Detailed Project Report for Restaurant Rating Prediction

Prepared by

Vikram Nayyar

vikramnayyar@live.com

November 30, 2021

Abstract

In the recent past, a number of restaurants have emerged. Subsequently, the restaurant business is witnessing a surge in the customer count. This has resulted in various restaurant categories at several locations with numerous cuisines, prices, services and offers. Therefore, predicting the customer's liking is remarkably challenging. To overcome the limitation, various data science models were trained on the dataset. The best validated model was selected. Using this model an application was developed for the user. This application accepts six user inputs, and accordingly; provides the accurate rating prediction.

1. Introduction

1.1 Problem Statement

In the past six years, over 1400 new eateries have been opened. Evidently, the population's willingness to visit restaurants; has phenomenally increased. Subsequently, numerous restaurant categories have emerged at several locations. Furthermore, they offer several cuisines, price ranges and services and offers.

Therefore, predicting the customer's liking is remarkably challenging. As; a large amount of finance and time is invested, evaluation of customer tendency is critical. To make the business successful the restaurant must attract customers.

1.2 Goal

The purpose of the product is to predict restaurant ratings. For a restaurant, maintaining a good rating is of chief importance. A highly rated restaurant consistently attracts large customers. This is essential to popularize the restaurant in town.

A good customer evaluation; certainly lays a lasting platform for the restaurant’s profit.

1.3 Product Merits

- Prior awareness of restaurant’s rating
- Accurately rates over 10,000 types of restaurants
- Awareness of market trend
- Multiple strategies for increased profits

2. General Description

2.1 Proposed Solution

A general product design is represented in figure 1. Based on the particular restaurant, a user enters the suitable restaurant parameters. GUI accepts the inputs; and assigns them values according to the model dictionary. Succeeding inputs are fed to the model.

The model processes the inputs and sends the predictions to the GUI. GUI displays the output to the user.

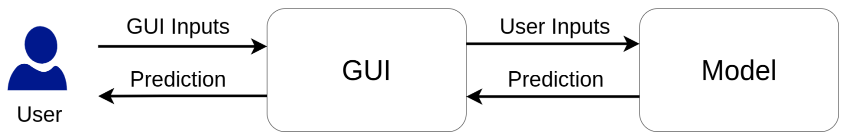
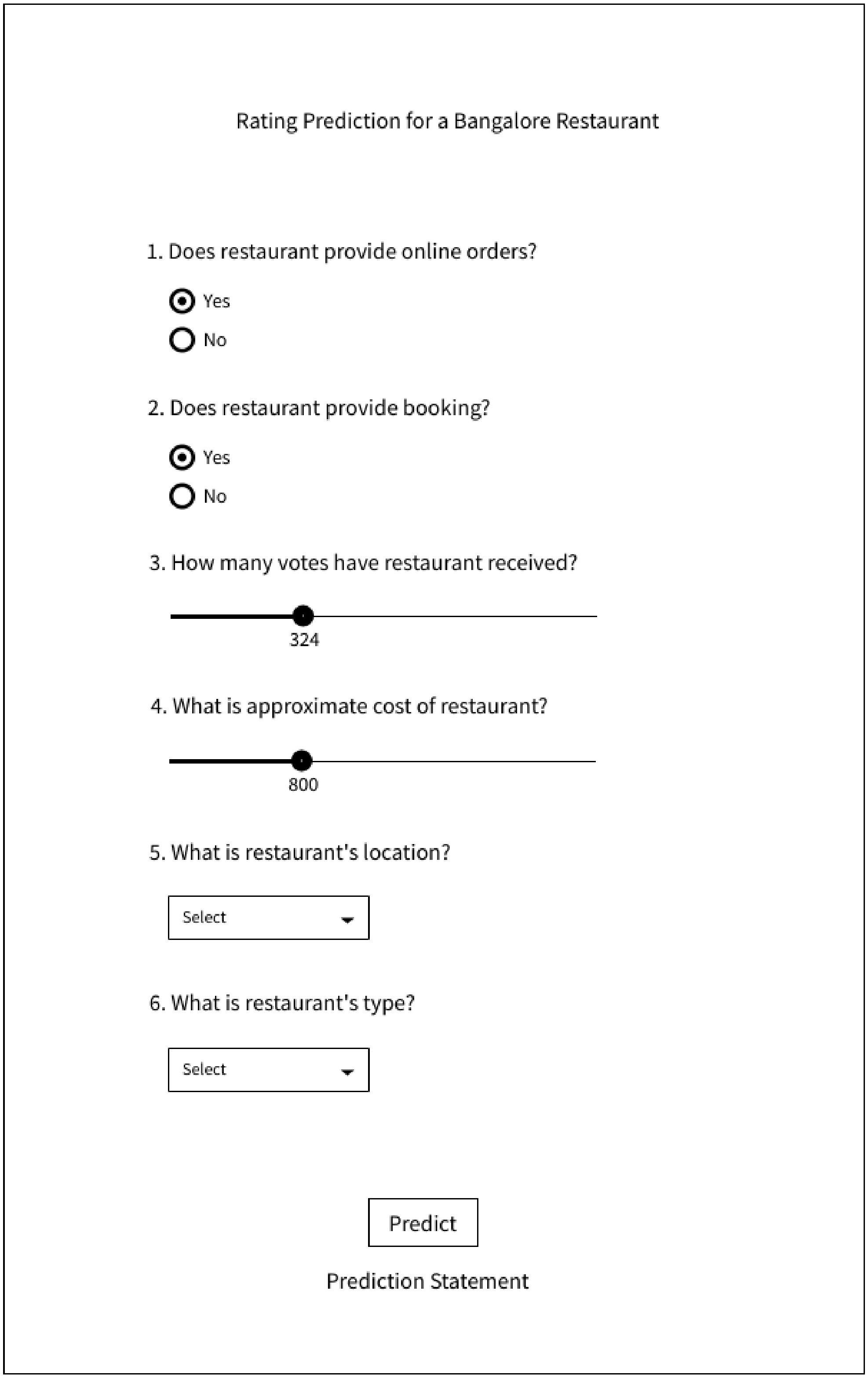


