Detailed Project Report

Restaurant Rating Prediction

Document Version Control

Date	Version	Description	Author
04/08/2022	1.0	Abstract Introduction General Description	Utkarsh Yeole
05/08/2022	1.1	Technical Requirements Data Requirements Data Preprocessing Design Flow	Utkarsh Yeole
06/08/2022	1.2	Data from User and its validation Rendering the Results Deployment Conclusion	Utkarsh Yeole

Contents

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 12,000 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.

1. Introduction

1.1 Why this DPR Document ?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on how the entire project has been designed end-to-end.

Key points :

- Describes the design flow
- Implementations
- Software requirements
- Architecture of the project
- Non-functional attributes like:
  - Reusability
  - Portability
  - Resource utilization

2. General Description

2.1 Problem Perspective

The Restaurant Rating Prediction may be a machine learning model that helps users to predict the rating of the restaurant and help them to understand about the price, quality, location etc. about the restaurant.

2.2 Problem Statement

The main goal of this project is to perform extensive Exploratory Data Analysis(EDA) on the Zomato Dataset and build an appropriate Machine Learning Model that will help various Zomato Restaurants to predict their respective Ratings based on certain features.

2.3 Proposed Solution

To solve the problem, we have created a User interface for taking the input from the user to predict the **Restaurant Rating** using our trained ML model after processing the input and at last the output (predicted value) from the model is communicated to the User. Detailed Project Report 6

2.4 Further Improvements

We also analysis the data used for training the ML model by considering different occasions such as Weekday, Season or any Social reasons, considering different angle of business. If we method such information and predict the **Restaurant Rating**, it will bring some loss to the restaurant but user can get benefit from that.

3. Technical Requirements

As technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have the device that has the access to the web and the fundamental understanding of providing the input.

3.1 Tools Used

• Python 3.9 is employed because the programming language and frameworks like NumPy, Pandas, Scikit - learn and alternative modules for building the model.

• Jupyter-Notebook is employed as IDE.

• For Data visualizations, seaborn and components of matplotlib are getting used.

• For information assortment prophetess info is getting used.

• Front end development is completed victimization HTML/CSS.

• Flask is employed for each information and backend readying.

• GitHub is employed for version management.

• Heroku is employed for deployment.

4. Data Requirements

The Data requirements is totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.xlsx).

4.1 Data Collection

The data for this project is collected from the Kaggle Dataset, the URL for the dataset is https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants?resource=download

4.2 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.

Variable	Type	#Unique Values	Description
url	object	51,717	contains the url of the restaurant in the zomato website
address	object	11,495	contains the address of the restaurant in Bengaluru
name	object	8,792	contains the name of the restaurant
online_order	category	2	whether online ordering is available in the restaurant or not
book_table	category	2	table book option available or not
rate	object	64	contains the overall rating of the restaurant out of 5
votes	int	2328	contains total number of rating for the restaurant as of the above mentioned date
phone	object	64	contains the phone number of the restaurant
location	category	93	contains the neighborhood in which the restaurant is located
rest_type	category	93	restaurant type
dish_liked	object	5271	dishes people liked in the restaurant
cuisines	object	2723	food styles, separated by comma
approx_cost(for two people)	float	70	contains the approximate cost for meal for two people
reviews_list	object	22513	list of tuples containing reviews for the restaurant, each tuple consists of two values, rating and review by the customer
menu_item	object	9098	contains list of menus available in the restaurant
listed_in(type)	category	7	type of meal
listed_in(city)	category	30	contains the neighborhood in which the restaurant is listed

4.3 Importing data into Database

Created associate API for the transfer of the info into the Cassandra info, steps performed are:

- Connection is created with the info.
- Created a info with name ZomatoInfo.
- cqlsh command is written for making the info table with needed parameters.
- And finally, a cqlsh command is written for uploading the Knowledge Set into data table by bulk insertion.

4.4 Exporting Data from Database

In the above created API, the download URL is also being created, which downloads the data into a csv file format.

