracy	Test Accuracy	Kappa Score	Training Time	Testing Time
0	Logistic Regression	0.796731	0.801182	0.602348	0.096612	0.001092
1	Naive Bayes	0.800115	0.803384	0.606783	0.007864	0.001595
2	SVM	0.820076	0.823959	0.648002	27.333839	10.009290
3	Decision Tree	0.818948	0.822133	0.644332	0.009691	0.000905
4	Random Forest	0.820076	0.823959	0.648003	0.609391	0.112813
5	XGBoost	0.820076	0.823959	0.648003	1.067929	0.011160
6	Neural Network	0.819754	0.823852	0.647786	5.041600	0.010224

All the model’s ROC graph is also constructed along with the annotations of area under curve in the le gend.



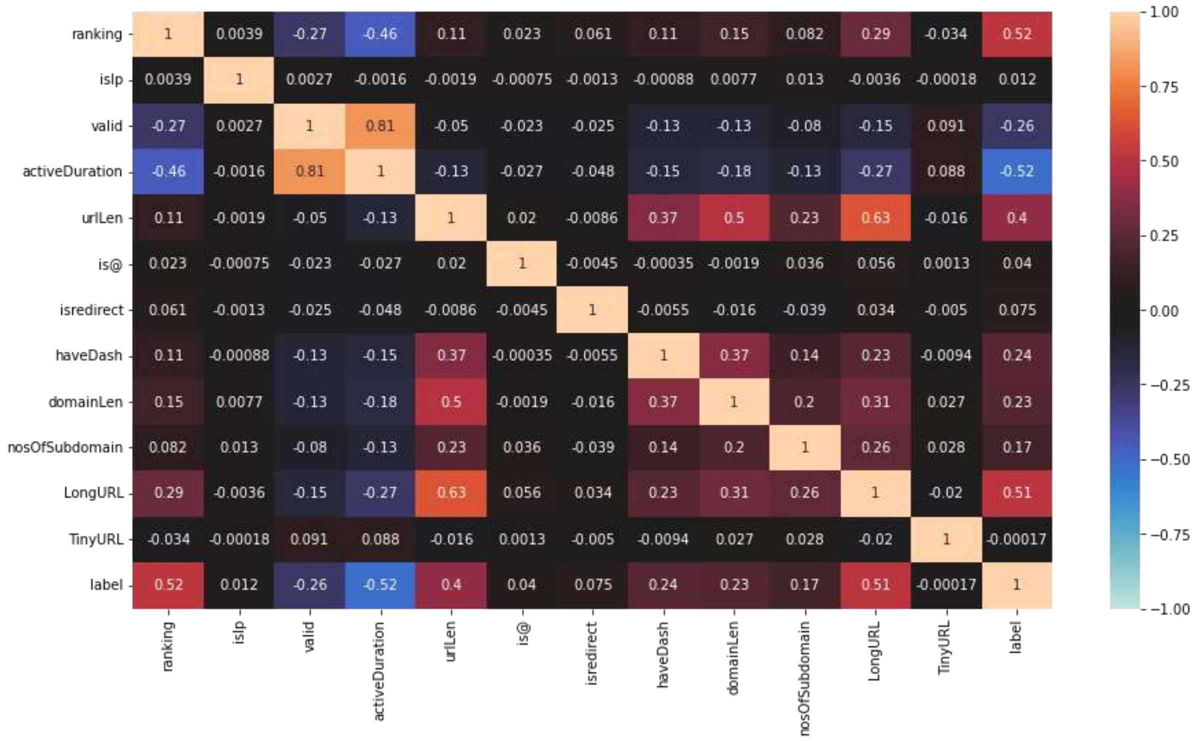
Choosing the Best Model:

From the observations of the ROC curves, the highest value observed is 82% and the models that achieve this score are Decision Tree, Random Forest, XGBoost, and Neural Network. Comparing these results to the testing accuracy scores and training time, it is observed that the Decision tree model achieves almost same accuracy score for least amount of training time. Considering these criteria the best model for the given data is determined to be Decision Tree Classifier. Since the observed accuracy score is only in the range of 82%, this score could be potentially increased by,

