iNeuron

PHISHING DOMAIN DETECTION
(Machine Learning)

LOW LEVEL DESIGN

Project Members:

1. Rishabh
2. Shivansh Srivastava
3. Ashish Diwakar

iNeuron.ai

Mar 2023

INTRODUCTION

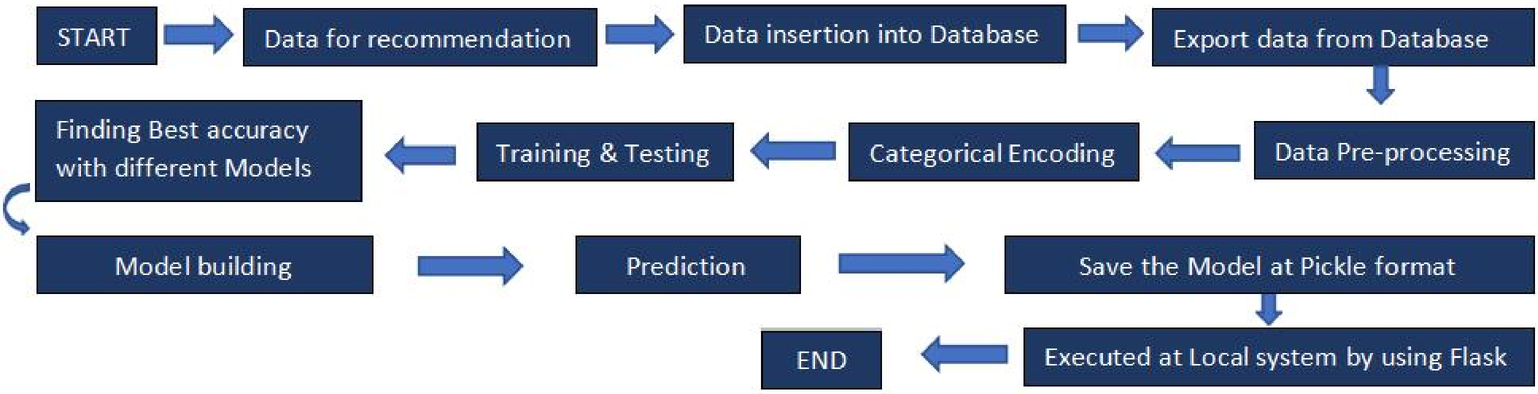
What is Low-Level design document -

The goal of LLD or a low-level design document (LLD) is to give the internal logical design of the actual program code for Mushroom classification. LLD describes the class diagrams with the methods and relation between features and level column. It describes the modules so that the programmer can directly code the program from the document.

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 requirements analysis and then refined during data design work.

ARCHITECTURE -



START
Data for recommendation
Data insertion into Database
Export data from Database
Data Pre-processing
Categorical Encoding
Training & Testing
Finding Best accuracy with different Models
Model building
Prediction
Save the Model at Pickle format
Executed at Local system by using Flask
END

ARCHITECTURE DESCRIPTION -

1. DATA DESCRIPTION-

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.

The dataset had 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

2. DATA INSERTION INTO DATABASE -

a) Database creation and connection-Create a database with name passed. If the database is already created, open the connection to the database.

b) Table creation in the database.

c) Insertion of files in the table.

3. EXPORT DATA FROM DATABASE -

Data export from database-The data in a stored database is exported as a CSV file to be used for data Pre-Processing and model training.

4. DATA PRE-PROCESSING -

Convert the domain names to a structured format: The domain names in the dataset can be converted to a structured format that can be easily used by the machine learning model. For example, the domain names can be split into subdomains and the top-level domain (TLD) using the dot symbol.

Feature engineering: Create new features that may be relevant for detecting phishing domains. For example, the number of subdomains, the length of the domain name, the presence of certain characters such as hyphens and underscores, and the TLD can all be potential features.

Standardization: If the features have different scales or units, standardization can be applied to make them comparable. For example, the number of subdomains can be standardized by dividing by the maximum number of subdomains in the dataset.

Encoding: Convert categorical features such as the TLD into numerical values using techniques such as one-hot encoding.

Splitting the dataset: Finally, the dataset can be split into a training set and a test set for model training and evaluation.

5. ML ALGORITHM -

All the ML algorithm is used to do classification and found the best model from that.

6. CATEGORICAL ENCODING -

All the datasets available on dataset was not on numerical, so that has been convertedon terms, which will be easy to do model building

7. TRAINING AND TESTING DATASET -

As here 80 % of dataset has been trained and 20% of dataset has been tested.

8. FINDING ACCURACY WITH DIFFERENT MODEL -

All the supervised machine learning algorithm were used to classify the output such as Logistic regression, Decision tree, Random forest, Gradient boosting classifier, XG Boost Classifier etc. found accuracy with every models.

9. MODEL BUILDING -

After checking accuracy with different model, model building was created with the best accuracy and saved the model in pickle format.

10. WEB FRAMEWORK -

By using flask API on the local system it been tested.

Conclusion -

This is a web-based application. We have used Flask for the user interface. We can domain name and check if the link entered is safe to use or is malicious.