**Detailed Project Report ( D.P.R )**



**Project On:**

**Title: Phishing Domain Detection**

**By: - Dharavath Ramdas**

**(iNeuron.ai)**

**Date: - 12/03/2023**

**Contents**

**Abstract**

**Introduction**

**Importance of Detailed Project Report (DPR)?**

1. **Description.**
   1. Problem Perspective.
   2. Problem Statement.
   3. Proposed Solution.
   4. Solution Improvement.
2. **Requirements.**
   1. Hardware Requirements.
   2. Technical Requirements.
3. **Data Requirements.**
   1. Data Collection.
   2. Data Description.
   3. Dataset Characteristic.
   4. License
4. **Data Preprocessing.**
5. **Model Workflow.**
   1. Model Selection
   2. Model Accuracy Score.
6. **Life cycle of a Machine learning project.**
7. **Data Collection / Inputs from the user.**
8. **Data Validation.**
9. **Rendering the results.**
10. **Deployment on Cloud.**

**Conclusion**

**Abstract.**

Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels. Phishing is popular among attackers because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to break through a computer's protection measures.

Phishing stands for a fraudulent process, where an attacker tries to obtain sensitive information from the victim. Usually, these kinds of attacks are done via emails, text messages, or websites. Phishing websites, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim. Discovering and detecting phishing websites has recently also gained the machine learning community’s attention, which has built the models and performed classifications of phishing websites. This paper presents two dataset variations that consist of 58,645 and 88,647 websites labeled as legitimate or phishing and allow the researchers to train their classification models, build phishing detection systems, and mining association rules.

The mail goal is to predict whether the domains are real or malicious Features in the dataset that are used for the prediction of url include

'directory\_length', 'time\_domain\_activation', 'length\_url','file\_length', 'qty\_slash\_url', 'qty\_plus\_directory', 'domain\_length', 'qty\_vowels\_domain', 'qty\_asterisk\_directory', 'qty\_hyphen\_directory', 'qty\_dot\_domain', 'qty\_underline\_directory', 'qty\_percent\_directory',

'qty\_dot\_url', 'qty\_hyphen\_url', 'qty\_hyphen\_file', 'qty\_hyphen\_domain',

'params\_length', 'qty\_underline\_url', 'qty\_tld\_url', 'qty\_plus\_params',

'qty\_percent\_url', 'qty\_equal\_params', 'qty\_dot\_params', 'qty\_percent\_params', 'qty\_underline\_params','phishing'.

For prediction, xgboost classifier has been chosen as a final algorithm to create the model we trained the system and achieved an accuracy of 93%.

**Introduction.**

Importance of DPR Documentation?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and written code. This also provides the detailed description on how the entire project has been designed end to end.

Key Points:

* Describes the Design flow
* Implementation
* Software requirements
* Architecture of project
* Non-functional attributes like:
  + Reusability
  + Portability
  + Resource utilization

1. **Description.**
   1. **Problem Perspective.**

The Phishing Domain Detection is a hyper-tuned machine learning classification model which helps to determine the whether the domains are real or malicious.

* 1. **Problem Statement.**

Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels. Phishing is popular among attackers because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to break through a computer's protection measures.

The mail goal is to predict whether the domains are real or malicious.

Approach: The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that’s best fit for the above case.

For Feature Engineering show:-

1. URL-Based Features

2. Domain-Based Features

3. Page-Based Features

4. Content-Based Features

* 1. **Proposed Solution.**

The solution proposed is to take the required batch file to predict the result. A pipeline has been created to get the prediction for the new dataset. A Flask webapp has been created to get the prediction based on certain inputs.

* 1. **Solution Improvements.**

The system can be made more futuristic by performing more hyper-tuning methods so that the prediction can be more accurately predictive. The project code has been designed in such a way that whenever new data will come, the model will go under training and if there will be an improvement in the model then the new model will be used for prediction.

**2. Requirements**

**2.1Hardware Requirements: -**

* A working computer to code with an active internet connection.

**2.2 Tools / Software Requirements: -**

* Python version used for this project 3.8 (This may get updated and some features might not be available in new version.)
* Python libraries such as NumPy, pandas, flask, Jinja2, HTML, matplotlib, PyMongo, seaborn and scikit-learn (Used for implementation of machine learning algorithms.)
* Jupyter Notebook & Visual studio code is used as an IDE for writing the code.
* Github is used as the version control system
* AWS is used for deployment using docker image.
* Apache Airflow has been used to monitor the ML model.
* A web app has been created and deployed using flask in azure

**3. Data Requirement.**

Whenever we are working on any project the data is completely dependent on the requirement of the problem statement. For this project, the problem statement was to create a hyper tuned classification machine learning model which can predict the determine the whether the domains are real or malicious on various parameters.

