A Predictive Study on Zomato Restaurants Ratings

Project Report submitted to the

SDM Post Graduate Centre, Ujire

in partial fulfilment of the degree of

MASTER OF SCIENCE

In

STATISTICS

by

Shrajan Kumar

Under the supervision of

Ms. Shwetha Kumari

Assistant Professor

Department of Post Graduate Studies in Statistics

SRI DHARMASTHALA MANJUNATHESHWARA

COLLEGE (Autonomous)

UJIRE - 574240

Karnataka, INDIA

FEBRAURY 2023

Contents

1 Chapter 1
Introduction

Introduction 1.1

Most of the people ordered the food on Zomato in Bengaluru city. There are more than 12,000 of Restaurants in Bengaluru. It is not an easy task to Analyse the popularity of those restaurants. Many factors will affect the popularity of any restaurant such as locality, online order facility, menu item, dish liked, reviews etc. It is very important to display rating score for each restaurant, because every person will not read all reviews before he/she orders the food, but he/she definitely see the rating scores. So predicting the rating score of restaurants has huge scope in online food platforms like Zomato. In such case Analysing factors which affect the ratings is also very important. This is the business problem, and in order to solve this problem. I have chosen Zomato Bangalore dataset from Kaggle.

1.1.1 Literature review

Vicky Malik, S. Prasad Babu Vagolu, Sunil Chandolu in April 2020 published a paper named Restaurants Rating Prediction using Machine Learning Algorithms. The aim of this paper is to find out the relationship between the dependent and independent variable using Regression. Based on various attributes like the food, quality, prize ambience of the restaurant it predicts the Restaurant Rating. This paper studies a number of features about existing restaurants of different areas in a city and analyses them to predict rating of the restaurant. This makes it an important aspect to be considered, before making a dining decision. Such analysis is essential part of planning before establishing a venture like that of a restaurant.

Bidisha Das Baksi, Harrsha P, Medha, Mohinishree Asthana, Dr. Anitha C in 2018 published a paper named Restaurant Market Analysis, which talks about restaurant market, location determination, predictive analysis, existing market conditions, customer satisfaction, regression algorithm. This paper studies various attributes of existing restaurants and analyses them to predict an appropriate location for higher success rate of the new restaurant. The study of existing restaurants in a particular location and the

growth rate of that location is important prior to selection of the optimal location.

1.1.2 Objectives

The objectives of this project are as below:

- To know How many Restaurants are accepting online orders
- To know how many restaurants provide book table facility.
- To know which location has the highest number of restaurants.
- To know which cuisines are most liked by bangaloreans.
- To determine if there is a difference in mean rating of restaurants with online order facility and without online order facility.
- To determine if there is a difference in mean rating of restaurants with book table facility and without book table facility.
- To know whether there is relationship between approx cost and ratings.
- Predicting the rating of restaurants based on reviews given by the users and other features like book table, online order, price, menu items.

1.1.3 Scope of study

Scope of study is limited towards those people who use Zomato service. From this study, we can have a better understanding of the Online Food Delivery Service Market. We will know about the consumer perception regarding the services they provide in that area and will get to know the variables affecting their perception. Therefore, these findings may help the service providers to work upon on these variables to fill up the gaps in the mindset of consumers.

2 Chapter 2

Methodology

2.1 Materials and methods

2.1.1 About the Data

A secondary data is collected from the ”Kaggle”. The data set contains 17 variables and 51,717 observations. Data is in csv format. Description about the variable considered in the analysis are given as follows:

- Url : This feature contains the url of the restaurant on the Zomato website.
- Address : This feature contains the address of the restaurant in Bangalore.
- Name : This feature contains the name of the restaurant.
- Onlineorder : Whether online ordering is available in the restaurant or not.

- Booktable : Table book option available or not • Rate : contains the overall rating of the restaurant out of 5.
- Votes : Contains total number of upvotes for the restaurant.
- Phone : Contains the phone number of the restaurant.
- Location : Contains the neighborhood in which the restaurant is located.
- Resttype : Restaurant type.
- Dishliked : Dishes people liked in the restaurant.
- Cuisines : Food styles, separated by comma.
- Approxcost(for two people): Contains the approximate cost of meal for two people.
- Reviewslist : List of tuples containing reviews for the restaurant, each tuple consists of two values, rating and review by the customer.
- Menuitem : Contains list of menus available in the restaurant.
- Listedin(type) : Type of meal.
- Listedin(city) : Contains the neighborhood in which the restaurant is located.