5. Data Preprocessing

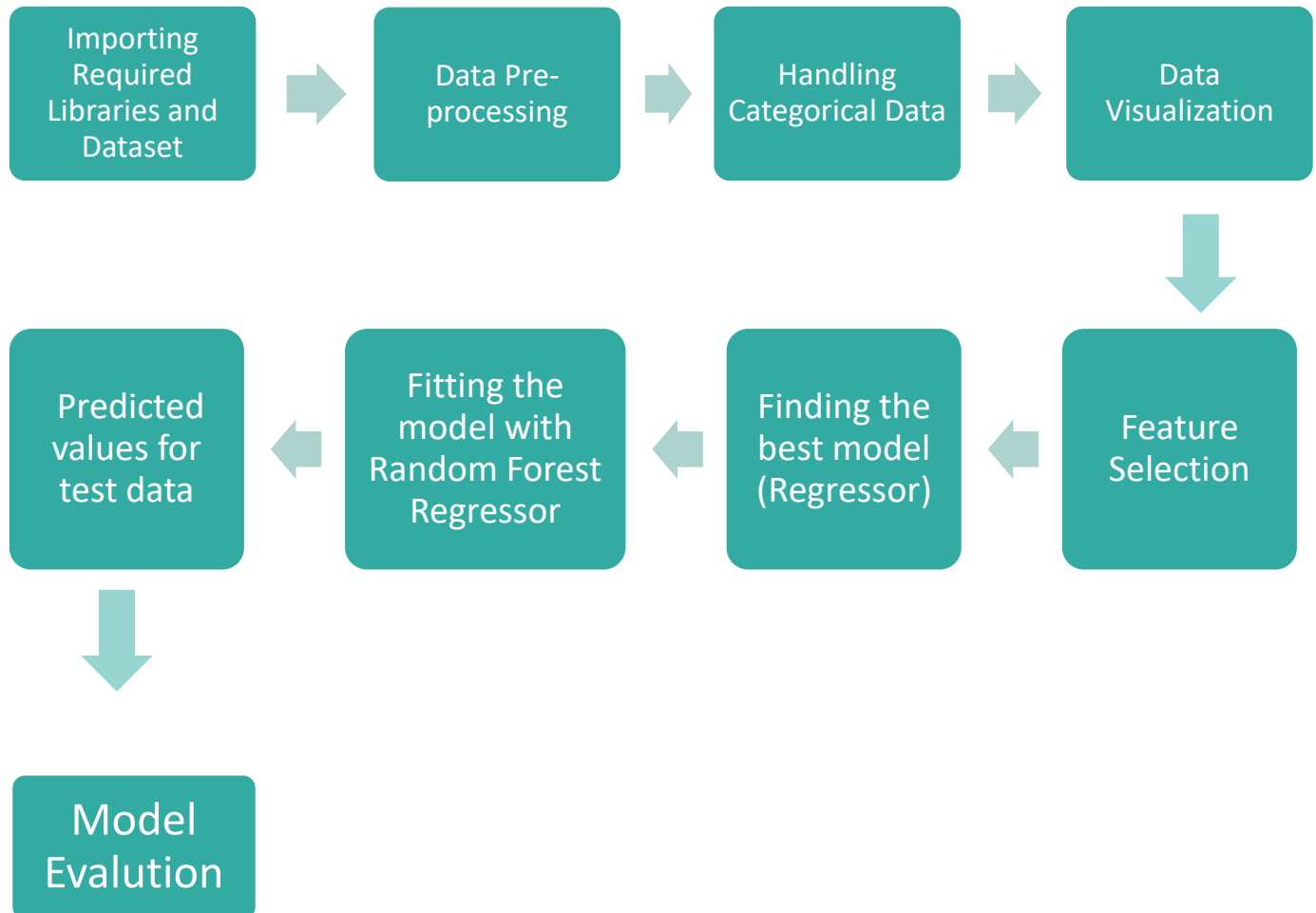- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Converted all the desired columns into Datetime format.
- Performed One – Hot encoding on the desired columns.
- Checking the distribution of the columns to interpret its importance.
- Now, the info is prepared to train a Machine Learning Model.

6. Design Flow

6.1 Modelling Creation and Evaluation

```
┌──────────────┐      ┌──────────────┐      ┌──────────────┐      ┌──────────────┐
│  Importing    │      │              │      │              │      │              │
│  Required     │  →   │  Data Pre-   │  →   │  Handling    │  →   │     Data     │
│  Libraries and│      │  processing  │      │ Categorical  │      │ Visualization│
│  Dataset      │      │              │      │    Data      │      │              │
└──────────────┘      └──────────────┘      └──────────────┘      └──────────────┘
                                                                          ↓
┌──────────────┐      ┌──────────────┐      ┌──────────────┐      ┌──────────────┐
│              │      │ Fitting the  │      │              │      │              │
│  Predicted   │  ←   │ model with   │  ←   │ Finding the  │  ←   │   Feature    │
│ values for   │      │Random Forest │      │ best model   │      │  Selection   │
│  test data   │      │  Regressor   │      │ (Regressor)  │      │              │
└──────────────┘      └──────────────┘      └──────────────┘      └──────────────┘
       ↓
┌──────────────┐
│    Model     │
│  Evalution   │
└──────────────┘
```

6.2 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.

6.3 Deployment Process

Saving the model using pickel	→	Create a Webpage for model deployment	→	Deployment on Local Machine using flask	→	Deployment on Heroku Cloud	→	Input value and Predict the final result

6.4 Logging

In logging, at each if an error or an exception is occurred, the event is logged into the system log file with reason and timestamp. These helps the developer to debug the system bugs and rectifying the error.

7. Data from User

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

8. 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.

9. Rendering the Results

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

10. Deployment

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

11. Conclusion

The Restaurant Rating Prediction system will predict the rating for helping the customers with the trained knowledge with set of rules. The user can use this system to recognize the approximate rating of the restaurant.

12. Frequently Asked Questions (FAQs)

Q1) What's the source of data?
The data for training is provided by the client in multiple batches and each batch contain multiple files.

Q2) What was the type of data?
The data was the combination of numerical and Categorical values.

Q3) What's the complete flow you followed in this Project?
Refer Page no 9 for better Understanding.

Q4) After the File validation what you do with incompatible file or files which didn't pass the validation?
Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed?
We are using different logs as per the steps that we follow in validation and modeling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q6) What techniques were you using for data pre-processing ?

• Removing unwanted attributes.

• Visualizing relation of independent variables with each other and output variables. • Checking and changing Distribution of continuous values.

• Removing outliers

• Cleaning data and imputing if null values are present.

• Converting categorical data into numeric values.

Q7) How training was done or what models were used?

• Before dividing the data in training and validation set, we performed pre-processing over the data set and made the final dataset.

• As per the dataset training and validation data were divided.

• Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used based on the recall, final model was used on the dataset and we saved that model**.**

Q8) How Prediction was done?

The testing files are shared by the client. We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q9) What are the different stages of deployment?

• First, the scripts are stored on GitHub as a storage interface.

• The model is first tested in the local environment.

• After successful testing, it is deployed on Heroku.