Low Level Design

Travel Package Purchase Prediction

Revision No: 1.0
Last Date of Revision: 23/07/23

Manasi Khillare

Naresh K

Document Version Control

Date Issued	Version	Description	Author
23/07/23	1.0	First Version of Complete LLD	Manasi Khillare Naresh K

Contents

1.Introduction

1. What is low level design document

2. Scope

2. Architecture

3. Architecture Description

1.Data Description

2.Data Gathering

3.Data Pre-processing

4.Feature Engineering

5.Parameter Tuning

6.Model Building

7.Deployment

4. Unit Test Cases

1.Introduction

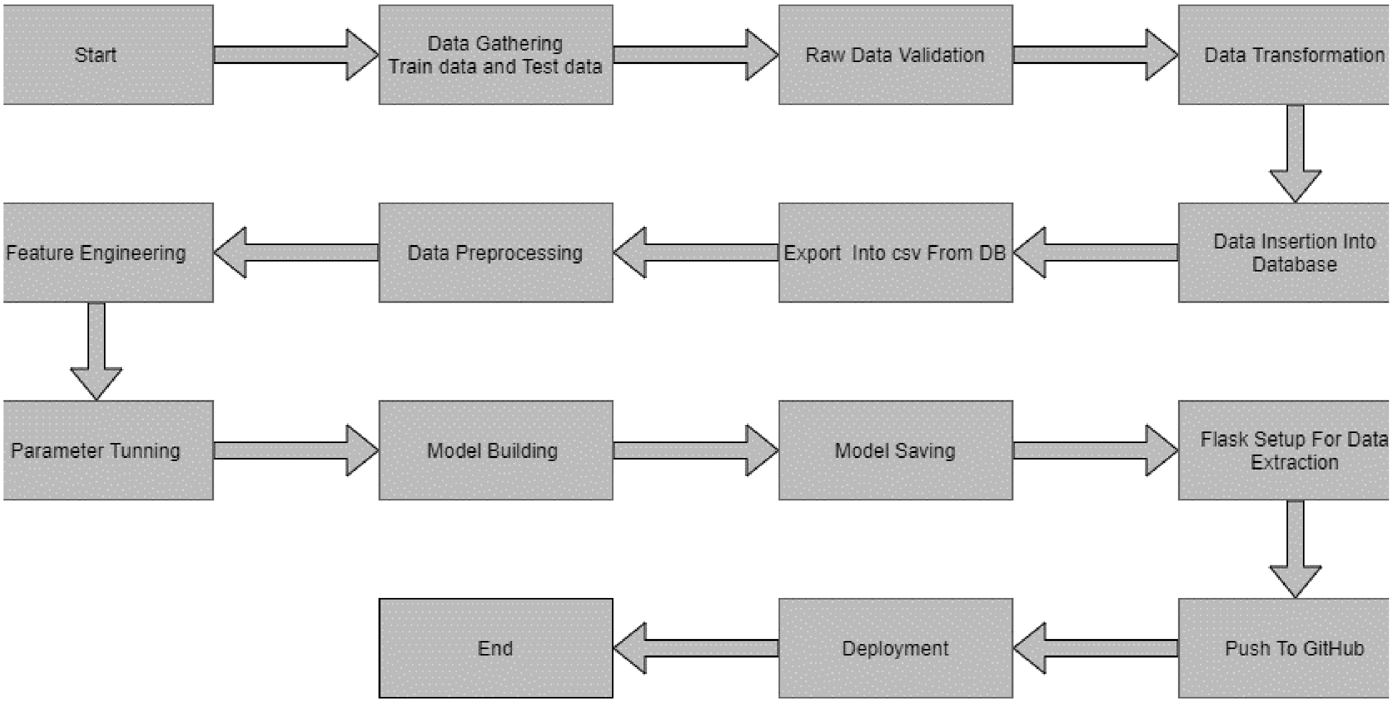
1.1 What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the House Price Prediction dashboard. LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level 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 data organization may be defined during requirement analysis and then refined during data design work.

2.Architecture



3.Architecture Description

3.1 Data description

Travel package purchase data is a publicly available recipe dataset. We have train(3421) and test(1467) datasets. It is a classification problem where we need to predict whether a customer purchases a travel package or not for the test data.

A brief description of columns is given below.