Fig. 1. General design of the data science application.

2.2 User Interface

Wireframe of the product’s UI is depicted in figure 2. In order to predict rating, the application expects 6 user inputs. Each input is properly labeled and is clearly annotated with the user selection. This prevents any confusion in selection. Please note; usual values are pre-selected for each user field. This is for the user's convenience.

The user selections are followed by the predict button. Clicking this button; inputs the user selections to the model, and the predicted rating is displayed below the predict button.



Rating Prediction for a Bangalore Restaurant

1. Does restaurant provide online orders?

● Yes
○ No

2. Does restaurant provide booking?

● Yes
○ No

3. How many votes have restaurant received?

324

4. What is approximate cost of restaurant?

800

5. What is restaurant's location?

Select

6. What is restaurant's type?

Select

Prediction Statement

Fig. 2. Wireframe of the data science application.

2.3 Future Scope

As the product is based on Bangalore restaurants, this can be extended to a number of cities. For deeper insights, the dataset from other platforms; like Swiggy or Dineout can be utilized.

2.4 System Environment

- **Programming Language:** Python 3.8.8
- **Numerical Calculations:** Numpy, Scipy
- **Plots:** Matplotlib, Seaborn
- **Model Training:** Scikit Learn, XGBoost, LightGBM, CatBoost
- **Application Interface:** Streamlit
- **Deployment:** Heroku

2.5 Constraints

Previously discussed wireframe; depicts a suitable layout and describes necessary inputs. Therefore, referring to the wireframe will prevent any oversight. Given a number of user inputs, the application should be prefilled with usual values. This will be convenient for the user.

New market trends emerge constantly. Therefore; after a suitable period, the model is required to be updated. This will maintain the model accuracy.

