
PHISHING DOMAIN DETECTION

Detailed Project Report

NOVEMBER 5

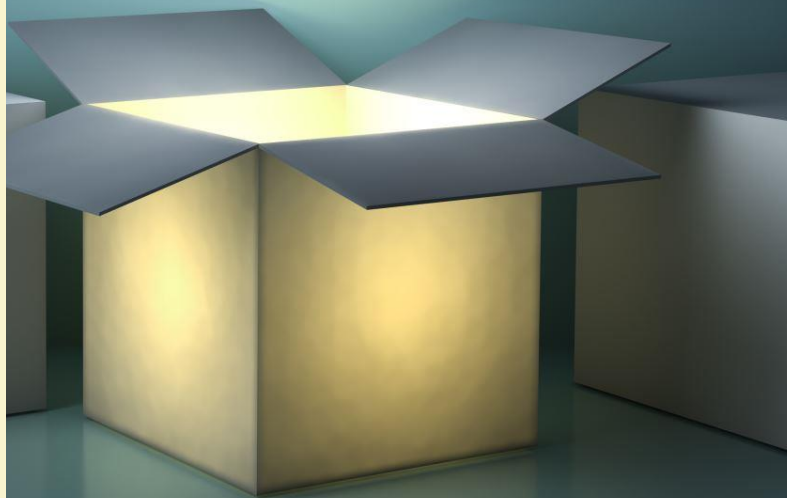
iNeuron.ai | PW Skills

Authored by: Suyash Waykar



SKILLS

iNeuron



Project Details

Problem Statement -

Phishing is a type of fraud in which an attacker impersonates a reputable company or person 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 main 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

Baseline Model: Logistic Regression since this is a classification problem.

Actual model: XG Boost

Dataset -

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.

This data consists of a collection of legitimate as well as phishing website instances. Each website is represented by a set of features which denote whether the 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

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












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.

