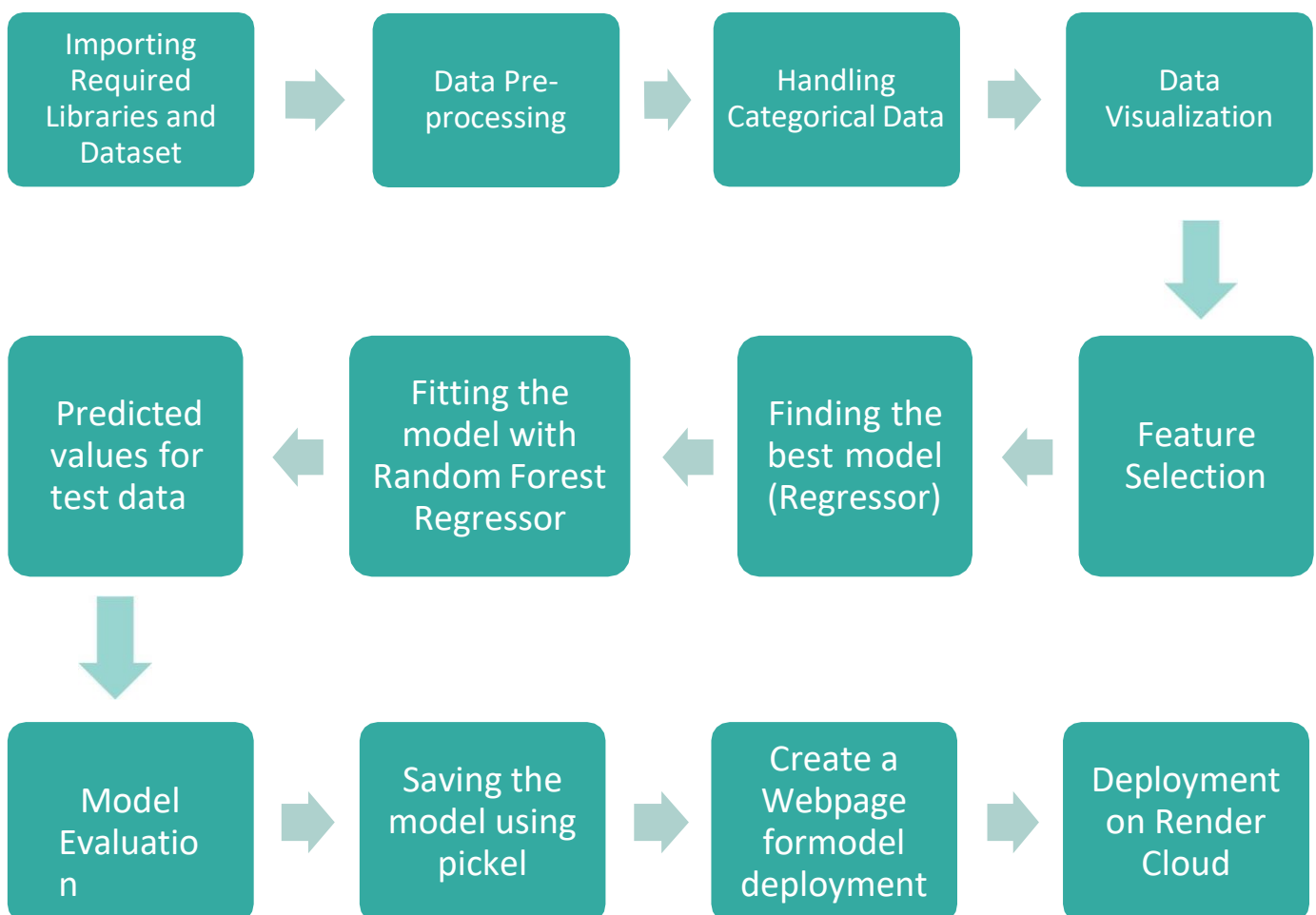# 1. Introduction

1.1 What this Architecture Design Document ?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the flight fare prediction code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

2. Architecture

Importing Required Libraries and Dataset	→	Data Pre-processing	→	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 Evaluation	→	Saving the model using pickel	→	Create a Webpage formodel deployment	→	Deployment on Render Cloud

3. Architecture Design

3.1 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

3.2 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.

url	Urls of the restaurants on Zomato website
address	Address of Restaurants
name	Name of the Restaurant
online_order	Whether online facility is provided or not
book_table	If advanced booking of table is allowed or not
rate	Average rate of the restaurant given by the customers
votes	Total number of votes
phone	Phone number of restaurant
location	Location of restaurant
rest_type	Type of restaurant
dished_liked	Most dished liked in that restaurant
cuisines	Types of cuisines served in that restaurant
approx_cost(for two people	What is the approximate cost of the 2 people's meal
reviews_list	Rate and reviews given by the customers

