PHISHING PREDICTION

Predict phishing website by ML

ABSTRACT

The basic idea of this ML model to provide a safe browsing environment for the IT industry employees and other people also, all industries are moving to online for scaling which lead to increase in websites as well as the phishing websites.

REVISION NUMBER – 1.0

Authored By:

Nihar Ranjan Samal

M.Sc. Data Science

Content

Document Version Control ……2

Abstract ……3

1.Introduction ……4

1.1. What Is LLD Document? ……4

1.2. Scope ……4

2. Architecture ……4

3. Architecture Design ……5

3.1. Data Collection ……5

3.2. Data Description ……5

3.3. Data Preprocessing ……5

3.4. Feature Selection ……5

3.5. Modeling ……6

3.6. Data From User ……6

3.7. Data Validation ……6

3.8. Rendering Result ……6

4. Deployment ……6

4.1. Unit Test Case ……7

Document Version Control

Date	Version	Description	Author
21/09/2022	1.0	Abstract, Introduction, General Description, Design Flow	Nihar Ranjan Samal

Abstract

The basic idea of this ML model to provide a safe browsing environment for the IT industry employees and other people also, all industries are moving to online for scaling which lead to increase in websites as well as the phishing websites. Phishing is popular among attackers because it's easier to trick (social engineering) someone to click on a link. It is very important to know if a website is fake or legitimate, mistake from employee may lead to a saviour lose to the company.

1.Introduction

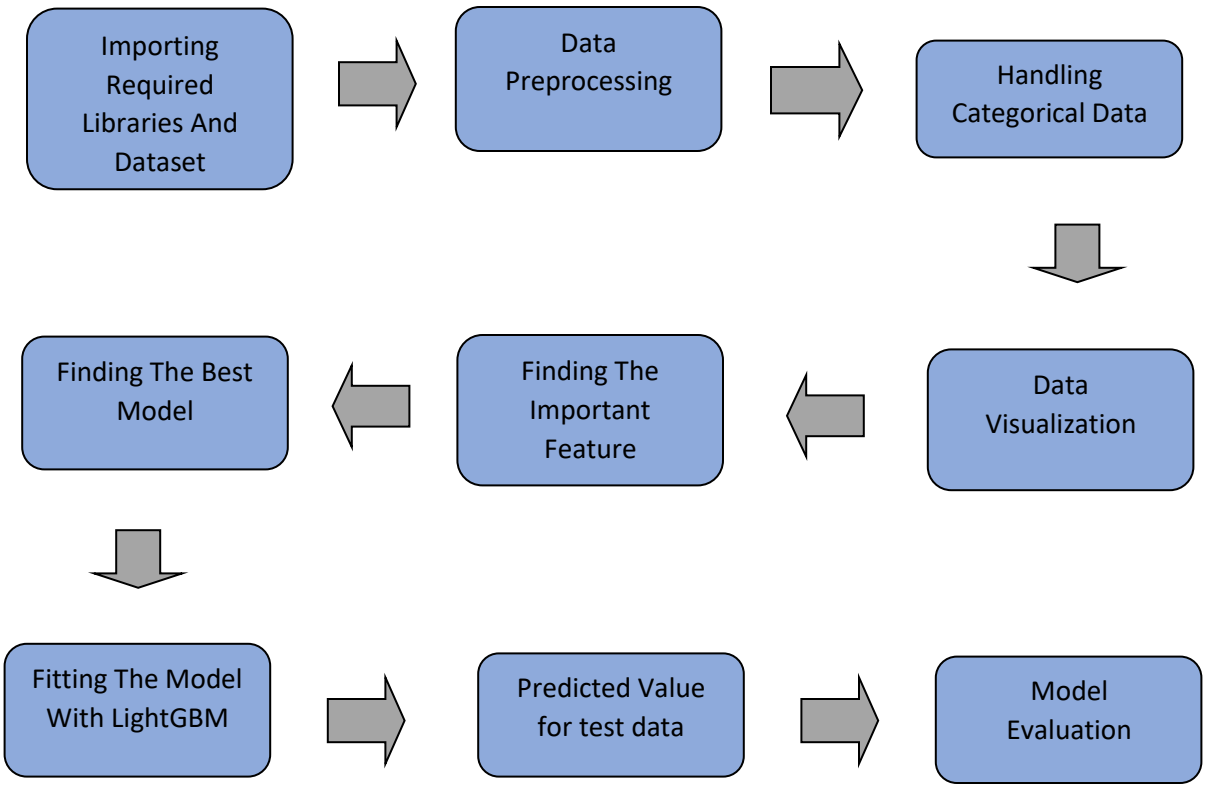
1.1 Why this LLD Document?

The main goal of the LLD document is to give the internal logic design of actual code implementation and supply the outline of the machine learning model and its implementation. Additionally, it provides the description how our project will designed end-to-end.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step_by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture



3. Architecture Design

This project is designed to make an interface for the User to predict the rating of restaurant.

3.1. Data Collection

The data for this project is collected from the Kaggle Dataset, the URL for the dataset is https://data.mendeley.com/datasets/72ptz43s9v/1

3.2. Data Description

The dataset contains 112 variables. The dataset contains more than 8800 records and total size of the dataset is approximately 40 MB. There are 112 entries from qty_dot_url to phishing and of data type 1 float64 and 111 int64.

3.3. Data Pre-processing

- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Checking the distribution of the columns to interpret its importance.
- Now, the info is prepared to train a Machine Learning Model.

3.4. Feature Selection

The features are selected on ranking by LightGBM.

3.5. Model Creation

The Pre - processed info is now envisioned and drawn insights helps us to select the feature that improves the accuracy of the model. The info is randomly used for modelling with different machine learning algorithms to create a model to predict the Phishing website. After performing on different algorithms, we use Random Forest Regression to create a model and then also perform Hyperparameter Tuning to improve the accuracy of the model.

3.6. Data From User

The data from the user is retrieved from the created HTML web page.

3.7. Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent to the prepared model for the prediction.

4. Deployment

The tested model is then deployed to AWS. So the user can access from any internet device.

4.1. Unit Test Case

Test case description	Pre-Requisites	Expected Results
Verify whether the Webpage is accessible to the User or not.	Webpage URL should be defined.	Webpage should be accessible to the User.
Verify whether the Webpage is completely loads for the User or not	1. Webpage URL is accessible. 2. Webpage is deployed.	The Webpage should be completely loads for the User when it is access
Verify whether the User is able to enter data in input fields or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to enter data in input fields.
Verify whether the User is able to submit details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to submit details to process.
Verify whether the User gets recommended results on submitting the details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable	The User gets recommended results on submitting the details.