This data set contains Different types of variables are Categorical and numerical features.

- Categorical features: These features have categories (Address, name, online order, location, book table, rest type, dish liked, cuisines ,menu item, listed in).
- Numerical features: These features have numerical values (rate, votes, approx cost)

2.1.2 Tools used Python:

Python is the programming language used for analysis and building model. Numpy, Pandas, Matplotlib, Seaborn, Scikit learn are the main python libraries that are used in this projects.

2.1.3 Statistical/ML techniques

• T-test: A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features. It can be used to check relationship between categorical predictor variable and numerical target variable. The formula for the two-sample t-test (the Student's t-test) is shown below.

T-test formula:

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{(s^2(\frac{1}{n_1} + \frac{1}{n_2}))}}$$

In this formula, t is the t-value, x1 and x2 are the means of the two groups being compared, s 2 is the pooled standard error of the two groups, and n1 and n2 are the number of observations in each of the groups. A larger t-value shows that the

difference between group means is greater than the pooled standard error, indicating a more significant difference between the groups.

• Bag of words: Bag of words is the most trivial representation of text into vectors. Each column of a vector represents a word. The values in each cell of a row show the number of occurrences of a word in a sentence. The values 7 corresponding to each word shows the number of occurrences of a word in a review. The bag-of-words model is a way of representing text data when modelling text with machine learning algorithms. The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modelling and document classification. Text data is used in natural language processing (NLP), which interacts between humans and machines using natural language. Text data helps analyse reviews. Text reviews provided by the customers are of different lengths. By converting from text to numbers, we can represent a review by a finite length of the vector. In this way, the length of the vector will be equal for each review, irrespective of the text length.

3 Chapter 3

Result and Discussion

3.1 Data Cleaning

The complete data contains 51717 data instances and 17 columns. Data is in the text format. So, it is mandatory to clean the data before analysing.

Text data is cleaned using the below steps:

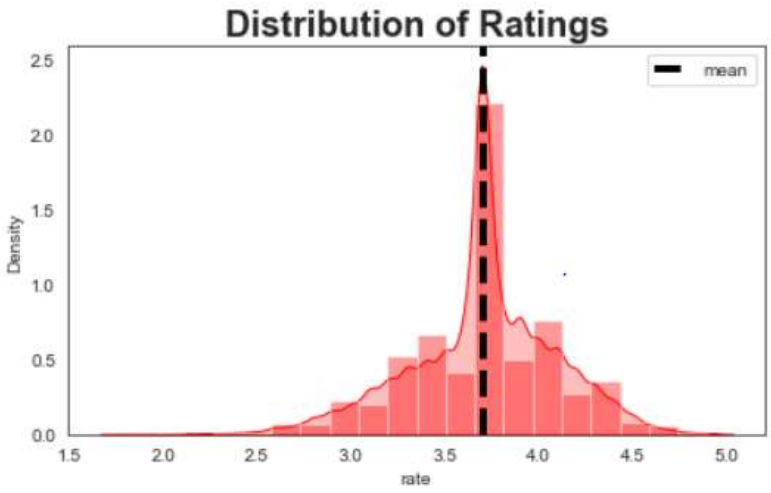
1. All stop words are removed.
2. All text datas converted to lower case
3. Unwated underscpres hyphens and (-) are removed
4. Unwanted symbols such as (/),(.),(,) are removed

