

RESTAURANT RATING PREDICTION

LOW LEVEL DOCUMENT

BY:
HARSHITHA AV
SPOORTHY CHANDRA.P

ABSTRACT

The basic idea of analyzing the Zomato dataset is to get a fair idea about the factors affecting the establishment of different types of restaurants at different places in Bengaluru, aggregate rating of each restaurant.

Bengaluru being one such city has more than 50,000 restaurants with restaurants serving dishes from all over the world. With each day new restaurants opening the industry hasn't been saturated yet and the demand is increasing day by day. Bengaluru being an IT capital of India, most of the people here are dependent mainly on the restaurant food as they don't have time to cook for themselves. With such an overwhelming demand for new restaurants, it has become important to study the ratings of restaurants.

Introduction

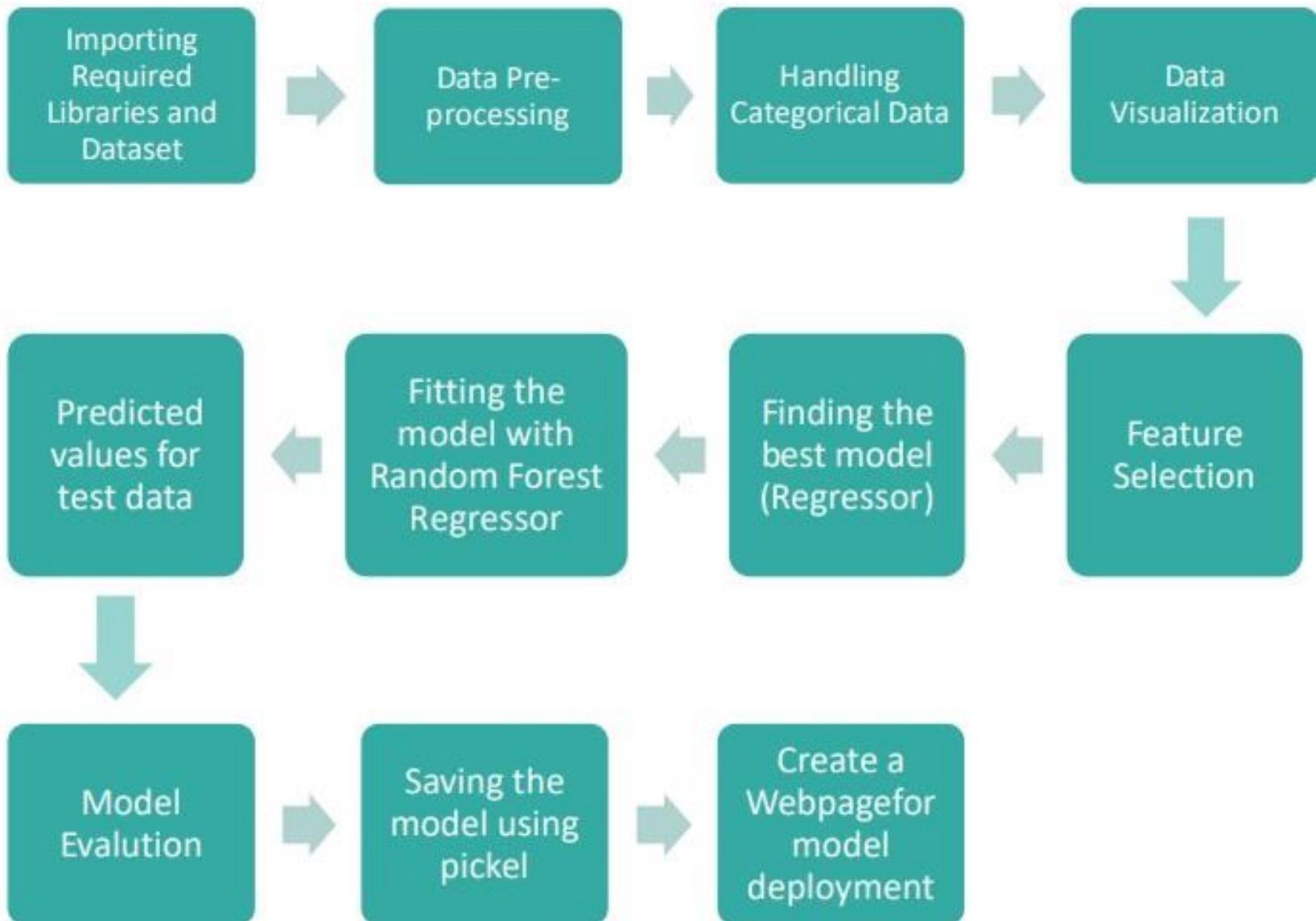
What is LLD Document ?