**3.1 Data Collection.**

Dataset link[: https://data.mendeley.com/datasets/72ptz43s9v/1](about:blank)

**3.2 About the dataset.**

These data consist of a collection of legitimate as well as phishing website instances. Each website is represented by the set of features which denote, whether website is legitimate or not. Data can serve as an input for machine learning process.

In this repository the two variants of the Phishing Dataset are presented.

Full variant - dataset\_full.csv

Short description of the full variant dataset:

Total number of instances: 88,647

Number of legitimate website instances (labeled as 0): 58,000

Number of phishing website instances (labeled as 1): 30,647

Total number of features: 111

Small variant - dataset\_small.csv

Short description of the small variant dataset:

Total number of instances: 58,645

Number of legitimate website instances (labeled as 0): 27,998

Number of phishing website instances (labeled as 1): 30,647

Total number of features: 111

**3.3 Dataset Characteristic.**

Below are the various attributes for the Insurance Phishing Domain Detection Dataset. Please go through the dataset\_full.csv for better clarity of the attributes when reading this document.

• attributes based on the whole URL properties presented in [Table 1](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0001),

Table 1. Dataset attributes based on URL.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_url | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_url | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_url | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_url | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_url | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_url | Number of ”=” sings | Numeric |  |
| 7 | qty\_at\_url | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_url | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_url | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_url | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_url | Number of ″′˜′signs | Numeric |  |
| 12 | qty\_comma\_url | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_url | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_url | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_url | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_url | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_url | Number of ”%” signs | Numeric |  |
| 18 | qty\_tld\_url | Top level domain character length | Numeric |  |
| 19 | length\_url | Number of characters | Numeric |  |
| 20 | email\_in\_url | Is email present | Boolean | [0, 1] |

• attributes based on the domain properties presented in [Table 2](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0002),

Table 2. Dataset attributes based on domain URL.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_domain | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_domain | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_domain | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_domain | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_domain | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_domain | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_domain | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_domain | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_domain | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_domain | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_domain | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_domain | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_domain | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_domain | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_domain | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_domain | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_domain | Number of ”%” signs | Numeric |  |
| 18 | qty\_vowels\_domain | Number of vowels | Numeric |  |
| 19 | domain\_length | Number of domain characters | Numeric |  |
| 20 | domain\_in\_ip | URL domain in IP address format | Boolean | [0, 1] |
| 21 | server\_client\_domain | ”server” or ”client” in domain | Boolean | [0, 1] |

• attributes based on the URL directory properties presented in [Table 3](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0003),

Table 3. Dataset attributes based on URL directory.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_directory | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_directory | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_directory | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_directory | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_directory | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_directory | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_directory | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_directory | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_directory | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_directory | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_directory | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_directory | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_directory | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_directory | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_directory | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_directory | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_directory | Number of ”%” signs | Numeric |  |
| 18 | directory\_length | Number of directory characters | Numeric |  |

• attributes based on the URL file properties presented in [Table 4](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0004),

Table 4. Dataset attributes based on URL file name.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_file | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_file | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_file | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_file | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_file | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_file | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_file | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_file | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_file | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_file | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_file | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_file | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_file | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_file | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_file | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_file | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_file | Number of ”%” signs | Numeric |  |
| 18 | file\_length | Number of file name characters | Numeric |  |

•attributes based on the URL parameter properties presented in [Table 5](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0005), and

Table 5. Dataset attributes based on URL parameters.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_params | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_params | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_params | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_params | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_params | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_params | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_params | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_params | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_params | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_params | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_params | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_params | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_params | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_params | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_params | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_params | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_params | Number of ”%” signs | Numeric |  |
| 18 | params\_length | Number of parameters characters | Numeric |  |
| 19 | tld\_present\_params | TLD1present in parameters | Boolean | [0, 1] |
| 20 | qty\_params | Number of parameters | Numeric |  |

•attributes based on the URL resolving data and external metrics presented in [Table 6](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0006).

Table 6. Dataset attributes based on resolving URL and external services.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | time\_response | Domain lookup time response | Numeric |  |
| 2 | domain\_spf | Domain has SPF 2 | Boolean | [0, 1] |
| 3 | asn\_ip | ASN 3 | Numeric |  |
| 4 | time\_domain\_activation | Domain activation time (in days) | Numeric |  |
| 5 | time\_domain\_expiration | Domain expiration time (in days) | Numeric |  |
| 6 | qty\_ip\_resolved | Number of resolved IPs | Numeric |  |
| 8 | qty\_nameservers | Number of resolved NS4 | Numeric |  |
| 9 | qty\_mx\_servers | Number of MX 5servers | Numeric |  |
| 10 | ttl\_hostname | Time-To-Live (TTL) | Numeric |  |
| 11 | tls\_ssl\_certificate | Has valid TLS 6/SSL 7certificate | Boolean | [0, 1] |
| 12 | qty\_redirects | Number of redirects | Numeric |  |
| 13 | url\_google\_index | Is URL indexed on Google | Boolean | [0, 1] |
| 14 | domain\_google\_index | Is domain indexed on Google | Boolean | [0, 1] |
| 15 | url\_shortened | Is URL shortened | Boolean |  |
| **16** | **phishing** | **Is phishing website** | **Boolean** | **[0, 1]** |

The first group is based on the values of the attributes on the whole URL string, while the values of the following four groups are based on the particular sub-strings, as presented in [Figure 1](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "fig0001). The last group attributes are based on the URL resolve metrics as well as on the external services such as Google

search index.