- **Customer ID - Unique customer ID Prod Taken: Whether the customer has purchased a package or not (0: No, 1: Yes)**
- **Age - Age of customer**
- **Type of Contact - How customer was contacted (Company Invited or Self Inquiry)**
- **City Tier - City tier depends on the development of a city, population, facilities, and living standards. The categories are ordered i.e. Tier 1 > Tier 2 > Tier 3**
- **Occupation - Occupation of customer**
- **Number of Persons Visiting - Total number of persons planning to take the trip with the customer**
- **Preferred Property Star - Preferred hotel property rating by customer**
- **Marital Status - Marital status of customer**
- **Number of Trips - Average number of trips in a year by customer**
- **Passport - The customer has a passport or not (0: No, 1: Yes)**
- **Own Car - Whether the customers own a car or not (0: No, 1: Yes)**
- **Number of Children Visiting - Total number of children with age less than 5 planning to take the trip with the customer**
- **Designation - Designation of the customer in the current organization**
- **Monthly Income - Gross monthly income of the customer**
- **Pitch Satisfaction Score - Sales pitch satisfaction score**
- **Product Pitched - Product pitched by the salesperson(type of package chosen)**
- **Number of Follow ups - Total number of follow-ups has been done by the salesperson after the sales pitch**
- **Duration of Pitch - Duration of the pitch by a salesperson to the customer(call duration)**

3.2 Data Gathering

Data source :
Train and Test data are stored in

3.3 Data Pre-Processing

In data pre-processing missing values have been replaced(with median in numeric columns and with mode in categorical columns). Outliers have been capped with nearest acceptable within the threshold. After pre-processing most of the data is normally distributed.

3.4 Feature Engineering

Features which are not important like customer Id has been removed. Most of the features which are categorical have been one hot encoded to convert them to numerical features. Best features have been selected based on K Best for all categorical features to enhance the accuracy of the model.

3.5 Parameter Tuning

Hyper-parameter tuning with cross validation has been performed with various models like decision trees, XG Boost classifier and stratified K fold with logistic regression, support vector classifier has been performed.

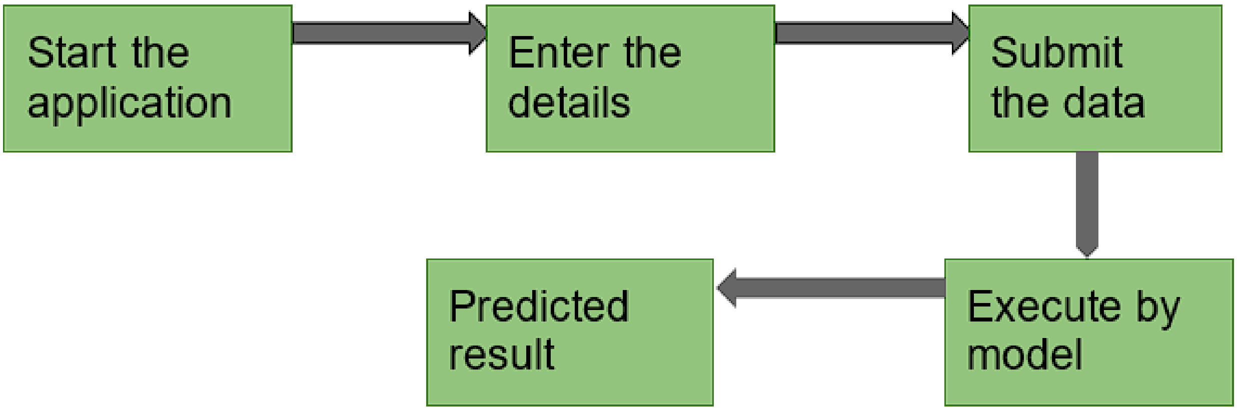
3.6 Model Building and Evaluation Metrics

After performing above mentioned operations, data has been passed into all the models and decision tree was the most accurate model(with 89% accuracy). Evaluation metrics like confusion matrix, AUC ROC

3.7 Deployment

The cloud environment was set up and the project was deployed from GitHub to Heroku Platform

App link - Will appear on Youtube
App demo link -



4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result	Achieved Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	The application URL should be accessible to the user	The application URL is accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed	The Application loads completely for the user when the URL is accessed
Verify whether the user is able to see input fields	1. Application is accessible	The user should be able to see the input fields	The user is able to see the input fields
Verify whether the user gets Submit button to submit the inputs	1. Application is accessible	The user should get Submit button to submit the inputs	The user is able to get Submit button to submit the inputs
Verify whether the predicted results are in accordance with the selections user made	1. Application is accessible	The predicted results should be in accordance with the selections user made	The predicted results are in accordance with the selections user made