Contents

[Problem Statement 1](#_Toc26738148)

[Data Set and Experiment 1](#_Toc26738149)

[Data Set 1](#_Toc26738150)

[Experiment 1](#_Toc26738151)

[1. Tools Used 1](#_Toc26738152)

[2. Data Cleaning, Feature extraction and Data Visulization 2](#_Toc26738153)

[3. Supervised learning models used with K-fold validations 4](#_Toc26738154)

[Evaluation 5](#_Toc26738155)

[Additions and Improvements 5](#_Toc26738156)

[Member Contributions 5](#_Toc26738157)

[GitHub and Kaggle Link 5](#_Toc26738158)

Agrawal, Harshit – 90411685

Kalawatia, Noopur R K - 1980 9834

Kunal, Kumar - 5100 5964

Singh, Richa - 5044 5135

Anti-Phishing Classifier

# Problem Statement

Phishing is one of the most common online security threats which is targeted towards both users and companies. A phishing website poses itself as a legitimate business website in order to obtain users credentials and personal information. Using these phishers can access financial information such as bank details, credit card information, health insurance details, social security information etc. making user vulnerable to identity theft and information breaches.

For this project we created an anti-phishing classifier which uses various features of the website to detect if the website is a legitimate website or a phishing website. We utilized 10 different features broadly classified into following features –

1. Website URL
2. Domain
3. Page Rank
4. Website content

# Data Set and Experiment

## Data Set

To train the model we needed sample which had both legitimate and phishing data. We pulled data from various sources –

1. PhishTank – public dataset containing ~10,000 phishing websites.
2. PhishStorm – dataset containing ~96,000 URLs: ~48,000 legitimate URLs and ~48,000 phishing URLs.
3. Alexa PageRank - ~1000 legitimate website URLs.

The dataset is segregated into training and testing datasets. The training dataset is used to perform k-fold validations over each of the models and the performance with each of the models is documented. Then the models are exposed to the completely untouched dataset. This performance gives us the real-world application efficiency.

## Experiment

### Tools Used

* 1. Data Cleaning
     + Pandas
     + WhoIs
  2. AWS
     + EMR
     + EC2
     + S3
     + Boto3 (For AWS operations)
  3. Data Visualization
     + Matplotlib
     + Seaborn
  4. Classification
     + Sklearn
     + Numpy
  5. UI application
     + Tkinter
     + Pandas

### Data Cleaning, Feature extraction and Data Visulization

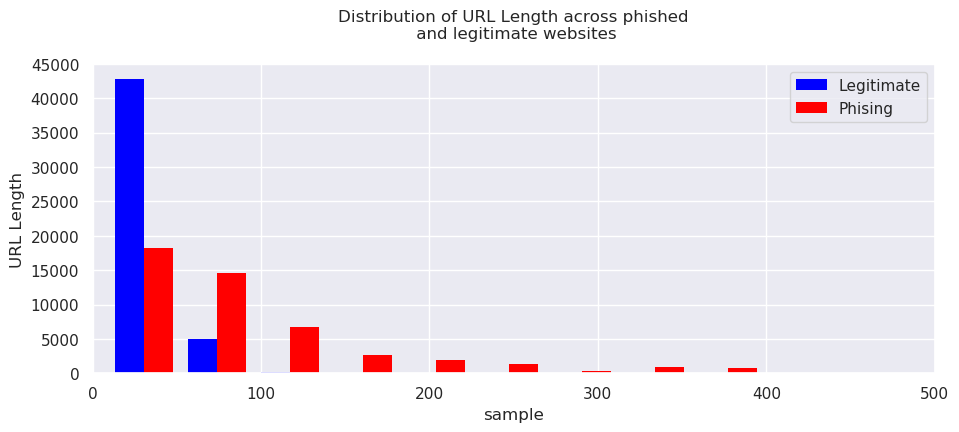
We merged the data from two sources mentioned above. We have utilized few features which were present in the existing dataset, but mostly extracted the features.