3. Design Architecture

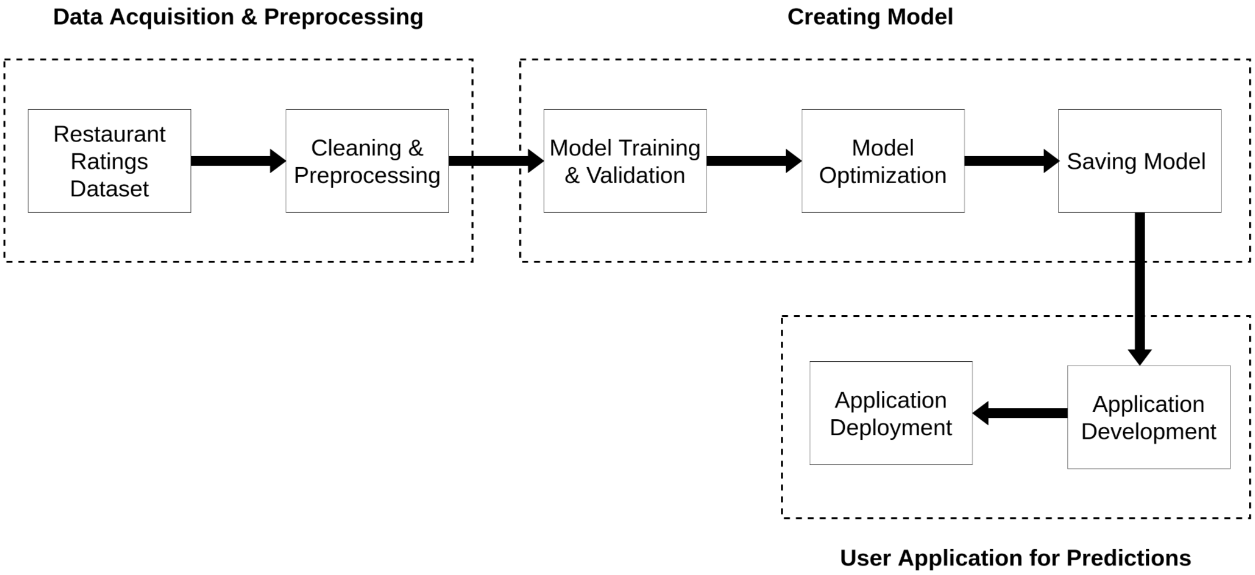

Data Acquisition & Preprocessing
Creating Model
Restaurant Ratings Dataset
Cleaning & Preprocessing
Model Training & Validation
Model Optimization
Saving Model
Application Deployment
Application Development
User Application for Predictions

Fig. 3. Process flow description of application.

The project scheme is shown in figure 3. The primary design constituents are described as follows

3.1 Data Acquisition & Preprocessing

Bangalore restaurants dataset was acquired from Zomato. This was uploaded by Himanshu Poddar in Kaggle. The dataset features are cleaned and unnecessary features are dropped. The features are converted to categorical variables and consequent outliers are removed. Subsequently, the data is split to training and test sets.

3.2 Modeling

A number of regression models are trained and validated. The model with highest accuracy is selected. The weak features corresponding to the selected model are removed. This further improves the accuracy. Finally, the resulting model is saved.

3.3 User Application

The model requires 6 different inputs to make rating prediction. These inputs are specific to the user and restaurant. Therefore, to accept different inputs from user; an application was developed. The application wireframe was described in the previous section. This application imports the saved model to make predictions.

4. Design Details

This section provides a detailed description of the project. The following subsections discuss different parts of the project.

4.1 Data Acquisition

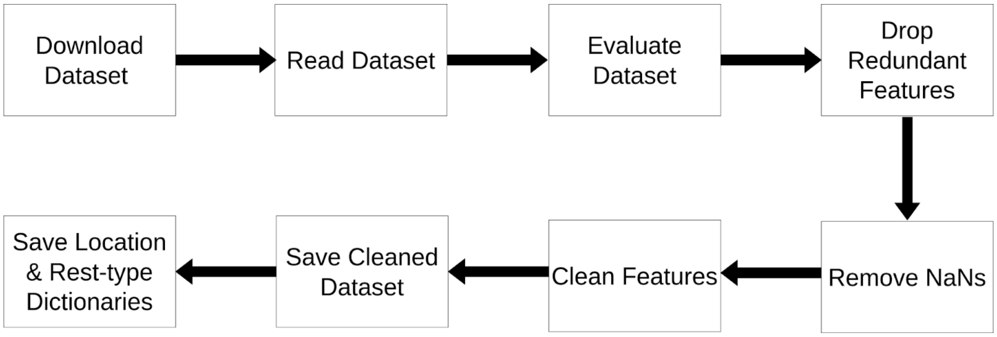


Fig. 4. The scheme of data acquisition.

Data acquisition process is described in figure 4. **"get_data.py"** script acquires the data. The script downloads the dataset using google drive link. The dataset features are evaluated. Redundant features and NaNs are removed. The features requiring cleaning are identified and subsequently cleaned. Besides, locations and restaurant type dictionaries are also saved. These dictionaries are later used by the Streamlit app.

4.2 Data Visualization

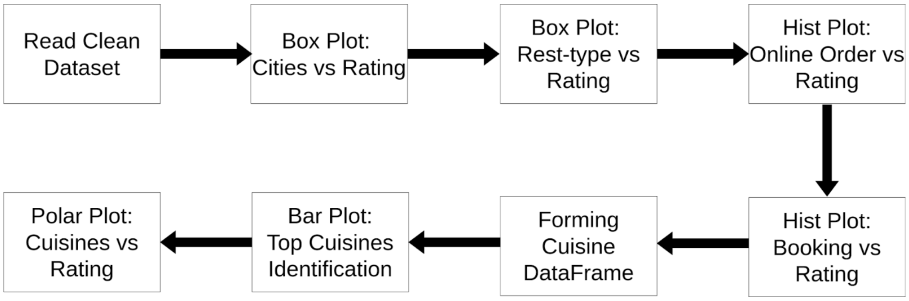

Read Clean Dataset
Box Plot: Cities vs Rating
Box Plot: Rest-type vs Rating
Hist Plot: Online Order vs Rating
Hist Plot: Booking vs Rating
Forming Cuisine DataFrame
Bar Plot: Top Cuisines Identification
Polar Plot: Cuisines vs Rating

Fig. 5. The scheme of data visualization.