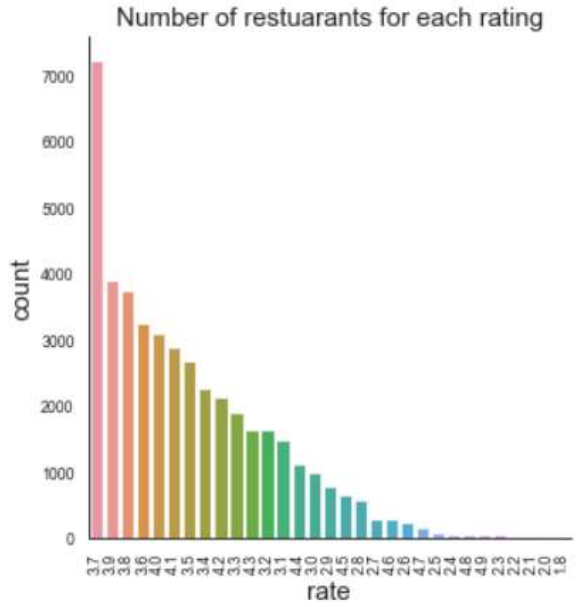
3.2 Data Analysis

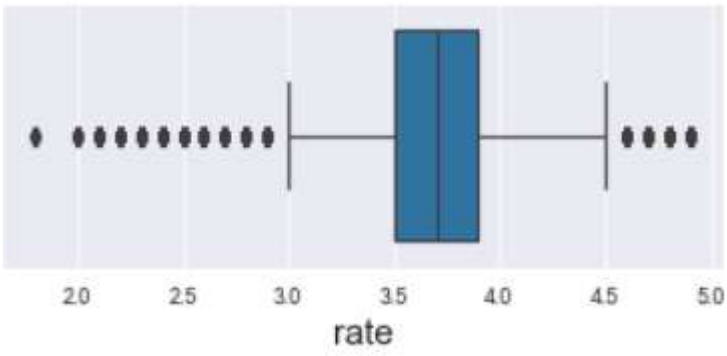
3.2.1 Exploratory Data Analysis

The first step is to analyse Distribution of Target variable (Rating).

Below plot shows the distribution of Ratings of restaurants in Bengaluru.:





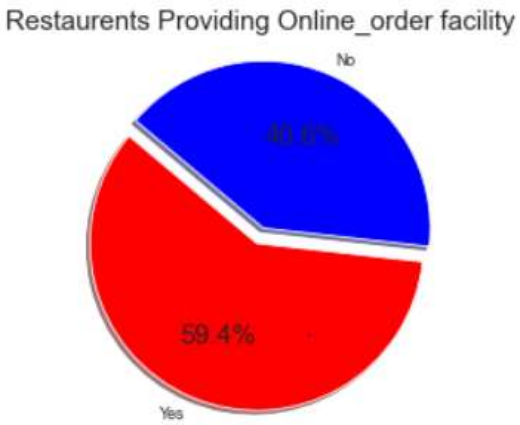


Here 50% of the rate distribution lies between 3.5 and 4.0 with an average rating of 3.7. Restaurants with rating higher than 4.5 are very rare. 3.7 is the most common rating.

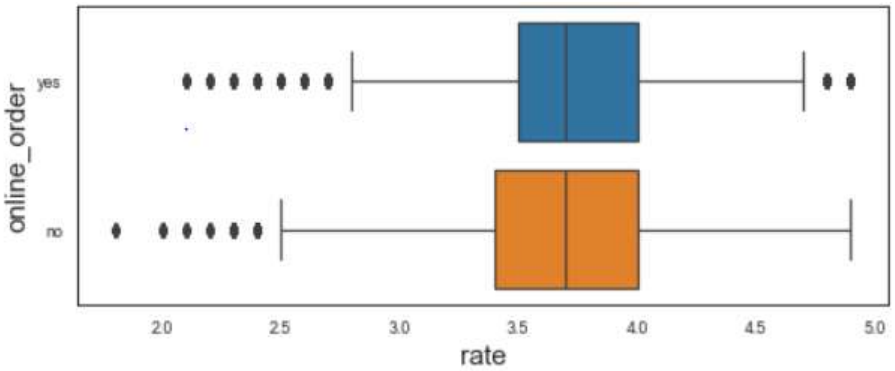
Analysing online order:

No of restaurants with online delivery: 30273 (i.e, 59.4%)

No of restaurant without online delivery: 20733 (i.e , 40.6%)



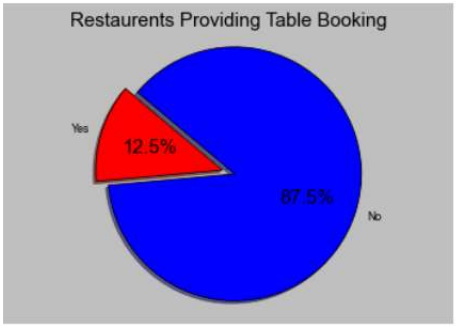
Here 59.4% of the restaurants provide online booking facility and 40.6% of hotels don't provide this facility.



