Architecture Design

Credit Card Default Prediction

Written By	Prajjwal Sule
Version	1.0
Date	13-June-2023

Document Control:

Version	Date	Author	Comments
1.0	13-06-2023	Prajjwal Sule	Introduction and Architecture Define

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Content

1. Introduction

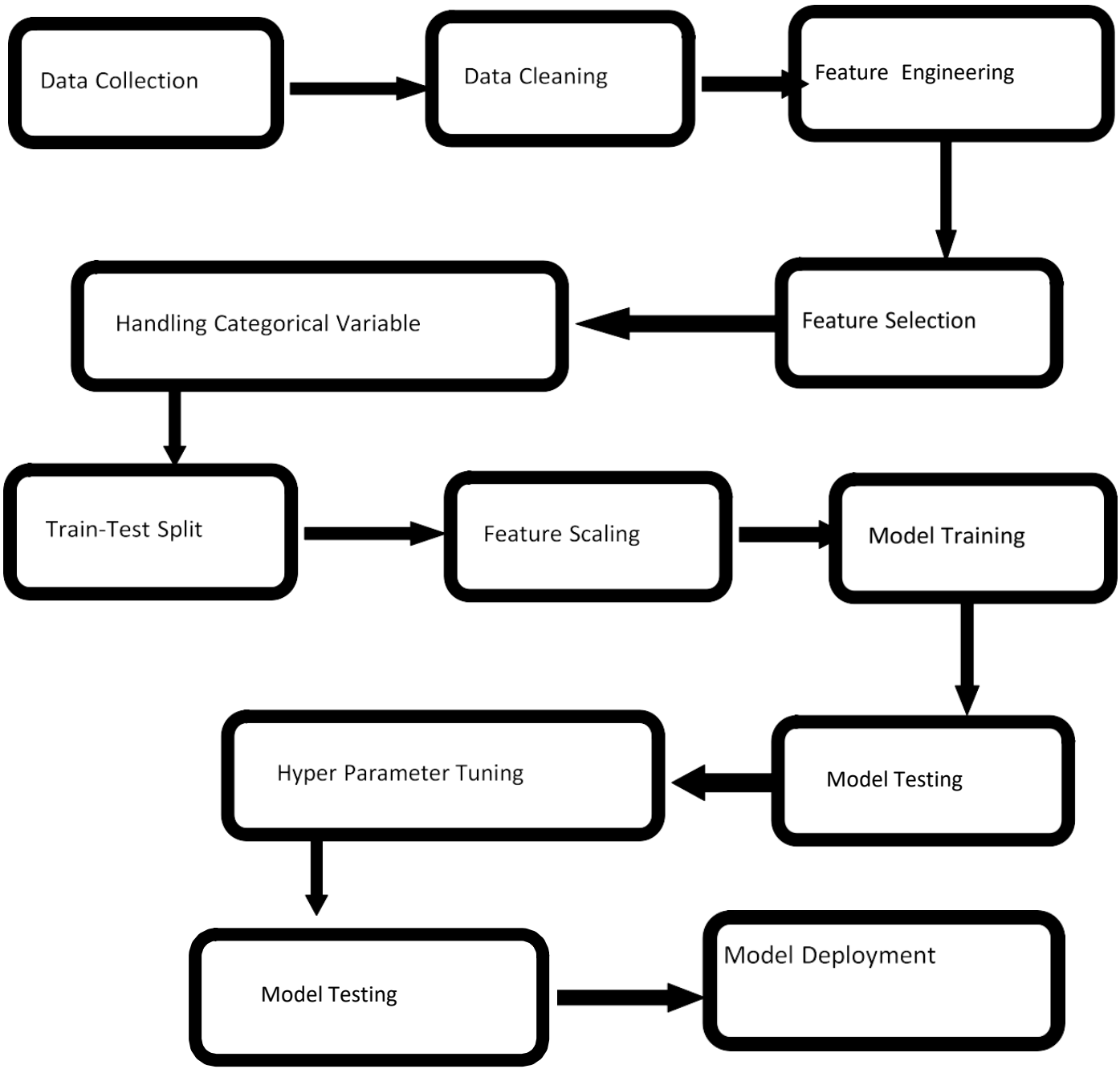
1.1 Why this Architecture Design Document?

The purpose of this document is to provide a detailed architecture design of the Credit Card Default Prediction Project by focusing on four key quality attributes:
usability, **availability**, **maintainability**, **testability**.

This document will address the background for this project, and the architecturally significant function requirements. The intension of this document is to help the development team to determine how the system will be structured at the highest level.

Finally, the project coach can use this document to validate that the development team is meeting the agreed-upon requirements during the evaluation of the team's efforts.

2. Architecture



3. Architecture Description

3.1 Data Description

We have 30000 Dataset row with column data includes 'ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_1', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'Default_payment_next_month'. These is given in the comma separated value format (.csv).

3.2 Data Cleaning/ Data Transformation

In the Cleaning process, we have cleaned up all the data because data is present in very bad format which was cannot recognized by machine. So, data Cleaning is done very first by data validation methods.

3.3 Exploratory Data Analysis

In EDA we have seen various insights from the data so we have selected which column is most important and dropped some of the columns by observing them spearman rank co-relation and plotting their heatmap from seaborn library also we done null value managed in an efficient manner and also implemented categorical to numerical transfer of column method here.

3.4 Event Log

The system should log every event so that the user will know what process is running internally. Logging is implemented using python's standard logging library.
Initial step-by-step description: -

- The system should be able to log each and every system flow.
- System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory to do.

3.5 Data Insertion into Database

Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database. Table creation in the database. Insertion of files in the table

3.6 Export Data into 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.

3.7 Data Pre-processing

Data Pre-processing steps we could use are Null value handling, Categorical to Numerical Transformation of columns, Splitting Data into Dependent and Independent Features, remove those columns which are does not participate in model building Processes, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.8 Model Creation/Building

After cleaning the data and completing the feature Engineering/ data Per-processing. we have done splitted data in the train data and test data using method build in pre-processing file and implemented various Classification Algorithm like Random Forest Classifier and XgBoost Classifier also calculated their accuracies on test data and train data.

3.9 Hyperparameter Tuning

In hyperparameter tuning we have implemented randomized search cv or grid search cv and from that we also implemented cross validation techniques for that. From that we have choose best parameters according to hyperparameter tunning and best score from their accuracies so we got 78% accuracy in our random forest classifier after hyper parameter tuning.

3.10 Model Dumping

After comparing all accuracies and checked all ROC, AUC curve accuracy we have choose hyper parameterized random forest classifier and Random Forest Classifier as our best model by their results so we have dumped this model in a pickle file format.

3.11 Data from User

Here we will collect user's requirement to predict whether a customer is a credible customer or not. We tool input as a Limit_Balance, Gender, Education, Marriage, Age, Pay_1, Pay_2, Pay_3, TotalBill_Amount_Till_6Month, TotalPay_Amount_Till_6Month.

3.12 Data Validation

Here Data Validation will be done, given by the user.

3.13 Model Call for Specific input

Based on the User input will be throwing to the backend in the variable format then it converted into Pandas data frame then we are loading our pickle file in the backend and predicting whether Customer is Credible (0) or not (1) as an output and sending to our Streamlit app.

3.14 User Interface

In Frontend creation we have made a user interactive page where user can enter their input values to our application. In these frontend page we have made with the help of Streamlit.

3.15 Deployment

We will be deploying the model with the help of Heroku cloud platforms.