Fig. 1

Fig. 1. Separation of the whole URL string into sub-strings.

The dataset in total features 111 attributes excluding the target *phishing* attribute, which denotes whether the particular instance is legitimate (value 0) or phishing (value 1). We prepared two variations of the dataset, the one where the total number of instances is 58,645 and the balance between the target classes in more or less balanced with 30,647 instances labeled as phishing websites and 27,998 instances labeled as legitimate. The second variant of the dataset is comprised of 88,647 instances with 30,647 instances labeled as phishing and 58,000 instances labeled as legitimate, the purpose of which is to mimic the real-world situation where there are more legitimate websites present. The distribution between the classes of both dataset variants is presented in [Figure 2](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "fig0002).

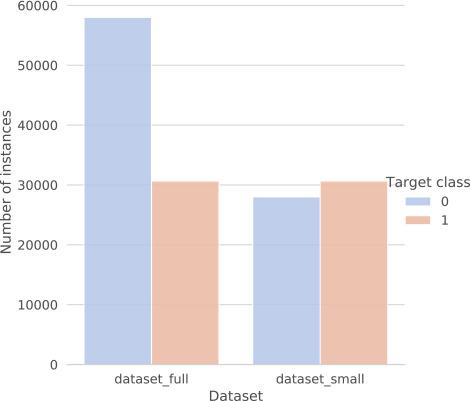
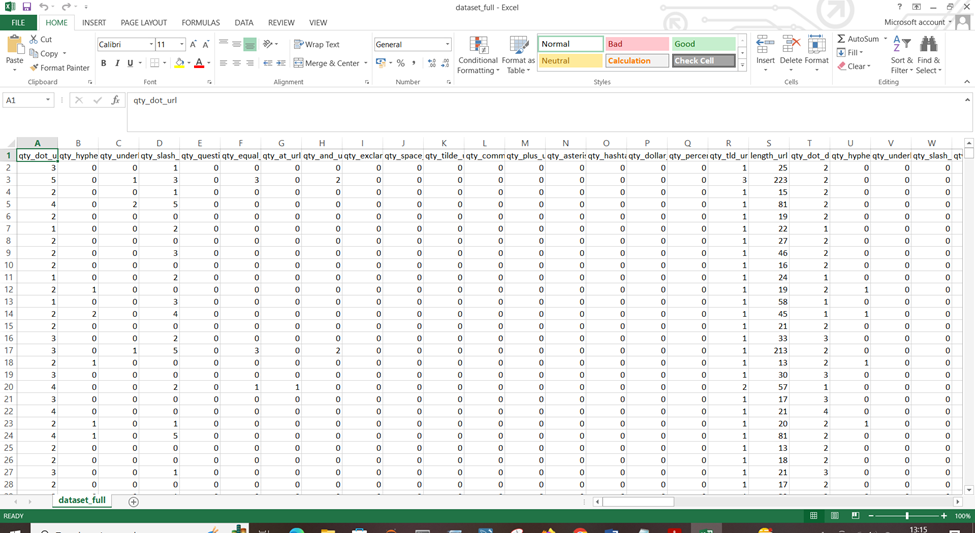


Fig. 2. The distribution between classes for both dataset variations. The *dataset\_full* denotes the larger dataset, while the *dataset\_small* denotes the smaller dataset variation. The target class *0* denotes legitimate websites while the target class *1* denotes the phishing websites.

**3.4 License.**

These data consist of a collection of legitimate as well as phishing website instances. Each website is represented by the set of features which denote, whether website is legitimate or not. Data can serve as an input for machine learning process. at <https://data.mendeley.com/datasets/72ptz43s9v/1>.