The method of data visualization is described in figure 5. Data visualization is executed using **"data_analysis.py"** script. Functions to form box plots, histograms, bar plots and polar plots are declared in "data_analysis_util.py". Using these functions different respective visualizations are obtained. Also, these visualizations are saved in "visualizations" directory.

4.3 Data Preparation

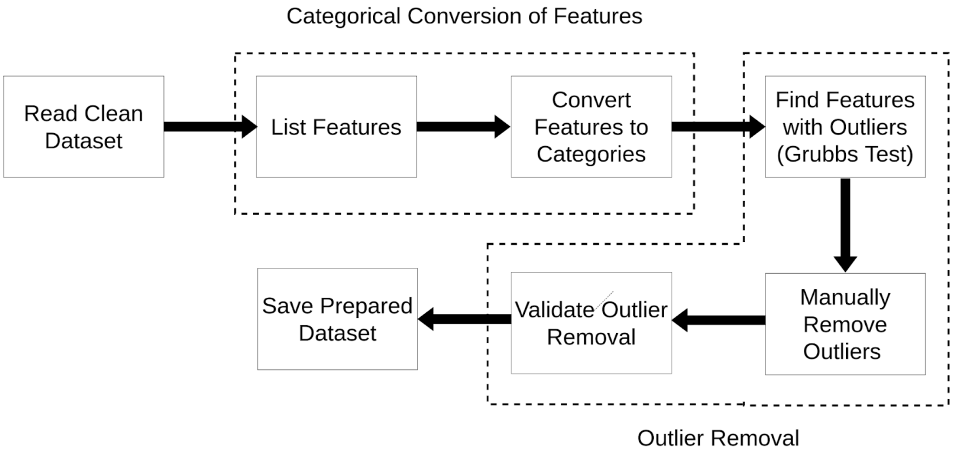


Fig. 6. The scheme of data preparation.

The data preparation process is depicted in figure 6. Data is prepared using the **“prepare_data.py”** script. Features requiring categorical conversion are listed and respectively converted. Subsequent outliers in features are identified using Grubbs Test. Manually, these outliers are removed and validated. Prepared dataset is saved.

4.4 Data Splitting

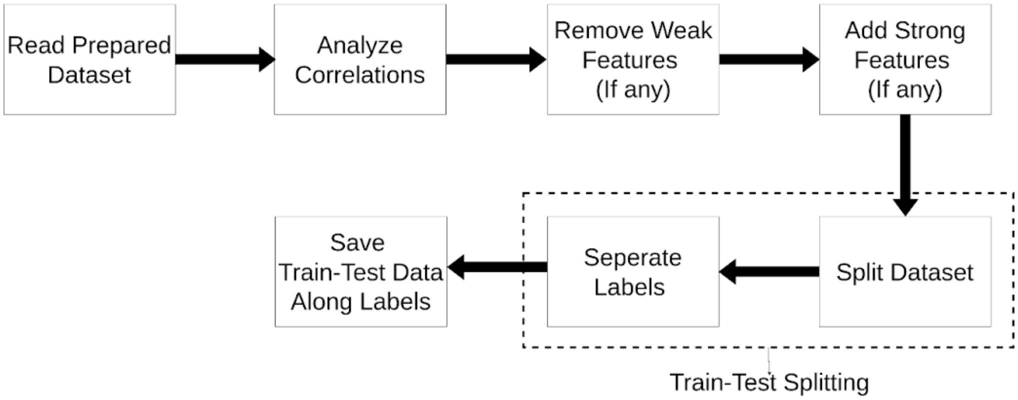

Read Prepared Dataset
Analyze Correlations
Remove Weak Features (If any)
Add Strong Features (If any)
Split Dataset
Seperate Labels
Save Train-Test Data Along Labels
Train-Test Splitting

Fig. 7. The scheme of data splitting.

Data splitting is accomplished using the “split_data.py” script. Figure 7 depicts the script processing. The script reads the clean dataset. Feature correlations are analyzed to determine weak and strong features. Accordingly, features can be dropped or new features can be added. Further, the dataset is split using stratified sampling. This ensures the fair splitting. Labels are separated from train and test sets. Train-test data along features is saved.