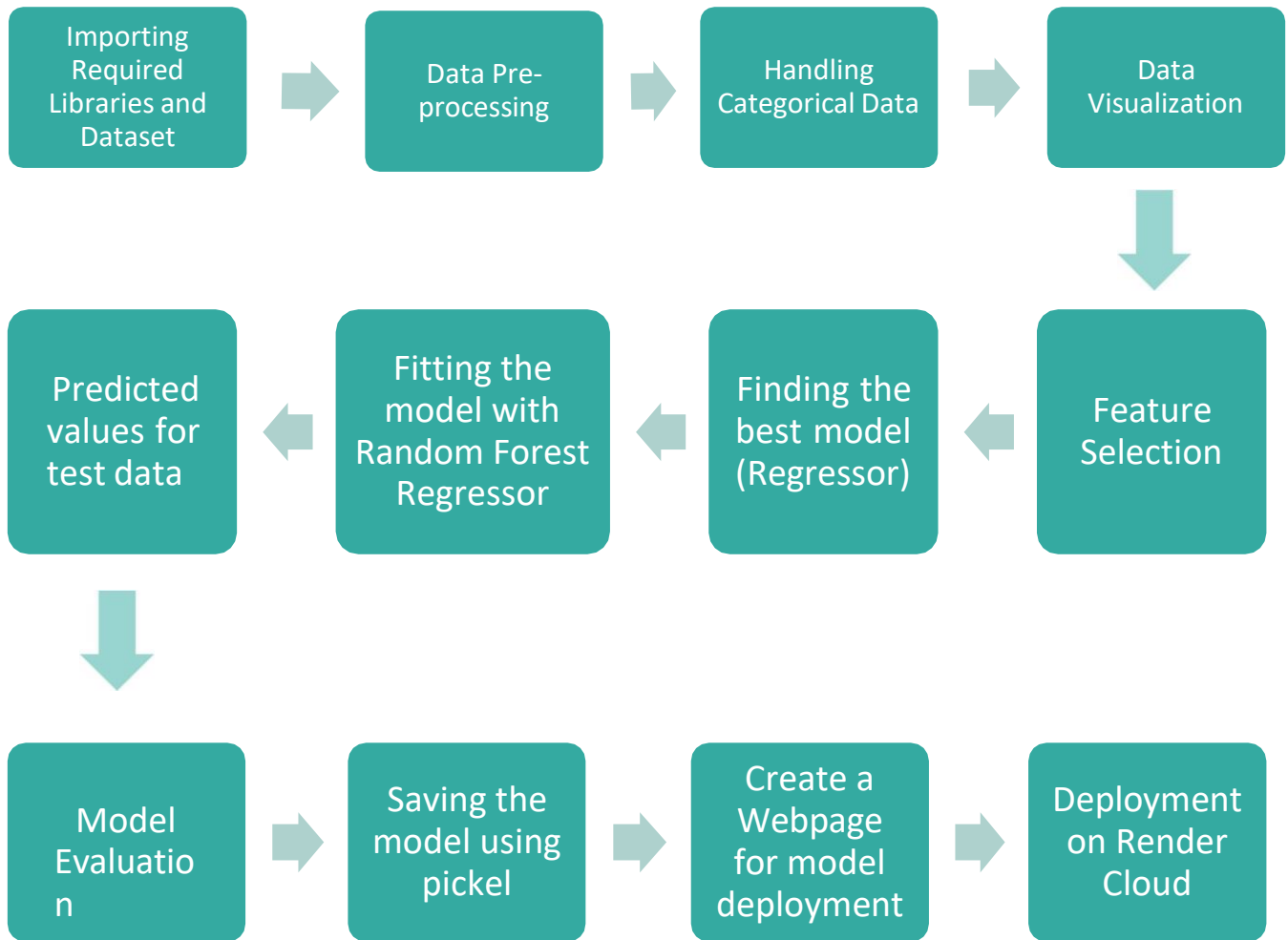|	
menu_item	List of Menu Items
listed_in(type)	Type of meal
listed_in(city)	City where the restaurant is located

3.3 Data Preprocessing

This included importing important libraries such as Seaborn, Matplotlib, pandas etc. We imported the same dataset mentioned above from Kaggle.

- Checked for info of the Dataset, to verify the correct datatype of theColumns.
- Checked for Null values, because the null values can affect the accuracy ofthe model.
- Converted all the illegal values into legal values.
- Performed Labeled encoding and 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.

3.4 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

```
┌──────────────┐     ┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│  Importing   │     │              │     │              │     │              │
│  Required    │ ──> │  Data Pre-   │ ──> │  Handling    │ ──> │    Data      │
│ Libraries and│     │  processing  │     │Categorical Data│    │Visualization │
│   Dataset    │     │              │     │              │     │              │
└──────────────┘     └──────────────┘     └──────────────┘     └──────────────┘
                                                                        │
                                                                        v
┌──────────────┐     ┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│  Predicted   │     │ Fitting the  │     │ Finding the  │     │              │
│  values for  │ <── │ model with   │ <── │  best model  │ <── │   Feature    │
│  test data   │     │Random Forest │     │  (Regressor) │     │  Selection   │
│              │     │  Regressor   │     │              │     │              │
└──────────────┘     └──────────────┘     └──────────────┘     └──────────────┘
        │
        v
┌──────────────┐     ┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│              │     │              │     │  Create a    │     │              │
│    Model     │     │ Saving the   │     │  Webpage     │     │ Deployment   │
│  Evaluatio   │ ──> │ model using  │ ──> │ for model    │ ──> │ on Render    │
│    n         │     │   pickel     │     │ deployment   │     │   Cloud      │
└──────────────┘     └──────────────┘     └──────────────┘     └──────────────┘
```

3.5 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.

3.6 Data from User

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

3.7 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.

3.8 Rendering the Results

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

3.9 Deployment

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