Tools used –

#### Features–

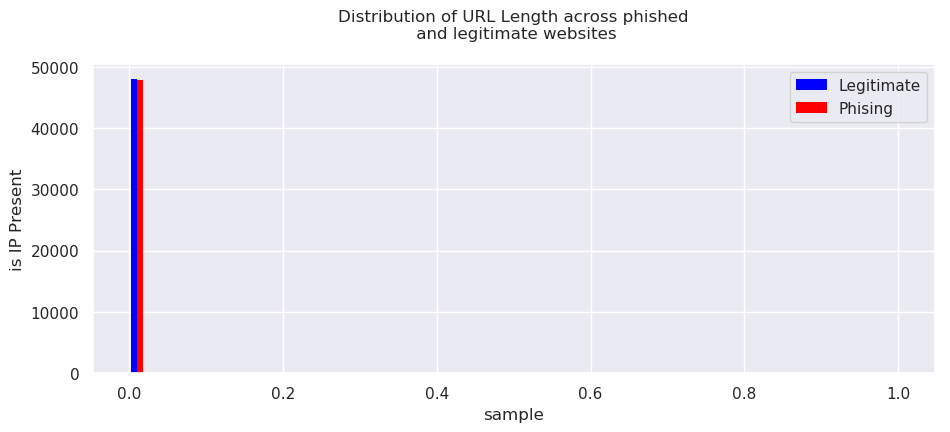
* 1. **URL Length**

Phishers generally uses long URLs to hide the doubtful part in the address bar.



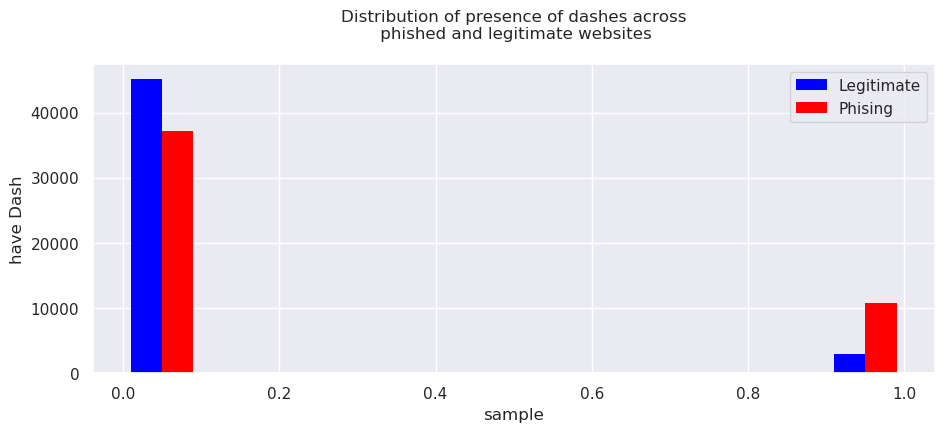
* 1. **Presence of IP address in the URL**

If an IP address is used as an alternative of the domain name in the URL users can be sure that someone is trying to steal their personal information. Although this feature clearly identifies a phishing website, but phishing websites are also evolving, hence, this doesn’t play a strong role anymore.



* 1. **Present of dash (-) in URL**

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage.

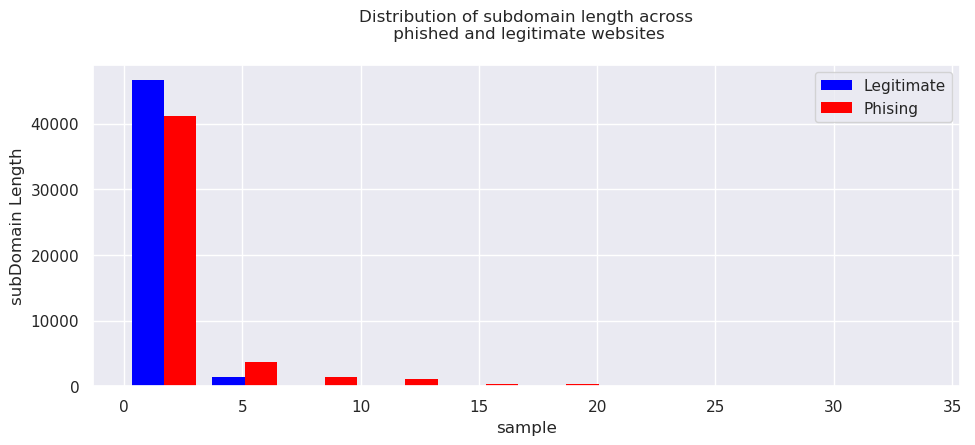


* 1. **Number of Domain Names**

Varying number of domains also is a sign of phishing website

* 1. **Number of Sub Domains**

If the dots are greater than two, it is classified as “Phishing” since it will have multiple sub domains.