4.5 Modelling

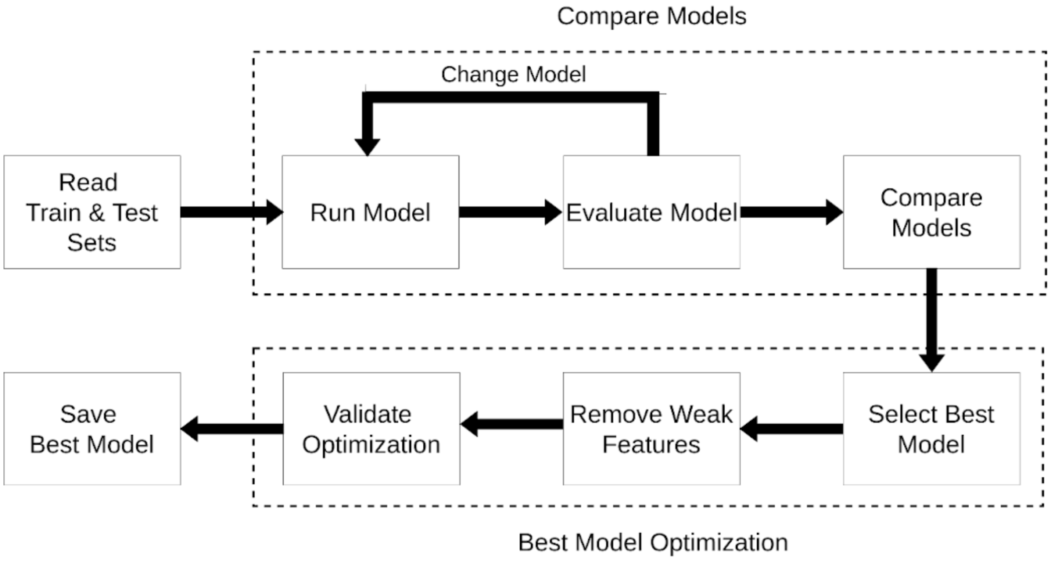


Fig. 8. The process of modelling.

The process of modelling is described in figure 8. The script “model_data.py” models the dataframe. This script reads train and test sets and runs a number of models. Each model is evaluated to compare different models. Based on accuracy, the best model is selected. Weak features in the model are identified and eliminated. Improved model performance is validated to finally save the best model.

4.6 Application Development

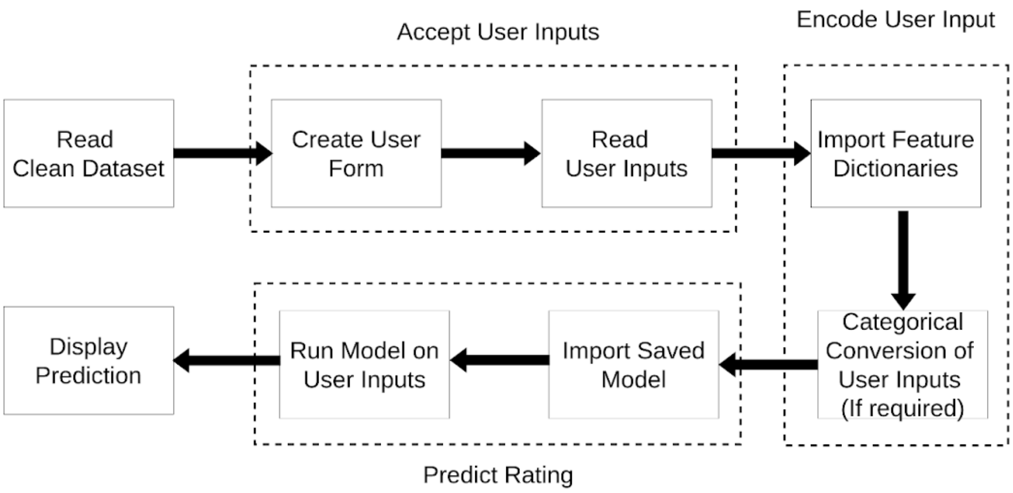


Fig. 9. The process of application development.

To provide user interaction an application is developed. This app accepts six user inputs; to predict the restaurant rating. The app is developed using **"app.py"** script. Figure 9 describes the operation of the script. It reads the clean dataset. To accept the user inputs; a user form is created. The dataset is used to extract user input options like locations, restaurant type etc. The received user inputs are converted to categorical variables. Features with many categories use dictionaries for categorical conversion. The model obtained from the previous step is imported. User inputs are fed to the model to obtain restaurant rating prediction. This prediction is displayed in the application.

4.7 Utility Scripts

"get_data_util.py", **"data_analysis_util.py"**, **"prepare_data_util.py"**, **"split_data_util.py"**, **"model_data_util.py"** and **"utility.py"** declare vital functions that are required by respective scripts.

Besides, **“run_project.py”** file runs all the project scripts sequentially (including application). Therefore, the entire project is executed with this script. This is shown in figure 10.

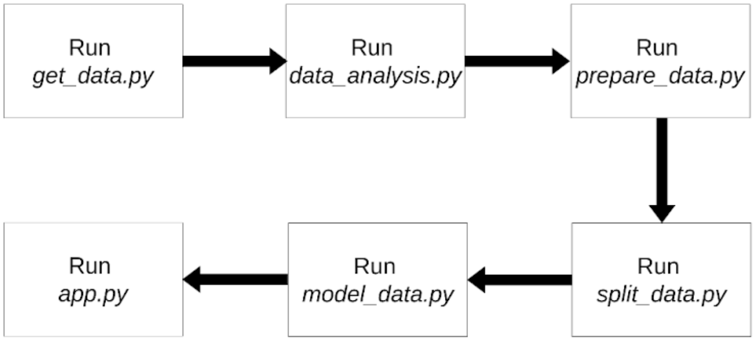


Fig. 10. The operation of “run_project.py”.