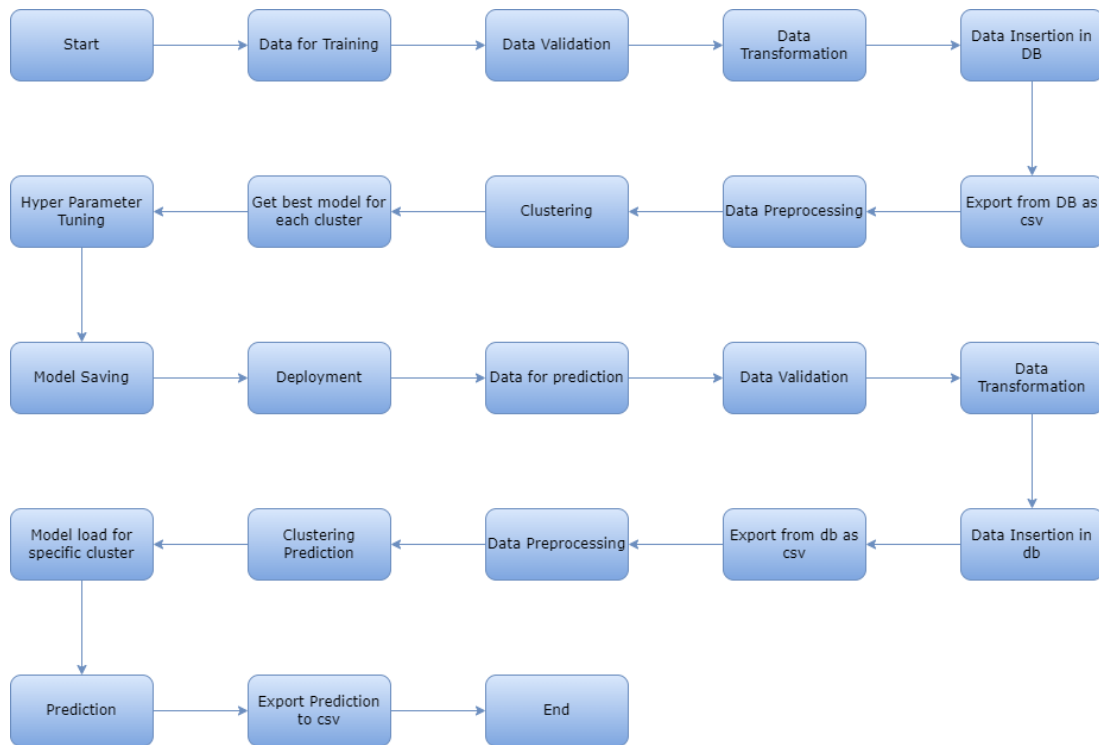
Total number of features: 11

Tools Used -

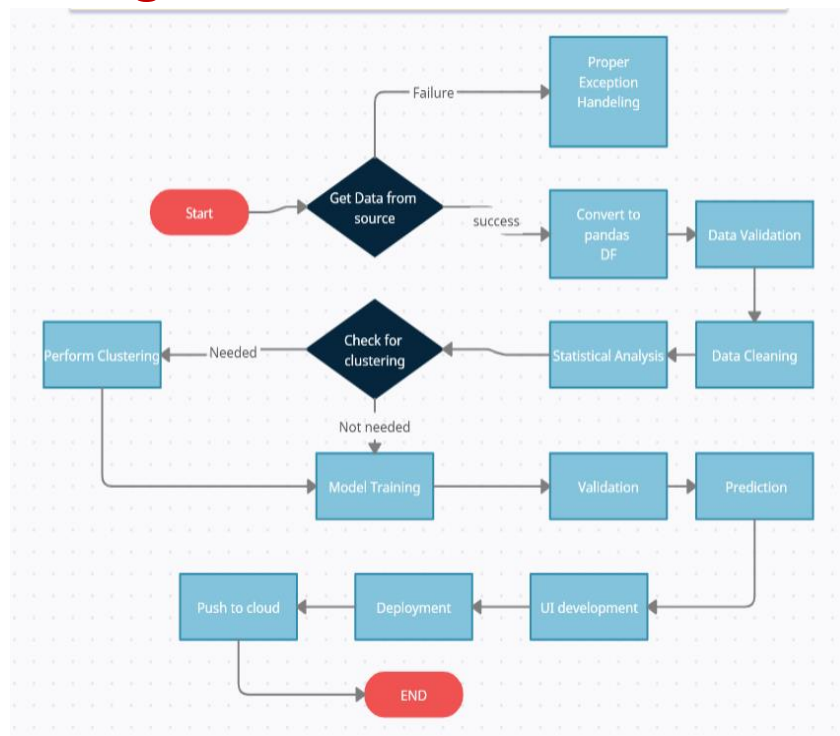
-  Pandas
-  NumPy
-  Flask
-  Cassandra
-  Matplotlib
-  Plotly
-  Scikit-Learn
-  HTML
-  GitHub
-  PyCharm
-  Python Programming



Architecture



Model Training/Validation



Data Validation and Transformation

- Name Validation - Validation of files name as per the DSA. We have created a regex pattern for validation. After it checks for date format and time format if these requirements are satisfied, we move such files to "Good_Data_Folder" else "Bad_Data_Folder".
- Number of Columns – Validation of number of columns present in the files, and if it doesn't match then the file is moved to "Bad_Data_Folder".
- Name of Columns - The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to "Bad_Data_Folder".
- Data type of columns - The data type of columns is given in the schema file. It is validated when we insert the files into Database. If the datatype is wrong, then the file is moved to "Bad_Data_Folder".
- Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad_Data_Folder".

Data Insertion in Database

- Table creation: - Table name "phishing" is created in the database for inserting the files. If the table is already present, then new files are inserted in the same table.
- Insertion of files in the table - All the files in the "Good_Data_Folder" are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table.

Model Training

- Data Export from Db: The accumulated data from db is exported in csv format for model training.
- Data Preprocessing
 - Performing EDA to get insight of data like identifying distribution, outliers, trend among data etc.
 - Check for null values in the columns. If present impute the null values.
 - Encode the categorical values with numeric values.
 - Perform Standard Scalar to scale down the values.
- Clustering –
 - KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using KneeLocator function. The idea behind clustering is to implement different algorithms on the structured data.
 - The KMeans model is trained over preprocessed data, and the model is saved for further use in prediction.
- Model Selection –

After the clusters are created, we find the best model for each cluster. By using 2 algorithms “SVM” and “XGBoost”. For each cluster both the hyper tuned algorithms are used. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

Prediction

- The testing files are shared in the batches and we perform the same Validation operations, data transformation and data insertion on them.
- The accumulated data from db is exported in csv format for prediction.
- We perform data pre-processing techniques on it.

-
- KMeans model created during training is loaded and clusters for the preprocessed data are predicted.
 - Based on the cluster number respective model is loaded and is used to predict the data for that cluster.
 - Once the prediction is done for all the clusters. The predictions are saved in csv format and shared.

Key Performance API

- + Detecting malicious websites will help in preventing cyber-attacks.
- + Comparison of accuracy of model prediction and real time data prediction.

THANKS!