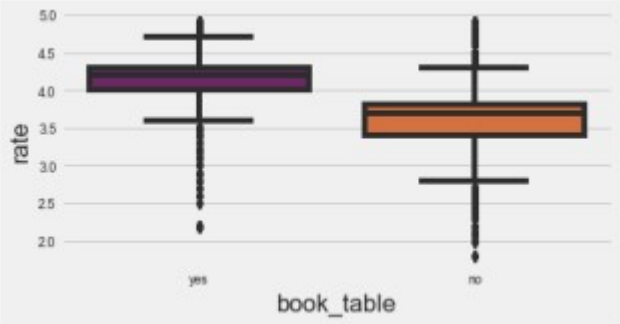
Restaurants are likely to get better rating if they provide online order But it can also be consequence that because Zomato offers home delivery for online orders also, so more people will give rating for online_order restaurants on their website.

Analysing book table facility:

No of restaurants having book table facility: 6391 (i.e , 12.5%)
No of restaurants having book table facility: 44615 (i.e , 87.5%)

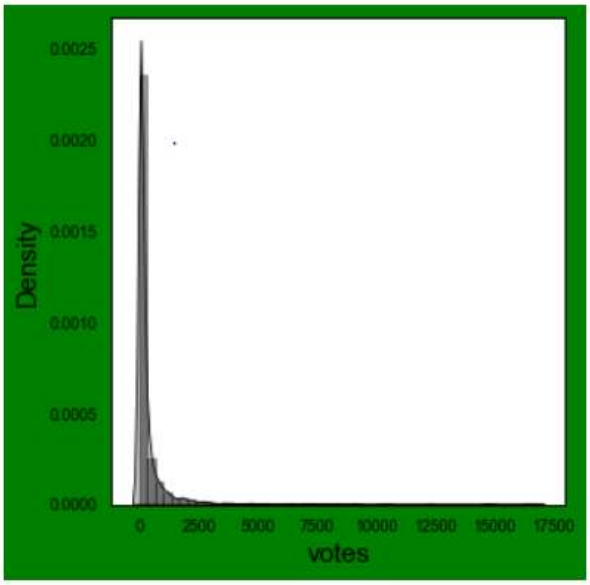


Here 12.5% of Restaurants provide book table facility and remaining 87.5% of reastaurants don't provide book table facility.

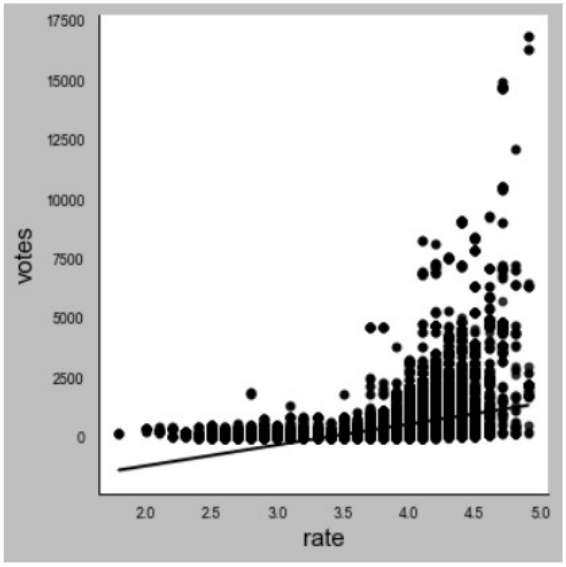


Restaurants are more likely to receive a higher rating if it offers table book table option.lower tail of Box plots which represents the restaurants that provide book table in advance, is greater than the median of the ratings of the restaurants that don't book table in advance.

Analysis of votes:

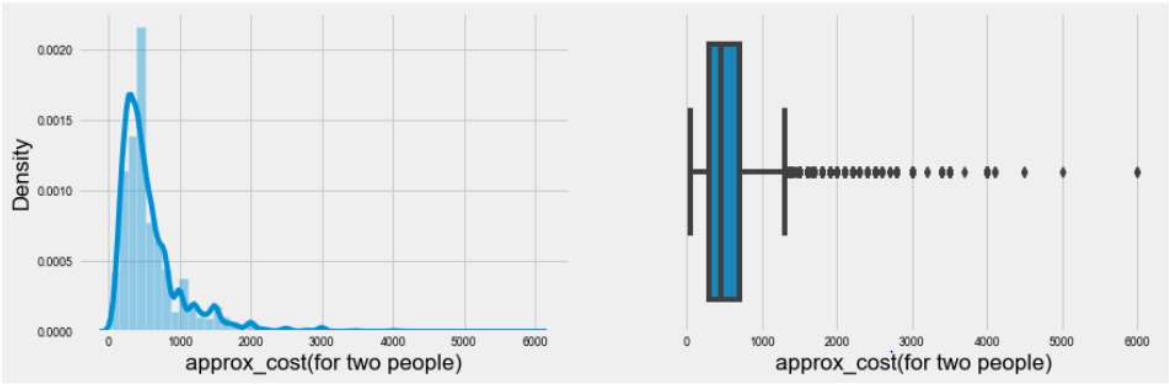


There are very few restaurants with high votes. Density is high at lower votes.



Restaurants with lower votes has lower ratings and restaurants with more votings are likely to have better ratings.

Aproximate cost for two people:



Distributin is right skewed. Aproximate cost for two people is less than 1000 in most of incidents.Thats most of the restaurants provides good dining for two people for less than 1000rs. On average cost lies between 300-500.

Analysis of location:

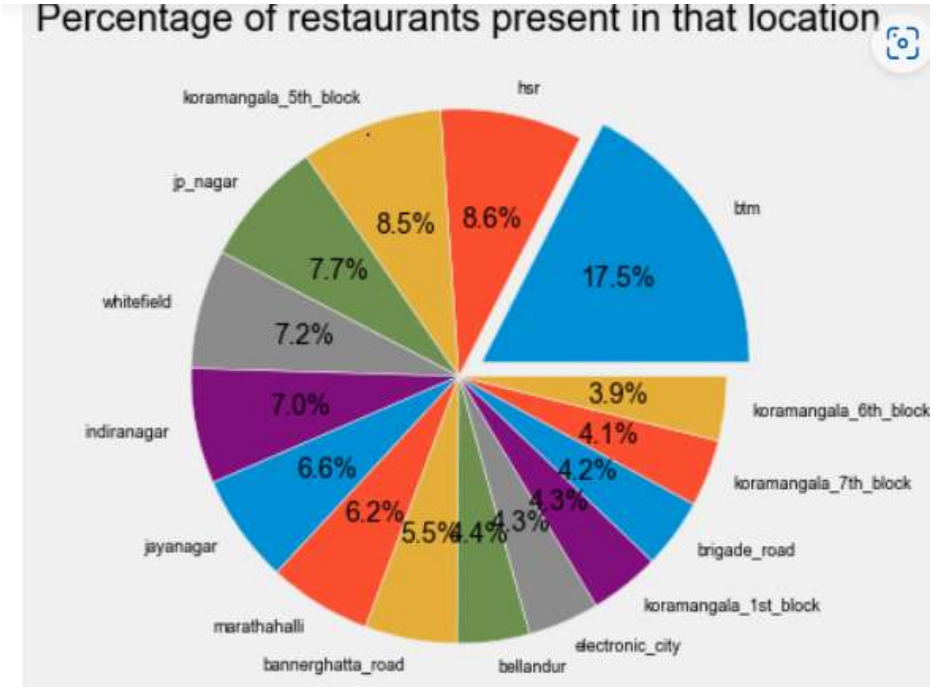
location	approx_cost(for two people)
sankey_road	2526.000000
lavelle_road	1352.561475
race_course_road	1344.800000
mg_road	1211.015912
residency_road	1007.592892

location	approx_cost(for two people)
cv_raman_nagar	318.666667
south_bangalore	323.626374
city_market	324.951220
north_bangalore	325.000000
yelahanka	325.000000

Sankey road ,levelle road and race course road are costliest area for customers.Cv.Raman Nagar is cheaper.

location	rate
lavelle_road	4.128484
st._marks_road	4.027811
koramangala_3rd_block	4.008081
koramangala_5th_block	3.999025
church_street	3.990388
sankey_road	3.952000

Lavelle road has restaurants with good rating but its also one of the costliest area.



Btm layout has highest no of restaurants

Analysing cuisines:

cuisines	
cuisines	
north_indian	2306
north_indian chinese	1997
south_indian	1348
bakery desserts	690
biryani	613
south_indiannorth_indian chinese	609
cafe	605
desserts	605

North Indian, chines southindian and desert are the most liked foods.