5. Results

The project analyzes the dataset to obtain various insights of Bangalore restaurants. Furthermore, to accurately predict the rating; a number of regression models are trained on the dataset. The results of these operations are described in the following subsections.

5.1 Data Analysis

5.1.1 Book Table vs Rating

Using histogram and box plot; the table bookings are compared with the rating. The plots are shown in figure 11. The figure clearly depicts that restaurants offering table bookings tend to possess higher ratings.

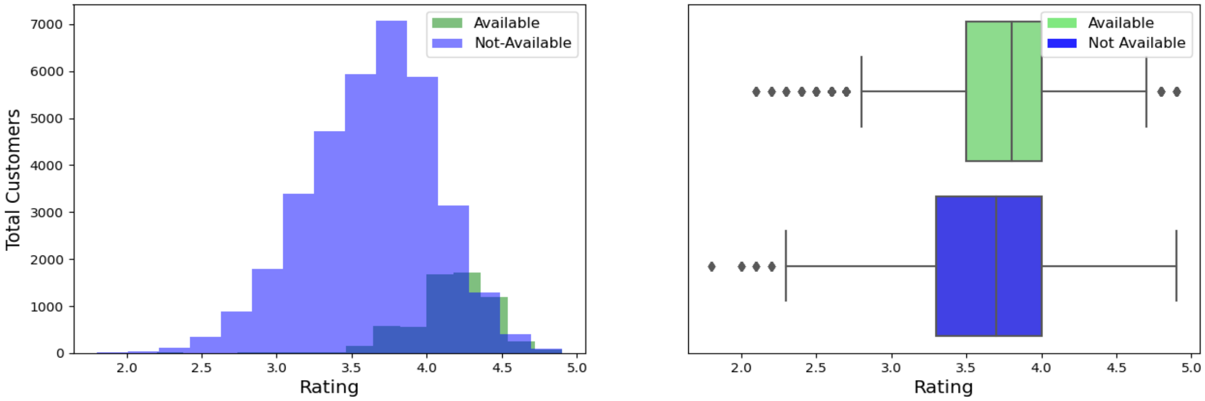


Fig. 11. Analysis of table booking with restaurant rating.

5.1.2 Online Order vs Rating

Using histogram and box plot; the online orders are compared with the rating. The plots are shown in figure 12. The figure suggests that the restaurants offering online bookings tend to possess higher ratings.

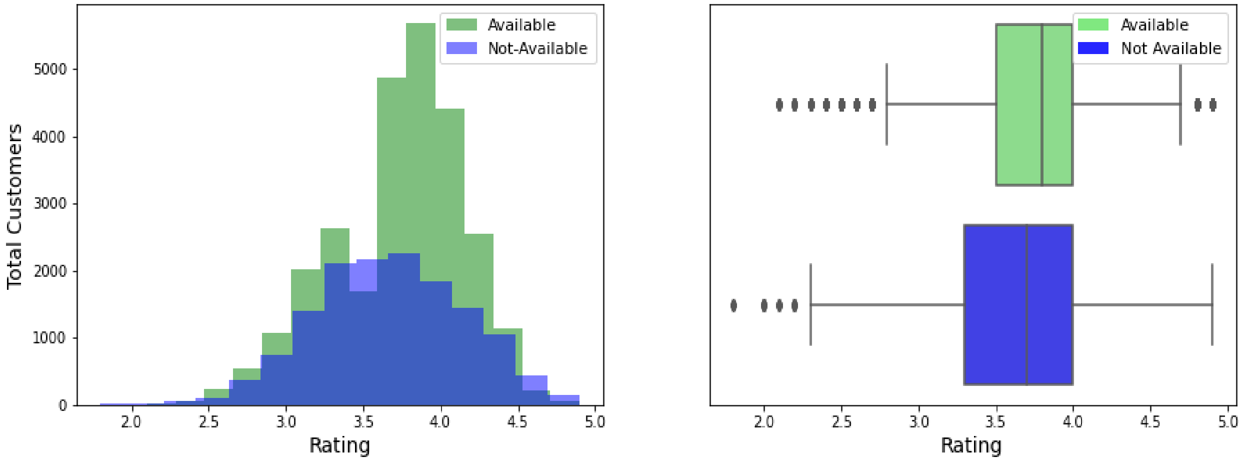


Fig. 12. Analysis of online orders with restaurant rating.

