Low Level Design

Restaurant rating prediction system

Written By	Tanuj shankarwar
Document Version	1
Last Revised Date	13 – Oct -2021

Document Control

Change Record:

Version	Date	Author	Comments
0.1	13 – Oct - 2021	Tanuj Shankarwar	Introduction & Architecture defined

Contents

1. Introduction

1.1. What is Low-Level design document?

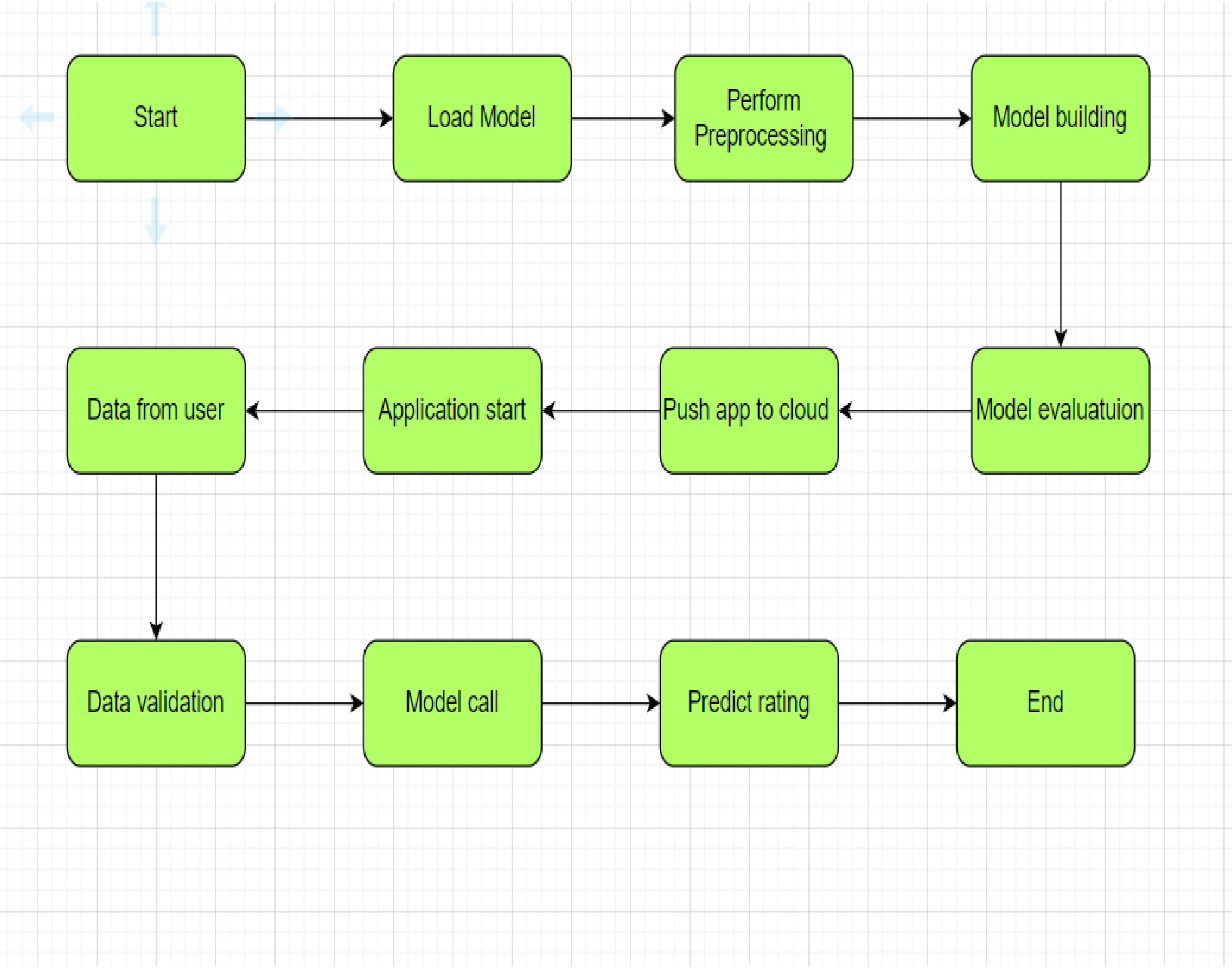
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can 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. 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 Description

3.1. Data Description

Zomato restaurants is publicly available dataset. It is a available in csv format on Kaggle. This dataset contains around 52,000 entries which is data of restaurants around Bengaluru.
Some of the columns in the dataset are address of the restaurant, location , contact no, restaurant type and the average cost for a meal.

3.2. Data Transformation

In the Transformation Process, we will convert our original dataset which is in CSV format to pandas dataframe Format. This makes it easy for us to manipulate and clean the data with the help of Pandas library.

3.4Data Pre-processing

Data Pre-processing is perhaps the most important step in the life cycle of a machine learning project. Considering the size of the dataset , it was no surprise that the data had a lot of missing values. In the preprocessing step , we imputed the columns with missing values . Mean was the strategy used in imputation . Columns like contact number , url are not really of any use to us so we dropped those columns.
Apart from this , data was also cleaned to get the row values in a suitable format before we could apply the algoritms and do exploratory data analysis .

3.10. Model Building

After data is preprocessed, we will use the train test split function from scikit learn library to separate the data into training and testing sets. We then apply one hot encoding to the respective columns and pass it into our pipeline. Now, we test various algorithms in order to predict the rating of a restaurant. According to our testing , Random Forest gives the best performance.

3.11. Data from User

Using streamlit , we have created the front end for the project Here we will collect basic data from user such as location of restaurant , online delivery, cost for two , type of restaurant etc. To the user , a set of input boxes will be available where he can select or type the data .

3.12. Data Validation

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

3.13. Model Call

Based on the data , the respective model will be loaded
and will be used to predict the restaurant rating.

3.14. Deployment

We will be deploying the model to Heroku. Heroku is a cloud platform as a service supporting several programming languages. With access to the link, any user predict the restaurant rating in real time .

Link-https://zomatoratepredictor.herokuapp.com/