Phishing Attack Domain Detection

10 Jan, 2022

Overview

This project focuses on solving the most common cyber attack called phishing. Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers.

The purpose of this LLD document is to give a detailed overview about the system.This document adds necessary detail to the current project description.We'll be using technical terms related to machine learning & software engineering here.

Goals of LLD

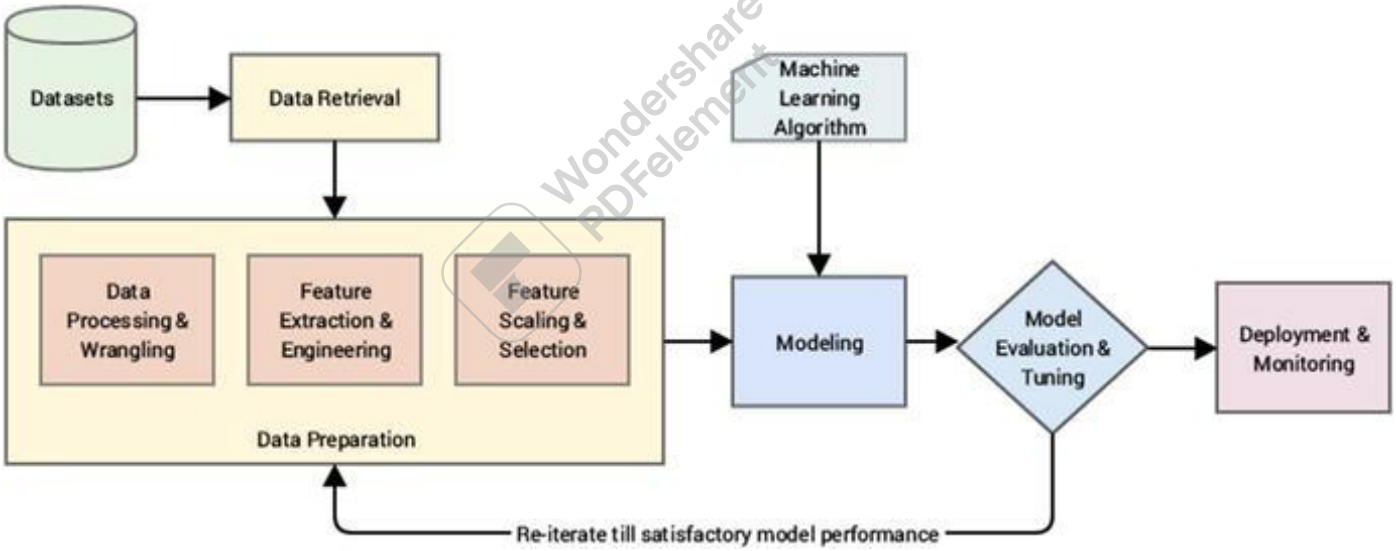
Explain Model Training Workflow : In this section we'll discuss the entire process flow for building an optimal ML model which can detect malicious URLs efficiently.

Explain Deployment process : After the model has been build,we'll have to make sure that we deploy it so that our users interact with the model,we'll discuss that in this section.

1. Model Training Workflow

In any Machine Learning project to develop and manage production-ready models we have to follow a series of processes. The project will follow the same approach as used in all ML projects and go through the following steps before deploying the model :

- Data collection
- Data preparation & Feature Engineering
- Model Training & Evaluation
- Model Deployment



We'll discuss each step in more detail in regards to our problem statement.

1.1 Data Collection

Data collection is one of the most important steps in any machine learning workflow as the efficiency of our production model is **directly proportional** to the quality of our training dataset.

In order to train a production ready classification model which can optimally predict malicious phishing URLs, we need to train the model on similar datasets.

Data sources : Public datasets, Kaggle datasets etc

Some sample datasets we'll be using for our problem are listed below :

- URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB
- Datasets for phishing websites detection - ScienceDirect
- Detect Malicious URL using ML | Kaggle