3.2.2 Conclusion

- Average rating of restaurants in Banglore is 3.7 .
- 65% of restaurants provide online order facility and remaining 35% don't provide online order facility.
- 14.5% of restaurants provide book table facility and remaining 85.5% don't provide book table facility.
- There are very few restaurants with high upvotes, density is high at lower votes.
- North Indian, Chinese and southindian are most liked cuisenes.

- In most of the cases aproximate cost for two people is less 1000 . On an average it lies between 300 to 500.
- Btm layout has highest no of restaurants.
- Lavelle road has restaurants with highest rating (on average 4.1).

3.3 Data splitting:

Data is split into train and test in the ratio 80:20 .
From splitted data, independent and target variables are considered separately. y denotes the target variable,x denotes the independent variables data matrix.
Shape of x train: (40804, 13)
Shape of y train : (40804,1)
Shape of x test : (10202, 13)
Shape of y test : (10202,1)

3.4 Statistical Tests

3.4.1 T-test

Objective: To determine if there is a difference in mean rating of restaurants with online book facility and without online book facility.
Null hypothesis (H0): There is no difference in mean ratings between restaurants which are having online order facility and not having online order facility.
Alternative hypothesis (H1): There is a difference in mean ratings between restaurants which are having online booking facility and not having online book facility.
Significance level = 5%.
statistic=13.797
pvalue=3.1507216314588605e-43

Conclusion:

p-value is less than 0.05, hence we reject null hypothesis and accept the alternative hypothesis. We conclude that there is a difference in mean ratings between restaurants which are having online booking facility and not having online book facility.

3.4.2 T-test

Objective:To determine if there is a difference in mean rating of restaurants with book table facility and without book table facility.
Null hypothesis (H0): There is no difference in mean ratings between restaurants which are having book table facility and not having book table facility.
Alternative hypothesis (H1): There is a difference in mean ratings between restaurants which are having book table facility and not having book table facility.
Significance level = 5%.
statistic=102.77380626812825, pvalue=0.0

Conclusion: p-value is less than 0.05, hence we reject null value and accept alternative hypothesis. We conclude that there is a difference in mean ratings between restaurants which are having book table facility and which do not have book table facility.

3.5 Predictive models:

Rating of restaurants is the target variable which need to be predicted.
The independent data contains numerical variables, categorical variables as well as text data.
Handling these data includes

- Standardizing numerical variables.
- Converting categorical variables into vectors by using CountVectorizer().
- Converting text to vectors by using Bag of Words method (using Countvectorizer tool).

After completing all these steps final shape of datasets are as given below:
x_train:(40804, 134822)
y train:(40804 ,)
x test:(10202, 134822)
y test:(10202 ,)
Result of prediction model is given below:

Model Name	R2 value
Linear Regression	0.63694
Decision Tree regression	0.77933
Random Forest Regression	0.81754

Here Random Forest regression model performed better compared to other two modles.

4 Chapter 4

Conclusion

Following are the over all conclusion

- Difference in mean rating of restaurants that provide online order facility and mean rating of restaurants that do not provide online order facility is statistically significant.
- Difference in mean rating of restaurants that provide book table facility and mean rating of restaurants that do not provide book table facility is statistically significant.
- Realtion between approximate cost(for two people) and ratings is significant.
- North Indian, Chinese and southindian are most liked cuisenes.
- 65% of restaurants accepts online order and remaining 35% don't accepts online order.
- Btm layout has highest no of restaurants (17.5%).
- 14.5% of restaurants provide book table facility and remaining 85.5% don't provide book table facility.
- Among the predctive models Random forest regression with (n_estimator=100) performed very well with R2 Score = 0.82.

5 Summary

Aim of this project is to analyse the different factors (features) that effect the ratings of the restaurants and predict the ratings of the restaurants by using the analysed features. Data set on which we performed analysis is based customer review. From analysis its found that approximate cost for two people , book table facility and online order facility affects the ratings of restaurants.
Among three predictive that we used Linear regression model doesn't perform well and decision tree regression performance is quite good but Random forest regression performed really well with R^2 score =0.82 this model can be further trained well by tuning hyper parameter and parameter.