* 1. **Representation of dataset\_full.csv file**

**4. Data Preprocessing.**

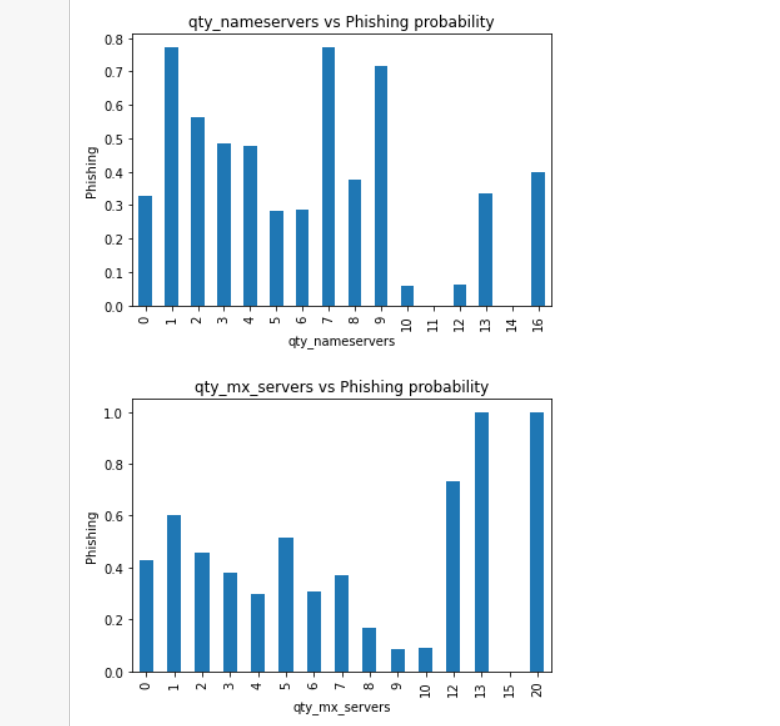
Have taken the insurance\_main.csv file as my dataset.

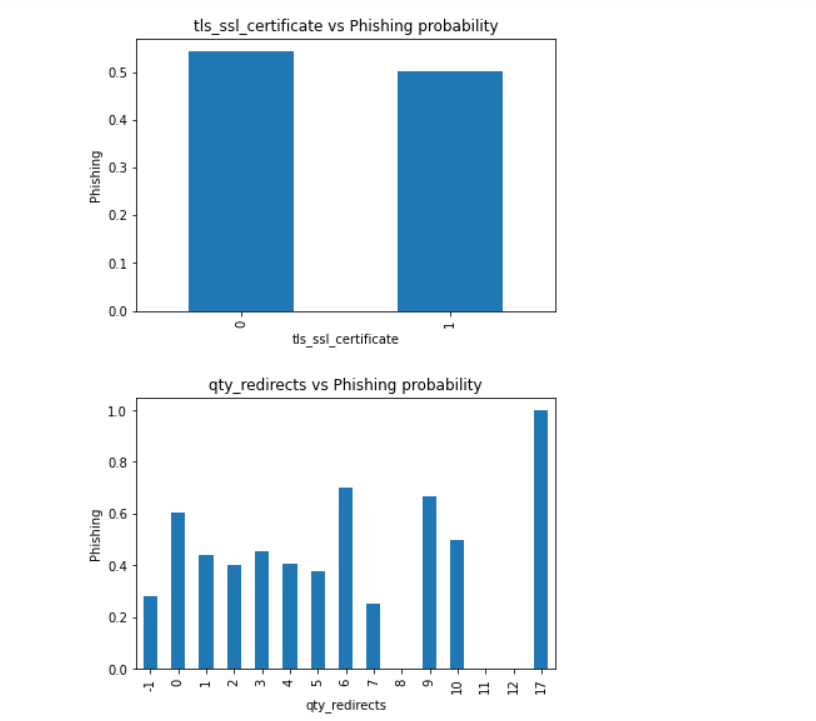
* All the necessary libraries were imported first such as Numpy, Pandas, Matplotlib, and Seaborn.
* Checking the basic profile of the dataset. To get a better understanding of the dataset.
  + Using Info method
  + Using Describe method
  + Checking for unique values of each column.
* Checking for null values, there are no null values present in our dataset.
* Used Matplotlib to plot the basic graphs which is described in the below sections separately

**1.2 Graphical Distribution of all the columns.**

**Bar plot:**

A barplot (or barchart) is one of the most common types of graphic. It shows the relationship between a numeric and a categoric variable. Each entity of the categoric variable is represented as a bar. The size of the bar represents its numeric value.





**Observations: Clearly higher the quantity of chars, higher the probability of phishing domain**

**The last 18 features, for which we will have to rely on external websites, don't look like they are contributing much except some.**