5.1.3 Restaurant Type vs Rating

Box plot compares the rating of different restaurant types. The restaurant types are arranged in ascending order of rating. “Buffet”, “drinks & nightlife” and “pubs & bars” restaurant types are rated highest.

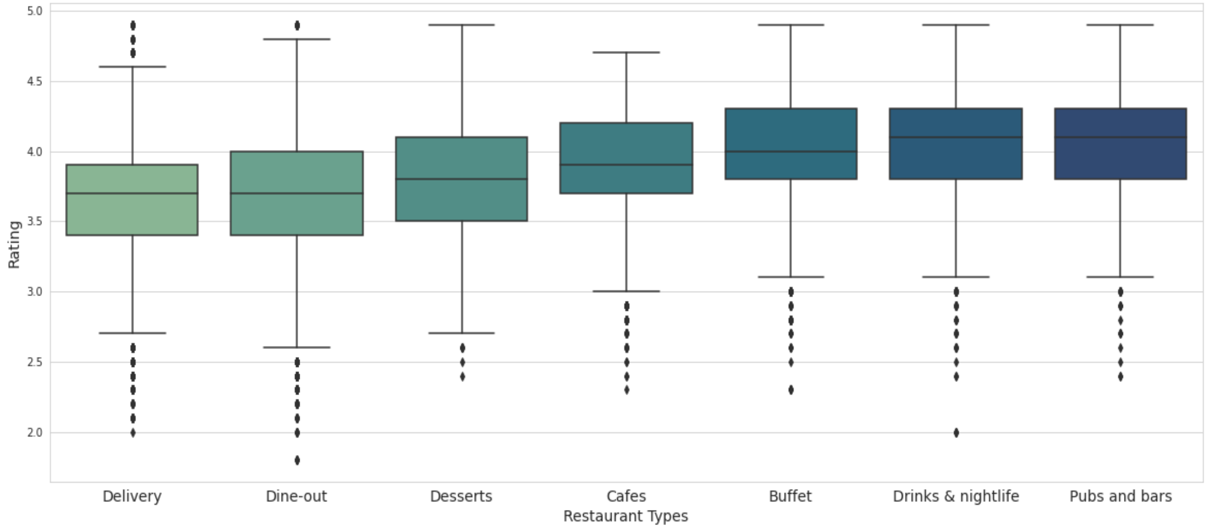


Fig. 13. Analysis of online orders with restaurant rating.

5.1.4 Top Cuisines

Figure 14 identifies top cuisines in Bangalore. As there are a very large number of cuisines, restaurant count is taken as the basis of identification. This eliminates disfavored cuisines.

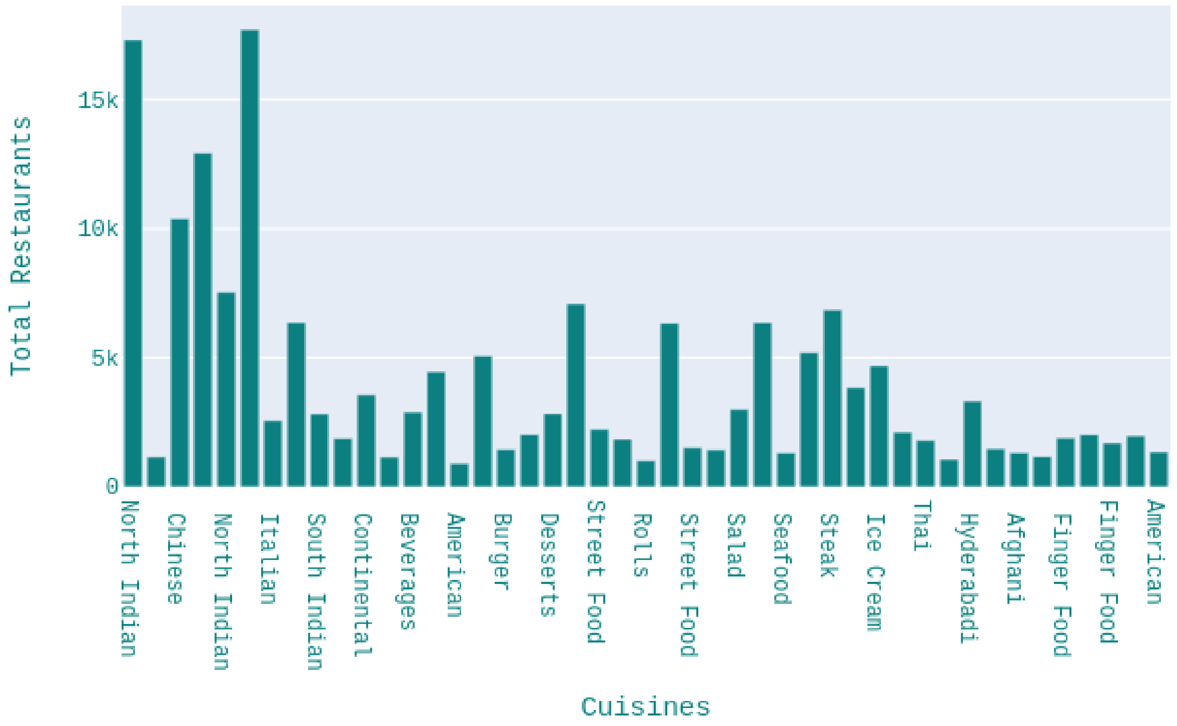


Fig. 14. Distribution of top cuisines across restaurants.