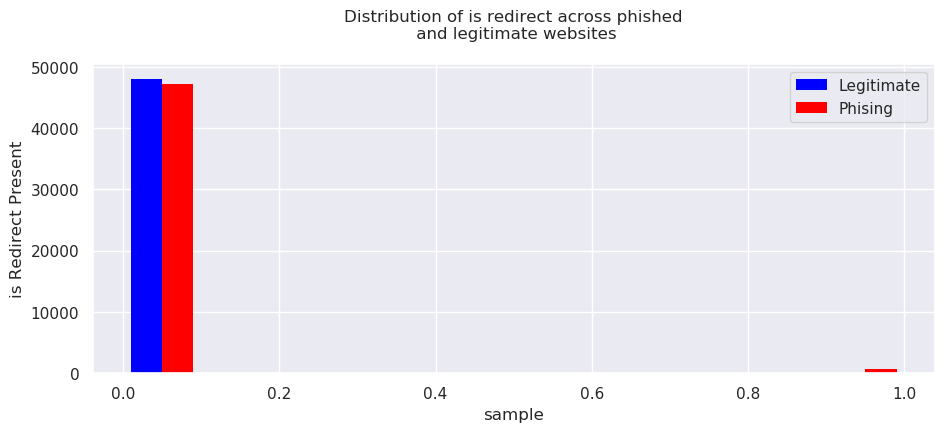


* 1. **Present of @ symbol**

Using “@” symbol in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol.

* 1. **Is Redirect link**

If a website is a redirect website, identified by the presence of “//”, there is a high possibility that the website is a phishing website.



* 1. **Domain name presence in WhoIs**

From [ICANN Lookup](https://lookup.icann.org/) using WhoIs API we are identifying if the domain is registered or not.

* 1. **Active Duration**

If a domain is registered in WhoIs, we are also extracting active duration of the website. Generally, phishing websites have short active duration as they are quickly taken down once they are reported. For the missing data we added default value of ‘0’.

* 1. **Page Ranking**

We are using PageRank information provided in PhisStorm data set, and if not present we are using Google page ranking.

*~~From the PhishStorm data we observed that for Page Rank for phishing websites was set as 10,000. Although this should be low, we implemented the same pattern in PhishTank dataset~~*

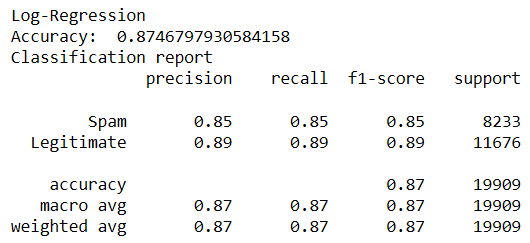
### Supervised learning models used with K-fold validations

The goal of using 10-fold validations is to obtain the best possible hyper parameter tuning over various runs over different divisions of the training partition of the dataset. Our goal is to compare how consistent each model is during the training and testing phase. This gives us an insight of how good the mined features are for all the different models and hints at whether the selected features need to be refined/rethought for more effective training. Performed Cross Validation with an 80-20 split for training and testing data for a data of ~106,000 datapoints