**url\_shortned: important, if it is present, probability jumps to 1**

**domain\_google\_index: unimportant, same probability for all values**

**url\_google\_index: important, higher probability, if not present**

**qty\_redirects : unimportant, looks like discrete uniform distribution**

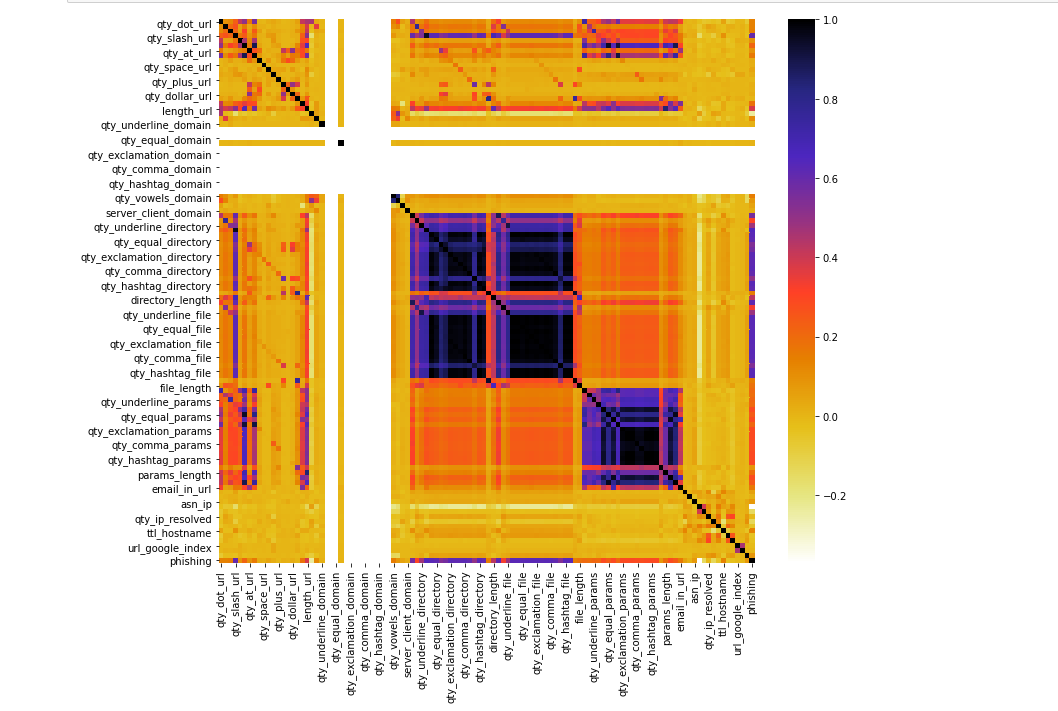
**tls\_ssl\_ceertificate: unimportant, same probability for all possible values**

**domain\_spf : unimportant, looks like discrete uniform distribution**

**email\_in\_url : very important, if present, higher probability of domain being phishing**

**Heatmap:**

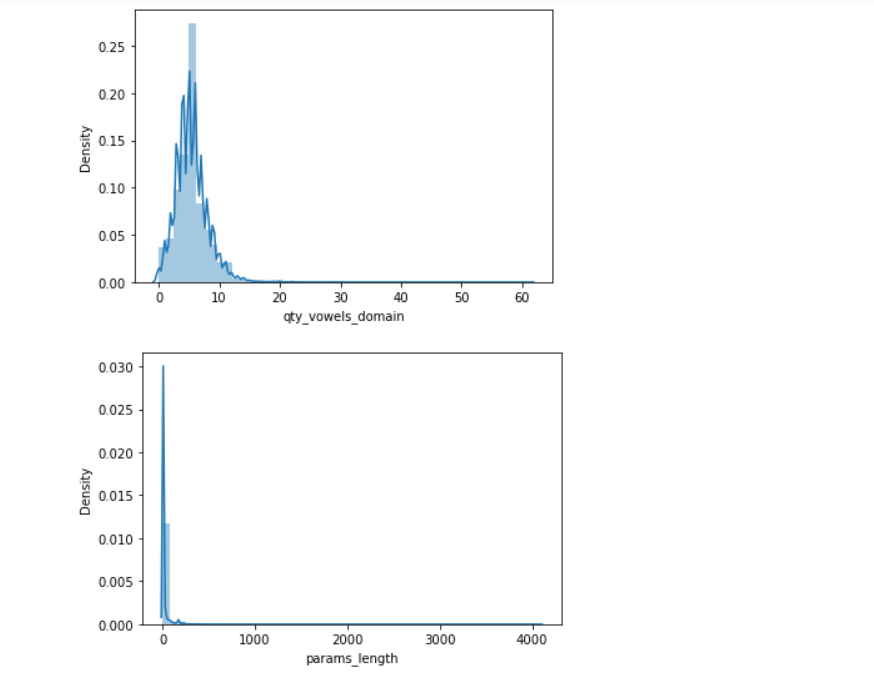
A heatmap is a graphical representation of data where the values are represented as colours. It is a useful visualization tool for understanding the relationships between different data values, and for identifying patterns and trends in the data.