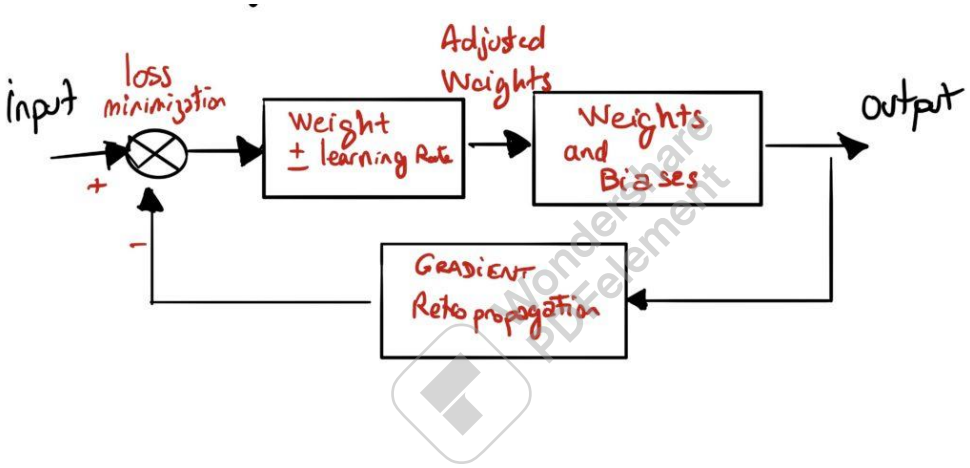
Some properties that make a good dataset for our problem :

- The dataset must contain an equal amount of positive and negative samples.
- The data must contain diverse features and characteristics with less sparsity among features.
- The data must contain both numerical and categorical features in balance.

1.2 Malicious URL features

The problem of classifying malicious phishing urls can be categorized as a ‘Binary classification’ problem which consists of 2 target classes (malicious and non-malicious urls).

Such classification is only possible when the incoming data (Urls in our case) have a certain kind of characteristics which can be converted into numerical format and are **‘Learnable’** by machine learning systems.



Our main objective for training any machine learning model is to make the model somehow learn these features of URLs,so that if we give it any input URL it can analyze its features and make an accurate prediction.

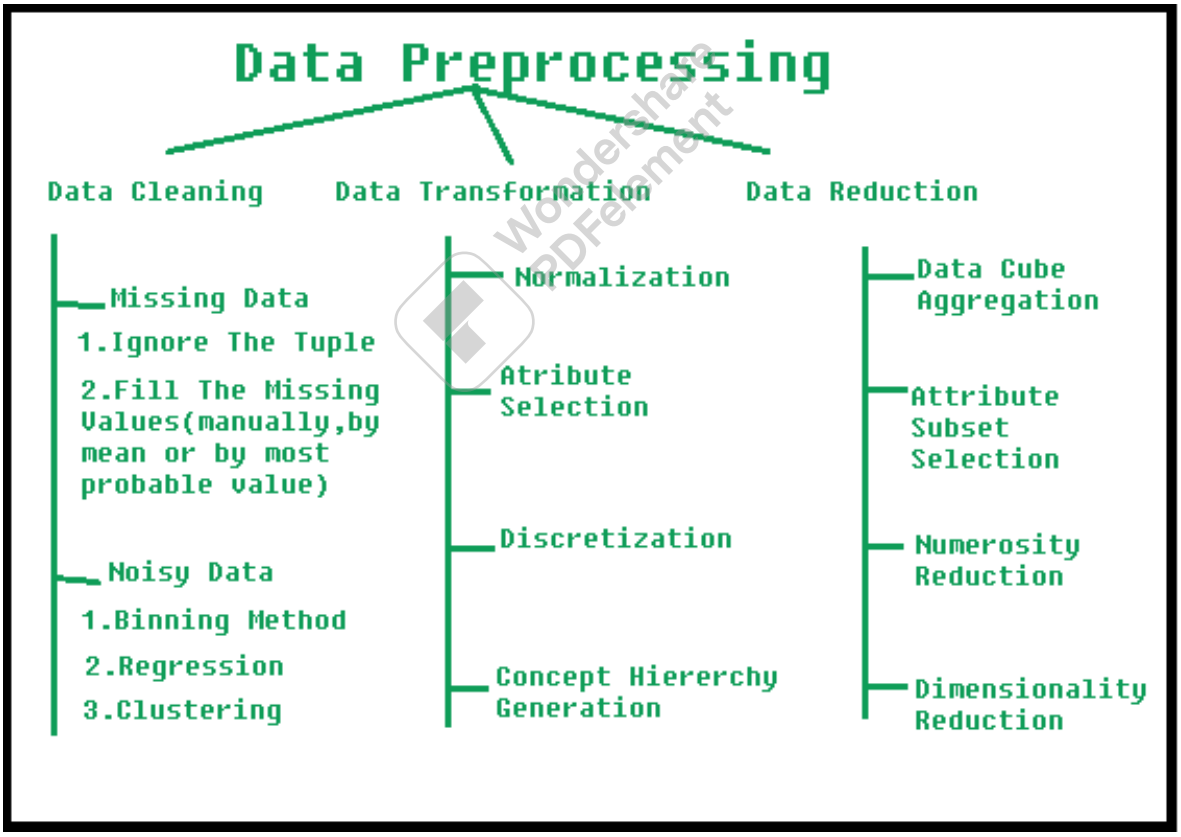
Some of the URL features we captured from the datasets we collected are listed below :

- Length Of Url
- Count Of Digits
- Count Of Number Of Directories
- Use of Shortening URL or not

1.3 Data Preparation & Feature Engineering

Most machine learning algorithms require data to be formatted in a very specific way, so datasets generally require some preparation before they can yield useful insights.