* 1. **Logistic -Regression**

Penalty = *L1, L2*

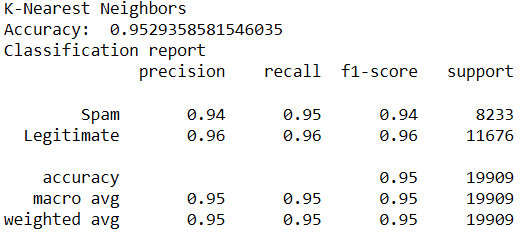
**Score** = 0.87468



* 1. **K Nearest Neighbors**

Metric = *minkowski* N Neighbors = *100* Weights = *Distance*

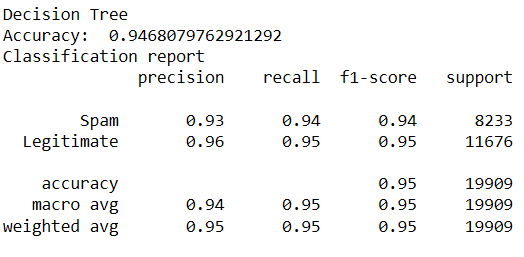
**Score** = *0.95294*



* 1. **Decision Tree**

Criterion = *gini* Min Sample Split = *50*

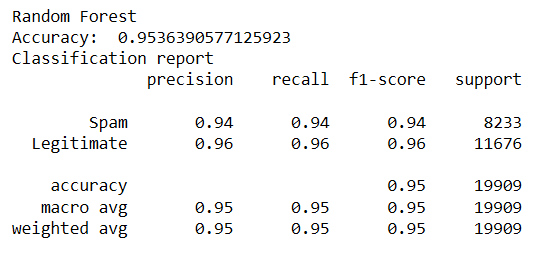
**Score** = *0.94681*



* 1. **Random Forest**

Bootstrap=*True* Criterion = *gini* Max Depth = *20* N Estimators=*100*

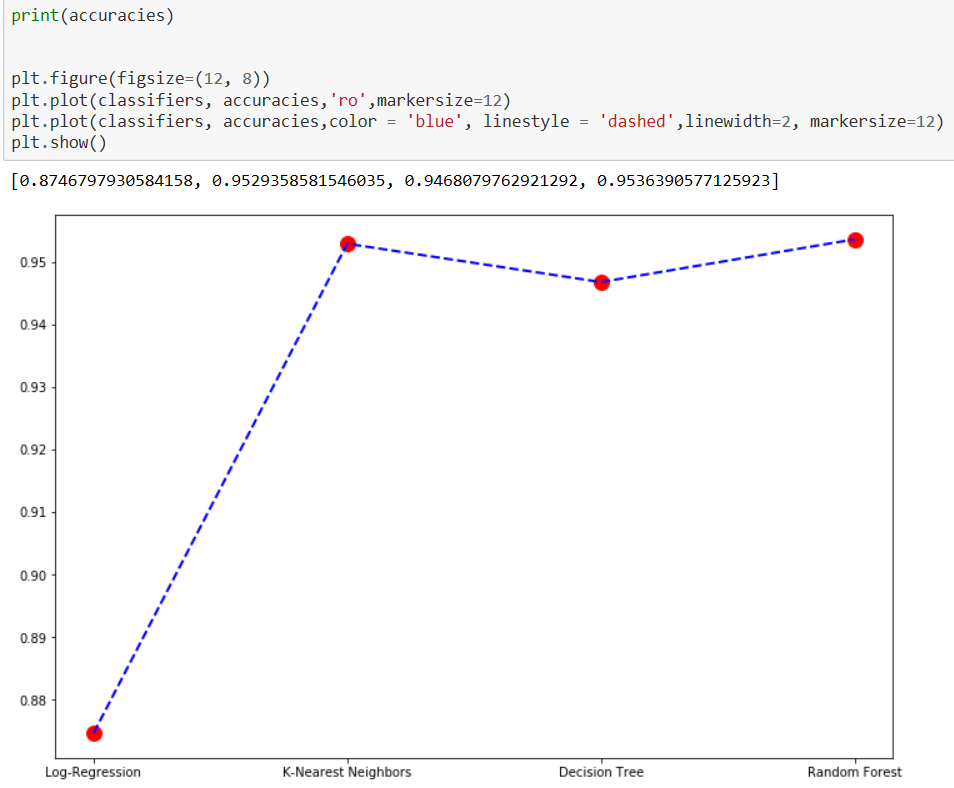
**Score** = 0.95364



# Evaluation

The metric used for analyzing the performance is F1-score. This mitigates the problems imposed by "accuracy" for imbalanced datasets.

|  |  |
| --- | --- |
| **Models used for Classification** | **Accuracy** |
| LogisticRegression | 87% |
| KNeighborsClassifier | 95% |
| DecisionTreeClassifier | 94.6% |
| RandomForestClassifier | 95.4% |



# Additions and Improvements

1. Increased dataset
2. Tweaked some features and fixed a bug while extracting isValid feature using WhoIs service which increased the accuracy of the model.
3. Recalibration few models to increase efficiency.
4. Created a UI based application using *Tkinter* in python, which takes user’s input and identifies if the site is phishing or a legitimate site. This can also be integrated with other applications and can be directly used to identify if the received link is phishing or not.

# Member Contributions

* Data Cleaning, feature creation and UI based application
  + Agrawal, Harshit
  + Singh, Richa
* Machine Learning models and Data visualization
  + Kalawatia, Noopur R K
  + Kunal, Kumar

# GitHub and Kaggle Link

* GitHub – <https://github.com/kunal4892/AntiPhishingClassifier>
* Kaggle – <https://www.kaggle.com/kunal4892/phishingandlegitimateurls>