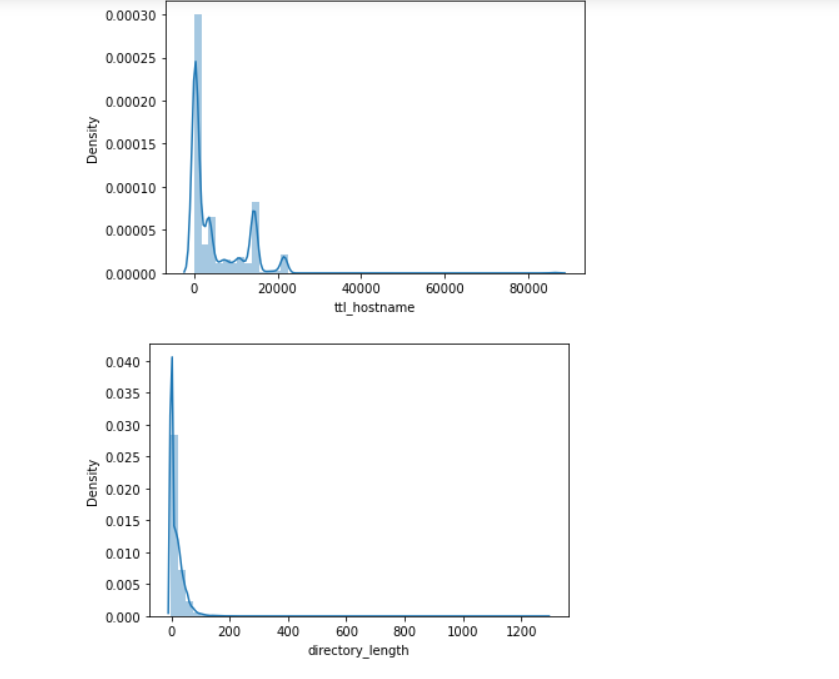


**Observation: Clearly, there are many features among which corelation is too high**

**Distplots:**

Distplots are useful for visualizing the distribution of a dataset and understanding the underlying patterns and trends in the data. They can be used to identify the presence of outliers in the data, and to compare the distribution of different datasets.



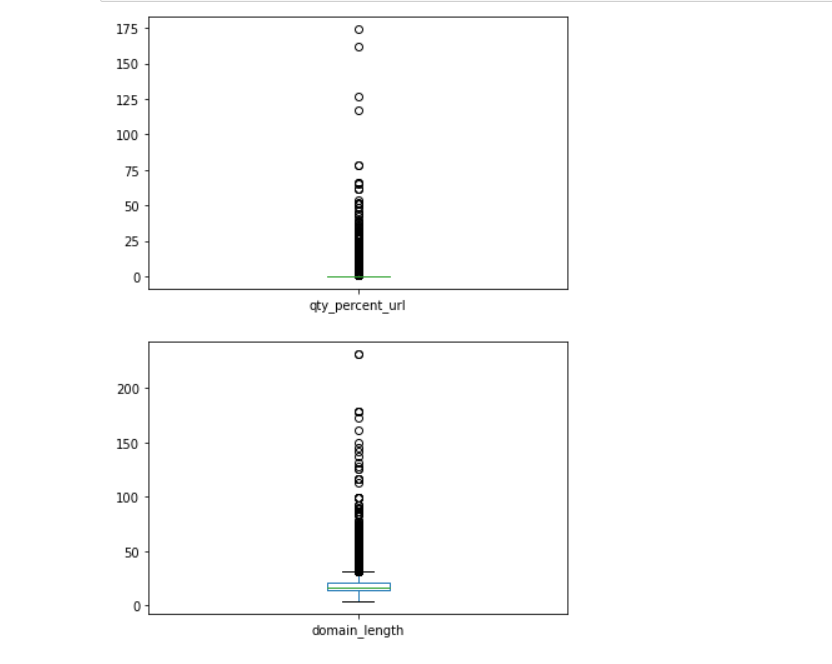
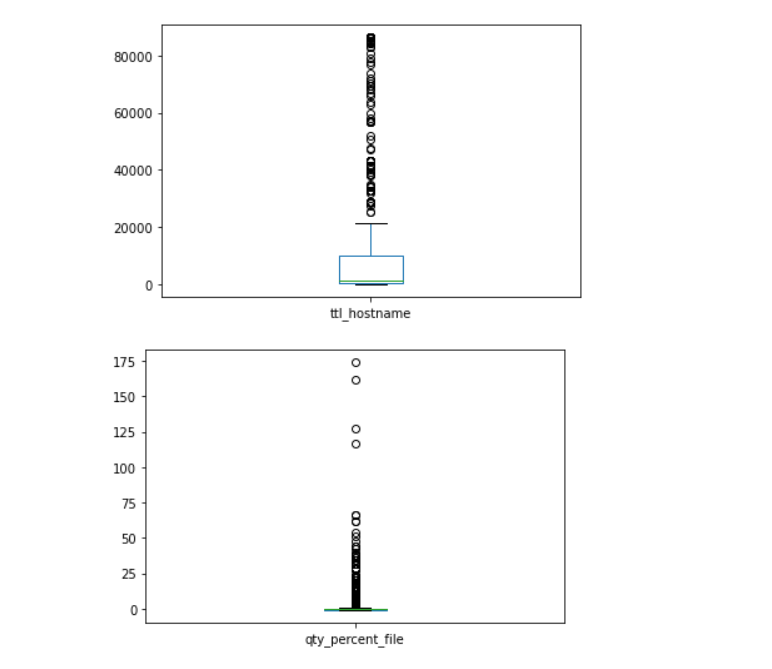


**Observation:**

* All columns are skewed

**Boxplot:**

Boxplots are useful for visualizing the distribution of a dataset and identifying patterns and trends in the data. They can be used to compare the distribution of different datasets, and to identify the presence of outliers in the data.

