Architecture Design

Phishing Domain Detection (Machine Learning)

By

Kavitha Narsapur

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 attacks are done via emails, text messages, or websites. Phishing websites have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim. Thus this project aims to detect phishing domains using machine learning models. Different models like Logistic Regression classifier, Decision Tree classifier, Random Forest classifier, K-Nearest Neighbor classifier, eXtreme Gradient Boosting classifier and Naïve Bayes classifier were built. Out of all the classifiers, Random forest classifier resulted in best accuracy of 97.16% and F1 score of 0.9717.

Introduction

Why this Architecture Design Document?

This Document summarizes the data used for phishing domain detection and also provides an overview of steps used in this project as Data preprocessing, Exploratory data analysis, Feature selection, Data balancing, Model building and Model deployment.

Scope

The documentation presents the structure of the system, such as the application architecture (layers), application flow (Navigation), and technology architecture. The document uses non-technical to mildly-technical terms which should be understandable to the administrators of the system. This software system will be a Web application. This system will be designed to detect phishing sites.

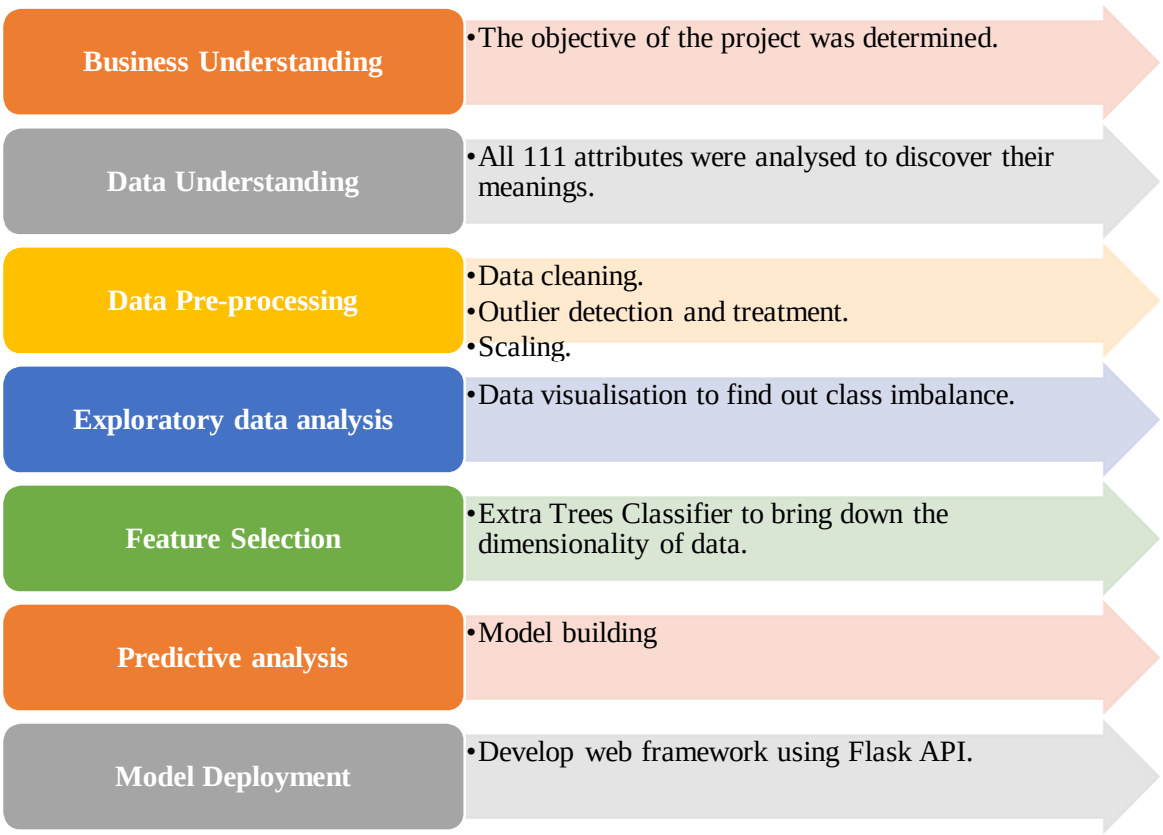
Dataset overview

This data consists 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 project the two variants of the Phishing dataset are presented.

Full variant:
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:
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

Architecture



Architecture Description

1. Business objective:

The objective of the project was to detect phishing domain sites.

2. Data Understanding:

An attempt was made to understand the meanings of all 111 variables present in the data. The data type of each variable was also determined.

3. Data pre-processing:

Data cleaning: Data was checked for the presence of missing values.

Outlier detection and treatment: Boxplots were used to detect outliers. Outliers were then treated by the method of capping and flooring using Interquartile range (IQR).

Scaling: All the attributes were brought to same scale using MinMax scaler.

4. Exploratory data analysis:

Data Imbalance detection: By means of graphical analysis, it was found that the data was imbalanced. Hence SMOTE (Synthetic Minority Oversampling Technique) was applied to balance the classes in the data.

5. Feature selection:

Extra trees classifier was used as feature selection method. Out of all 111 features, only the features selected by the Extra trees classifier were selected and used for further model building process.

6. Predictive analysis:

Train and Test data creation: Train and Test data were created by splitting the data and 70% of the data was used as train data and 30% was used as test data.

Model building: Different models like Logistic Regression classifier, Decision Tree classifier, Random Forest classifier, K-Nearest Neighbor classifier, eXtreme Gradient Boosting classifier and Naïve Bayes classifier were built.

Model evaluation: The built models were tested on the test data and evaluation metrics used were Accuracy and f1-score.

7. Model Deployment:

Best performing model was saved in Pickle format and the model was tested using Flak API on local system.

User I/O workflow

Conclusion

The web framework developed in this project can be used by users to check if a domain is legitimate or not.