Some datasets have values that are missing, invalid, or otherwise difficult for an algorithm to process. If data is missing, the algorithm can't use it. If data is invalid, the algorithm produces less accurate or even misleading outcomes.



Below is the list of some operations we must do on our dataset before considering using it for model training :

- Handle missing/null values in dataset
- Encode categorical values using One-hot encoding method
- Handle outliers values in dataset
- Normalize the dataset values
- Handle skewness in features
- Feature selection

1.4 Tools for Automated EDA

Most of the times data exploration and processing is done by humans,but including automated tools for such tasks can help us find patterns which cannot be easily seen and it also speeds up the data processing pipeline.

Below are some tools you can incorporate in your data processing operations :

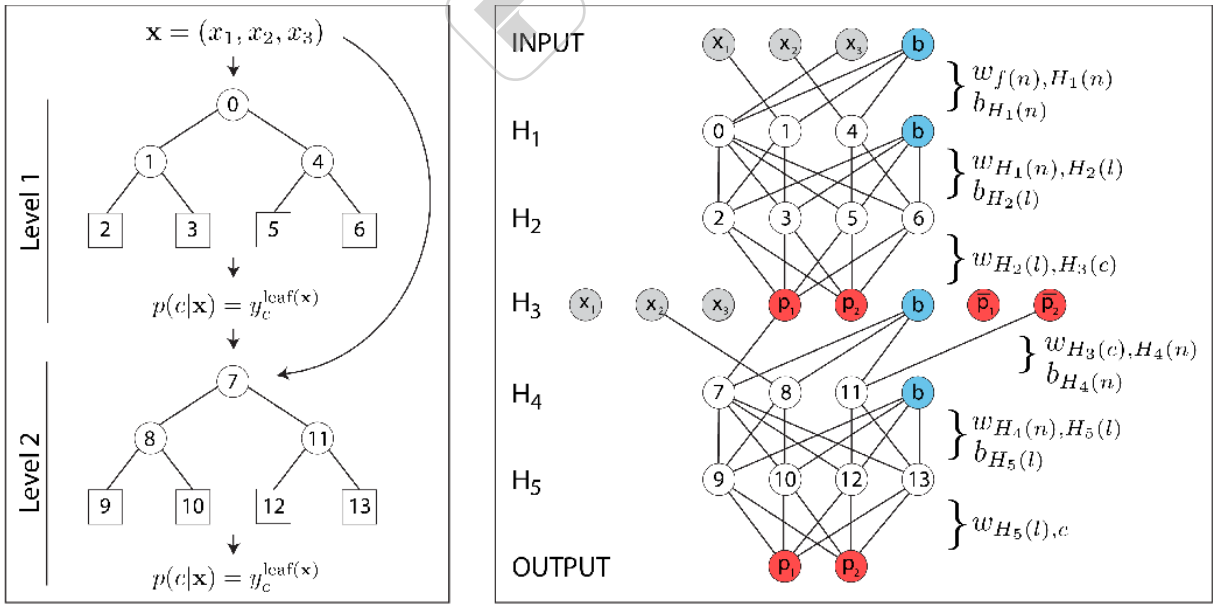
1.5 Training the model

After the data has been brought to a suitable format after processing and feature engineering, we can start with training machine learning models.

Depending on the type of data given, the accuracy and performance of various machine learning algorithms can also vary, therefore we must train multiple algorithms and gauge their performance to choose the one model which gives the best performance.