**Observation:**

* There are outliers present in almost all columns but as per the problem statement, these are important outliers. Also, we are getting good training and testing accuracy, so not removing outliers.

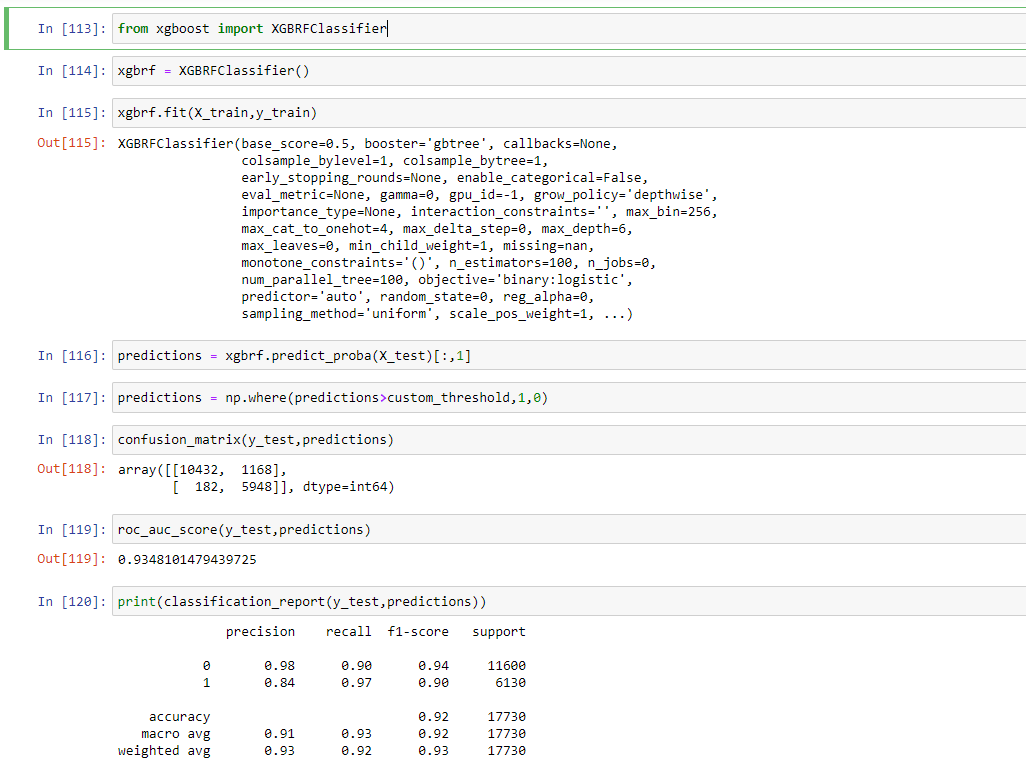
**5. Model Workflow**

**5.1 Model Selection**

* After this the data was split into 2 sets X and y. X contains all the columns except the target column in our case (pishing), y contains only the Target column.
* Using train test split we first split the dataset into X\_train,X\_test, y\_train, y\_test .
* The following libraries were imported to create Regression models.
  + from sklearn.ensemble import RandomForestClassifier
  + from sklearn.linear\_model import LogisticRegression
  + from sklearn.ensemble import AdaBoostClassifier
  + from sklearn.neighbors import KneighborsClassifier
  + from sklearn.ensemble import ExtraTreesClassifier
  + from sklearn.ensemble import GradientBoostingClassifier
  + from sklearn.naive\_bayes import GaussianNB
  + from xgboost import XGBClassifier
  + from xgboost import XGBRFClassifier

**5.2 Model Accuracy Scores.**

**On high level testing, Random Forests and XG Boost seem to be giving best results**



* The model was pickled using the Python pickle library and was ready for use into our Backend system.

**6. Life cycle of a Machine learning project**

A machine learning project typically goes through the following stages:

**1. Problem definition:** The first step is to define the problem that you want to solve using machine learning. This involves understanding the business context, determining the goals and objectives of the project, and identifying the data that you will need to train the model.

**2. Data collection and preparation**: The next step is to collect and prepare the data that you will use to train the model. This may involve scraping data from the web, collecting data from APIs, or using a pre-existing dataset. You will also need to clean and preprocess the data to get it into a suitable format for training.

**3. Exploratory data analysis:** Once you have collected and prepared the data, you will need to explore it to get a better understanding of its characteristics and any patterns that may exist. This will help you to identify any issues with the data and inform the development of the machine learning model.

