ARCHITECHTURE DESIGN

CREDIT CARD DEFAULT PREDICTION

Written By / Author	Nishika Patel Vaibhav Joshi
Document Version	V2.0
Last Revised Date	04/07/2023

Document Version Control

Date Issued	Version	Description	Author
03/07/2023	V1.0	First Version of Complete HLD	VAIBHAV JOSHI
04/07/2023	V2.0	Final Version of Complete HLD	NISHIKA PATEL

Approval Status:

Version	Review Data	Reviewed By	Approved By	Comments
V1.0				

Contents

1. Introduction

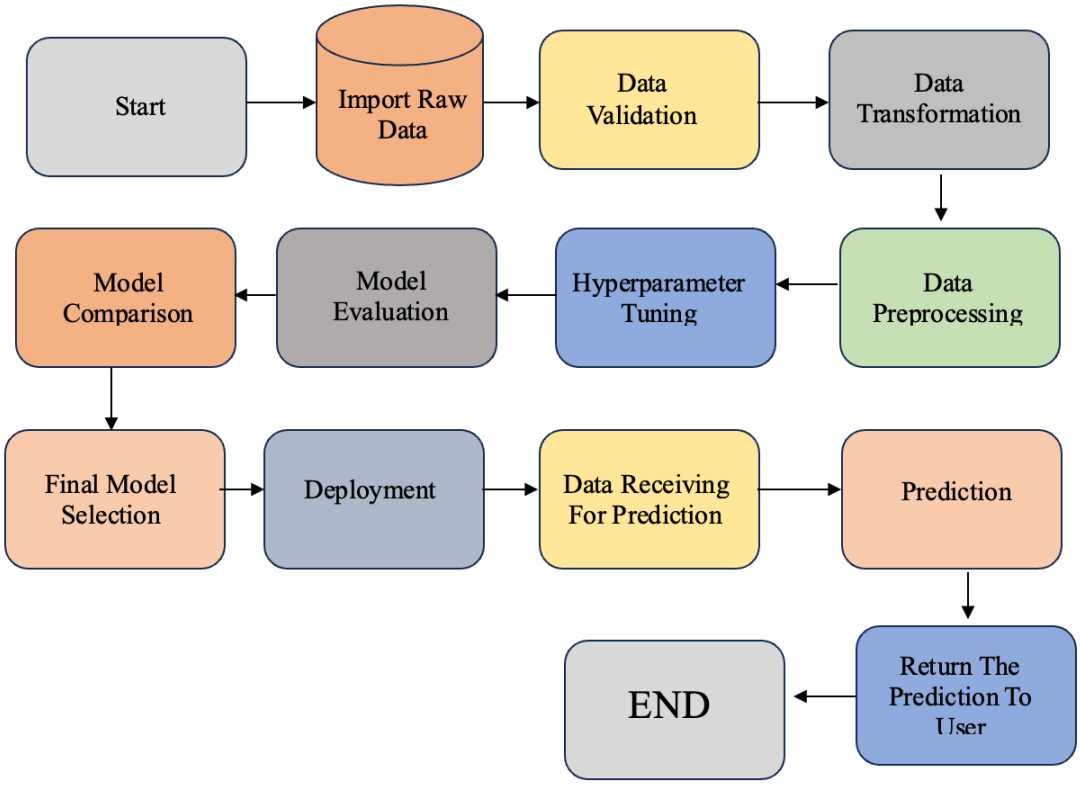
1.1 Why this Architecture design document?

The purpose of this document is to provide a detailed architecture design of the ANALYZING SWIGGY Project by focusing on each of the attributes of our architecture. This document will address the background of 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.

1.2 Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and t hen refined during architectural design work.

2. Architecture



2.1 Architecture Description –

2.1.1 Data Analysis:

We used the Seaborn and Matplotlib libraries to create different graphs to better comprehend the data and information distribution. We proceeded with the visualisation and analysis because there were no null values in the data.

We analysed the data utilising visualisation for each unique feature and noted the significant key elements that could influence the final forecasts.

2.1.2 Exploratory Data Analysis (EDA) –

- "Exploratory Data Analysis" (EDA) is a "Data Exploration" step in the Data Analysis Process, where a number of techniques are used to better understand the dataset being used.
- Understanding the Dataset can refer to a number of things including but not limited to…
- Extracting Important "Variables".
- Identifying "Outliers", "Missing Values", or "Human Error".
- Understanding the Relationships between variables.
- Ultimately, maximizing our insights of a dataset and minimizing potential "Error" that may occur later in the process.
- In other words, it will gives you a better Understanding of the "Variables" and the "Relationships" between them.
- Here, we make use of dataprep module to automate our EDA process.
- It provides the following information:
- **Overview**: detect the types of columns in a DataFrame.
- **Variables**: variable type, unique values, distinct count, missing values Quartile statistics like minimum value, Q1, median, Q3, maximum, range, interquartile range Descriptive statistics like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness.
- **Correlations**: highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices
- **Missing Values**: Bar Chart, Heatmap and spectrum of missing values.

2.1.3 Data Pre-processing:

Important libraries like Seaborn, Matplotlib, Pandas, and others had to be imported as part of this. The identical dataset from Kaggle that was stated above was imported. Additionally, we looked to see whether any columns had a standard deviation of zero. Additionally, the pre-processing checks for the presence of null values; the dataset has no null values, therefore we continue using it in its current state.

2.1.4 Train Test Split:

This library was imported from sklearn in order to split the final dataset into two halves, with 70% of the data being used to train the model and the remaining 30% being used to predict the same.

2.1.5 Model Selection:

Our datasets will be trained using four different algorithms: XGBoost Classifier, Random Forest, Decision Tree and Logistic Regression. Any model we employed is saved in the model's comparison table.

2.1.6 Prediction:

We used the user's input to inform our predictions. Then, after loading the relevant model that had been previously saved during training, we made a prediction. The pickle library, which saves the model in binary mode, was used to save the model.

2.1.7 Deploy:

This one has been set up on a local host, where a web page has been developed to receive input from the user and provide a prediction.

Here is a picture of the same thing.