The main goal of the LLD document is to give the internal logic design of actual code implementation and supply the outline of the machine learning model and its implementation. Additionally, it provides the description how our project will designed end - to - end.

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.

ARCHITECTURE



Architecture Design

This project is designed to make an interface for the User to predict the rating of restaurant.

Data Collection

The data for this project is collected from the Kaggle Dataset, the URL for the dataset

<https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants>

Data Description

The dataset contains 17 variables all of which were scrapped from the Zomato website. The dataset contains details of more than 50,000 restaurants in Bengaluru in each of its neighborhood. The total size of dataset is approximately 547 MB.

url: contains the url of the restaurant in the zomato website

address: contains the address of the restaurant in Bengaluru

name: contains the name of the restaurant

online_order: whether online ordering is available in the restaurant or not

book_table: table book option available or not

rate: contains the overall rating of the restaurant out of 5

votes: contains total number of rating for the restaurant as of the above mentioned date.

phone: contains the phone number of the restaurant.

location: contains the neighborhood in which the restaurant is located.

rest_type: restaurant type.

dish_liked: dishes people liked in the restaurant.

cuisines: food styles, separated by comma.

approx_cost(for two people): contains the approximate cost for meal for two people.

reviews_list: list of tuples containing reviews for the restaurant, each tuple.

menu_item: contains list of menus available in the restaurant.

listed_in(type): type of meal.

listed_in(city): contains the neighborhood in which the restaurant is listed.

Importing dataset and libraries:

We have imported certain libraries such as numpy,panda,matplotlib and many more.

The ZOMATO dataset is imported in the form of csv.

Data Preprocessing

Checked for info of the Dataset, to verify the correct datatype of the Columns.

Checked for Null values, because the null values can affect the accuracy of the model.

Converted all the illegal values into legal values.

Checking the distribution of the columns to interpret its importance.

Prepared the relevant data from the dataset.

Now, the info is prepared to train a Machine Learning Model.

Modelling Creation

After preprocessing the data, we visualize our data to gain insights and then these insights are randomly spread and split into two parts, train and test data. After splitting the data, we use Random Forest Regressor to model our data to predict the Restaurant Rating.

UI Integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

Data from User

The data from the user is retrieved from the created HTML web page.

Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent to the prepared model for the prediction. Importing Required Libraries and Dataset Data Preprocessing Handling Categorical Data Data Visualization Predicted values for test data Fitting the model with Random Forest Regressor Finding the best model (Regressor) Feature Selection Model Evalution Saving the model using pickle Create a Webpagefor model deployment.

Rendering the Results

The data sent for the prediction is then rendered to the web page.

Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.

Test Case Description	Pre-Requisites	Expected Results
Verify whether the Webpage is accessible to the User or not.	Webpage URL should be defined.	Webpage should be accessible to the User.
Verify whether the Webpage is completely loads for the User or not	1. Webpage URL is accessible. 2. Webpage is deployed.	The Webpage should be completely loads for the User when it is accessed.
Verify whether the User is able to enter data in input fields or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to enter data in input fields.
Verify whether the User is able to submit details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to submit details to process.
Verify whether the User gets recommended results on submitting the details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable	The User gets recommended results on submitting the details.