**4. Preprocessing:**

Data preprocessing refers to the process of preparing data for use in training a machine learning model. This typically involves a number of steps, including:

1. Collecting data from various sources: This may include scraping data from websites, accessing databases, or collecting data from sensors or other devices.
2. Cleaning the data: This involves removing any invalid or missing data, correcting errors, and ensuring that the data is in a consistent format.
3. Normalizing or scaling the data: This involves transforming the data so that it is on the same scale, which can help improve the performance of some machine learning algorithms.
4. Splitting the data into training and test sets: This involves dividing the data into two sets, one for training the model and the other for evaluating its performance.
5. Extracting features: This involves selecting the relevant data and transforming it into a form that can be used by a machine learning algorithm.
6. Encoding categorical data: If the data includes categorical variables, they need to be encoded as numerical values in order to be used by most machine learning algorithms.

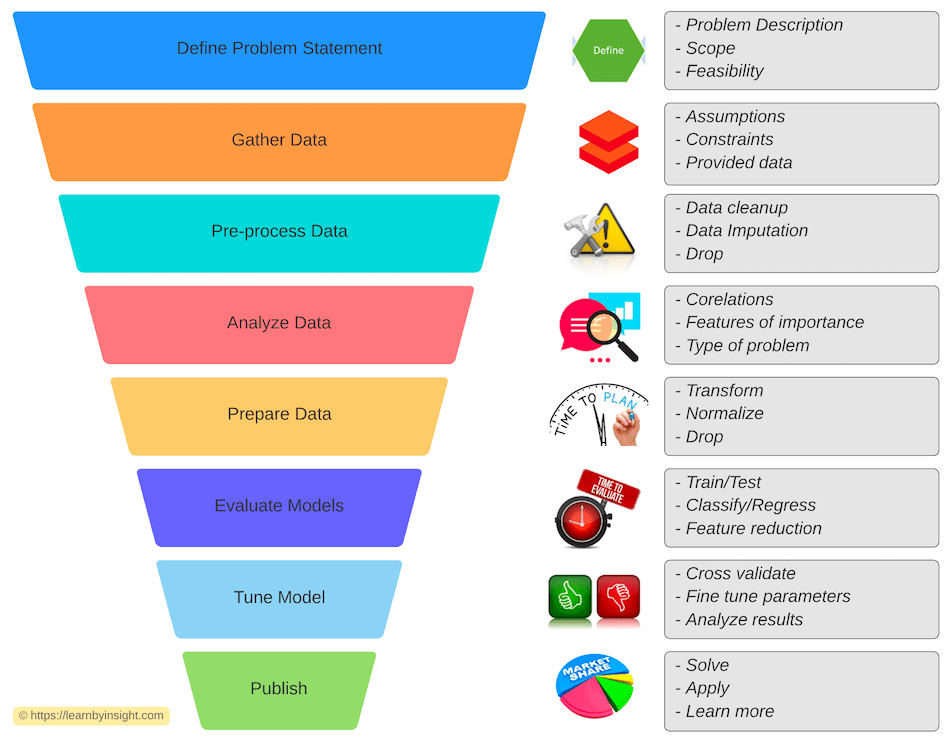
Data preprocessing is an important step in the machine learning process, as it helps ensure that the data is in a suitable form for training a model and can help improve the model's performance.

**4. Model selection and training:** The next step is to select an appropriate machine learning model and train it on the data. This will involve selecting a model type, choosing the hyperparameters for the model, and training the model on the data using an optimization algorithm.

**5. Model evaluation:** Once the model has been trained, you will need to evaluate its performance to see how well it is able to make predictions. This will involve using a variety of evaluation metrics, such as accuracy, precision, and recall, to assess the model's performance.

**6. Model deployment:** If the model performs well and meets the project goals, it can be deployed to a production environment where it can be used to make predictions in real-time. This may involve integrating the model into a web or mobile application, or using it to automate a business process.

**7. Model maintenance:** Even after a model has been deployed, it will still need to be maintained and updated over time. This may involve retraining the model on new data, fine-tuning the hyperparameters, or implementing additional features to improve its performance.



**8. Deployment**

1. This model is deployed on AWS EC2 instances. The following are the steps to deploy the model on the AWS platform:

* Create an AWS account
* Create an ECR
* Create S3 bucket
* Create an EC2 instance
* Edit security group
* Connect to an EC2 instance
* Install Docker
* Add the runner in the GitHub
* Add all the secret keys in the GitHub
* In the GitHub actions, run the continuous delivery and deployment workflow once after starting the runner in the ec2 instance
* Start the instance & locate the Docker run.sh file for to initiate the “Runner” to pick the job.
* Use Apache airflow to monitor the model and perform batch prediction

1. A web app has been created and deployed using flask in azure

**Conclusion:**

We have successfully built end-to-end ML projects using machine learning that can help predict the medical expenses of the users based on various conditions. This type of system can help users to get a whether the domain is real or fake. Along with end to end pojects, a flask webapp has been created as well to get the result based on certain inputs like url and dataset.