The machine learning models considered to train the dataset in this project are the following :

- Decision Trees
- Random Forest
- Neural Network (Multilayer Perceptron)

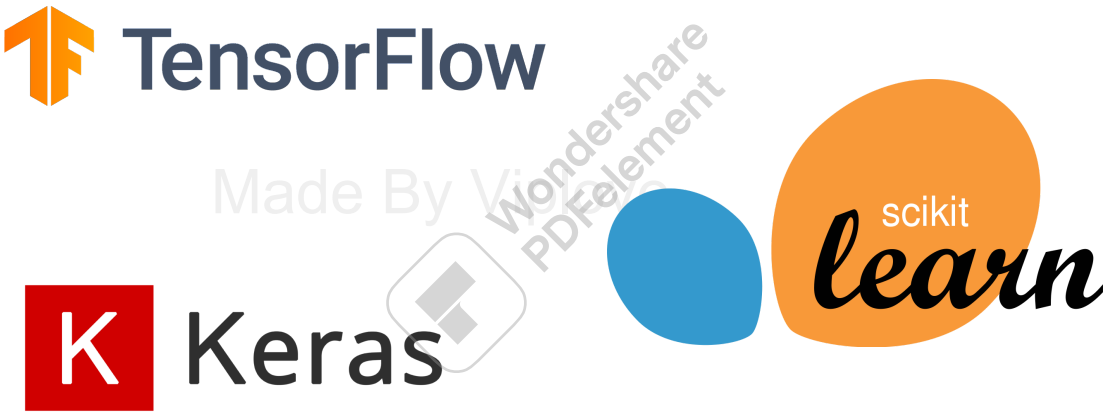


(a) (b)

1.6 Machine Learning Tools

Building machine learning models is a bit different than traditional programming. Here we use a lot of mathematical concepts and often it is very difficult and time-consuming to manually write software for machine learning models. Hence we use software frameworks which provide an abstract interface and make it easy to build machine learning models.

Below are some of the machine learning frameworks we'll be using in our project :

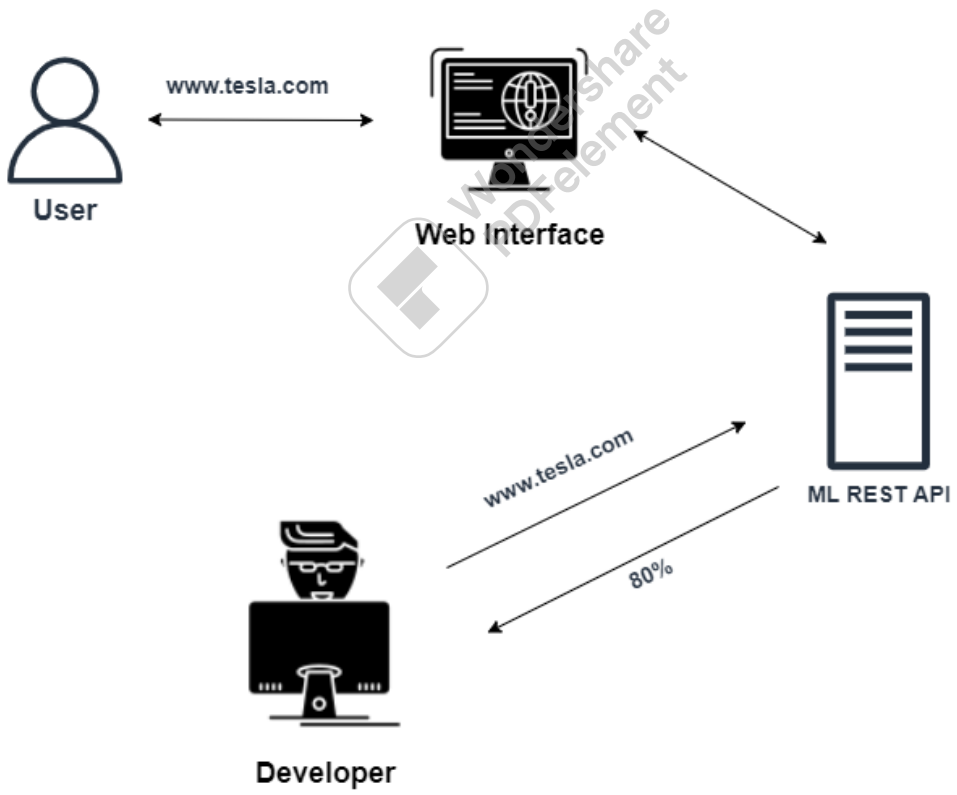


Also, training machine learning models can be computationally very expensive, hence below are some cloud services which provide free GPU :

2. Model Deployment

Deployment is a key step in an organization gaining operational value from machine learning. The simplest way to deploy a machine learning model is to create a Web Microservice like an REST API. But to make it more acessable,such that even Non-technical people can interact with the model,we'll also be creating a Web Interface for it.

Below diagram describes the architecture of the system after deployment,it describes how both the developer and a user exchange data with the ML model.



2.1 Deployment Tools

Below are the following tools we'll be using for this project :

- ML Rest API - Flask or FastAPI Python Libraries
- Web Interface - ReactJS Javascript Web Framework
- Cloud Service - Heroku