5.1.3 Cuisines vs Rating

The rating of top cuisines are compared using a polar plot. This is shown in figure 15. “Italian” and “American” are rated highest.

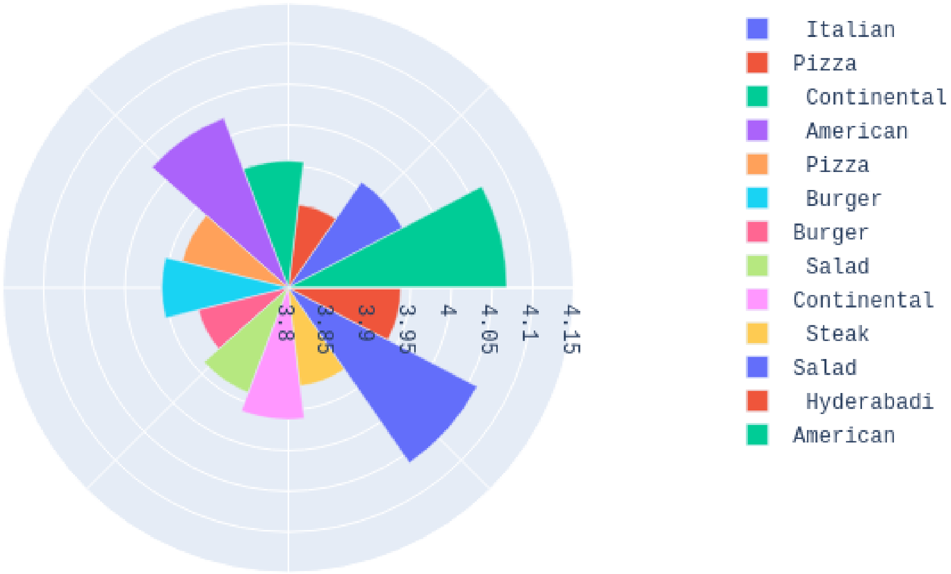


Fig. 15. Analysis of cuisines with restaurant rating.

5.2 Model Evaluation

The dataset is modelled using a number of models. The performance of these models is provided in table 1. Extra-Trees Regressor has the highest accuracy.

Table 1: Performance comparison of different data science models.

SNo	Model	R2-Score
1	Extra-Trees Regressor	0.849330
2	Random Forest Regressor	0.839837
3	Bagging Regressor	0.839510
4	XGB Regressor	0.520387
5	Catboost Regressor	0.297755
6	LGBM Regressor	0.262219
7	Gradient Boosting Regressor	-0.020233
8	Adaboost Regressor	-0.851571

Extra-Trees Regressor is selected. To optimize the performance of Extra-Trees Regressor, feature importance is determined. This is shown in figure 16.

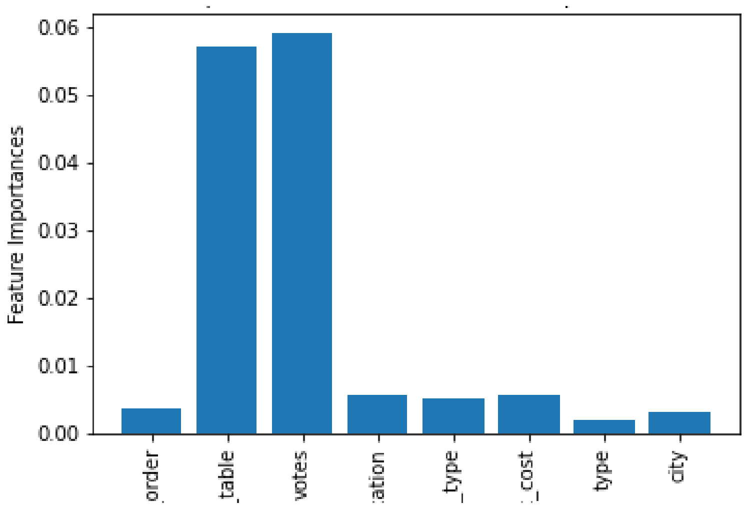


Fig. 16. Comparison of different feature importance.

Weak features are removed to obtain an approximate R2 Score of 0.9174 and mean square error (MSE) of 0.0149.