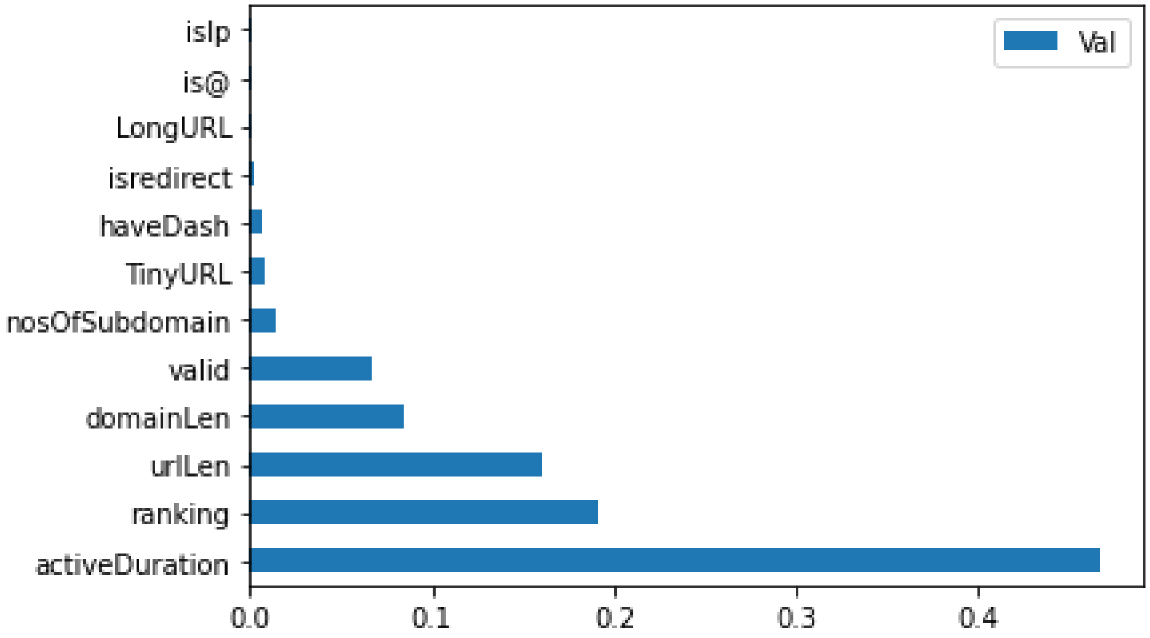
1) Proper feature selection
2) Hyper parameter tuning

Best Model Tuning:

In the Initial Iteration we had converted the continuous data into binary format. However, since the best model for the problem has been identified, we try to determine the actual set of important features without any transformation to existing features. The observed correlational graph is shown below.



From the above correlation map it is observed that the features of Rank, ActiveDuration, URL Length and Long URL are observed to have high correlation. This is again ascertained using feature importance technique, whose results are shown below.



How ever from the feature importance graph, we can see that the Long URL has almost 0 importance than tinyURL feature. This might be because the Long URL feature is in direct corelation with URL length feature.

From the above graphs we infer that active duration of the URL, rank of the URL and URL length are of higher importance. Hence we consider only these features for our final model. From the considered feature selection, The accuracy score has a jump from 83% to around 91%.

Further more after using the method of grid search, we determined that for the selected features, and the best parameters of the Decision Tree model, The final Accuracy score is around the range of 92 – 93%.

Result:

From the initial exploratory analysis to the Final model selection, the major criteria considered is accuracy score. The final Decision Tree model after the Grid search method resulted in best parameters as, criteria: entropy, Max depth for the tree is 12 and min sample split is 10. For these criteria the accuracy score is observed to be around the range of 92-93%. To further validate the model, we have considered the cohen kappa score, which determines the interrater reliability or precision. A general cohen kappa score of greater than 80% determines a near perfect agreement and the final model achieves the cohen kappa score of 86%.

Conclusion:

For the given dataset of legitimate and phishing websites, the major challenge was in determining the features to be engineered that has good impact on the target variable. Since most of the data in the data set was prepared from google’s whois API, there was very limited scope for feature engineering and the potential features that were engineered were of less importance in classifying the given URLs. From the default dataset we can observe that only the features impacting the classification are considered in the model and most of the features are not derived from the URL. Hence to determine the new site that is not part of the training data to be classified properly, the new testing url is parsed to who is API to get the required features and then the received features are passed to the model for prediction. To simplify the classification process, the best model is saved in a pickle file and is being read from the file for every prediction.

Future Works:

Since most of the features are not URL based, a URL based text processing technique could be used to improve the prediction of phishing sites without relying on the third party API. Further expansion on this work would suggest the usage of TF-IDF model or some Natural Language processing technique to determine the phishing websites as most of the phishing websites might not follow English words. Potential reduction on features from currently considered features count of four could also be explored.