6 Bibliography

- https://www.kaggle.com/himanshupoddar/zomato-bangalore-restaurants
- Vicky Malik, S. Prasad Babu Vagolu, Sunil Chandolu (2020) : Restaurants Rating Prediction using Machine LearningAlgorithms.
- Bidisha Das Baksi, Harrsha P, Medha, Mohinishree Asthana (2018): Restaurant Market Analysis.
- https://matplotlib.org/stable/contents.html
- https://pandas.pydata.org/docs/
- https://scikit-learn.org/stable/

7 Appendix

Python Code

```
# Importing Libraries:
import numpy as np  #NumPy is a general-purpose array-processing package.
import pandas as pd  #It contains high-level data structures and manipulation tools designed to make data analysis fast and easy
import matplotlib.pyplot as plt  #It is a Plotting Library.
import re
import seaborn as sns  #Seaborn is a Python data visualization library based on matplotlib.
from sklearn.linear_model import LinearRegression  #Linear Regression is a regression algorithm.
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split #Splitting of Dataset.
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score
from sklearn import neighbors
import warnings
```

```
warnings.filterwarnings("ignore")

dt=pd.read_csv("cleaned_data.csv")
dt.isnull().sum()
dt['reviews_list'].fillna('no reviews',inplace=True)
dt.dropna(inplace=True,axis=0)
dt.isnull().sum()
#information about data
dt.info()
#All unique ratings
dt.rate.unique()
# Distribution of Ratings of restaurants in Bengalore.
fig = plt.figure(figsize=(7,4))

sns.set_style('white')
sns.distplot(dt['rate'], bins = 20,  color= 'red',kde_kws={"shade": True});
plt.axvline(x= dt.rate.mean(),ls='--',color='black',linewidth=4,label="mean")
plt.title("Distribution of Ratings",fontweight='bold',fontsize=20);
plt.legend(["mean"],prop={"size":10});

sns.set_context("paper",font_scale=1,rc={"font.size": 15,"axes.titlesize": 15,"axes.labelsize":
15})
b=sns.catplot(data=dt,kind='count',x='rate',order=dt['rate'].value_counts().index)
plt.title("Number of restuarants for each rating")
b.set_xticklabels(rotation=90)
plt.show()

fig = plt.figure(figsize=(12,7))
ax6 = fig.add_subplot(3,2,3)
sns.boxplot(dt['rate'],ax=ax6)
plt.show()

print("1st quantile of rate is:",np.quantile(dt['rate'],0.25))
print('2nd quantile of rate is:',np.quantile(dt['rate'],0.5))
print('3rd quantile of rate is:',np.quantile(dt['rate'],0.75))
print('4th quantile of rate is:',np.quantile(dt['rate'],1))

sl=[((dt.rate>=1.5)&(dt.rate<2)).sum(),((dt.rate>=2)&(dt.rate<2.5)).sum(),
   ((dt.rate>=2.5)&(dt.rate<3)).sum(),((dt.rate>=3)&(dt.rate<3.5)).sum(),
    ((dt.rate>=3.5)&(dt.rate<4)).sum(),((dt.rate>=4)&(dt.rate<4.5)).sum(),
    ((dt.rate>=4.5)&(dt.rate<5)).sum()]

Sl
labels=['1.5-2','2-2.5','2.5-3','3-3.5','3.5-4','4-4.5','4.5-5']
```

```
colors = ['Red','blue','Green','yellow','indigo','pink']
plt.pie(sl,colors=colors, labels=labels, autopct='%1.0f%%', pctdistance=0.5,
labeldistance=1.3)
fig = plt.gcf()
plt.title("Percentage of Restaurants according to their ratings", bbox={'facecolor':'1',
'pad':5})

fig.set_size_inches(10,10)
plt.show()

print("No of restaurants with online delivery:",(dt['online_order']=='yes').sum())
print("No of restaurants with no online delivery:",(dt['online_order']=='no').sum())

dt['online_order'].value_counts(normalize=True)*100
dt.groupby('rate').online_order.value_counts().unstack()

z=dt["online_order"].value_counts()
labels = 'Yes', 'No'
sizes = [z.yes, z.no]
colors = ['red', 'blue']
explode = (0.1, 0,)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)
plt.title("Restaurents Providing Online_order facility")
plt.axis('equal')
plt.show()

fig = plt.figure(figsize=(8,3))
ax1 = fig.add_subplot(1,1,1)
sns.boxplot(x=dt['rate'],y=dt['online_order'])

fig = plt.figure(figsize=(10,5))
fig.patch.set_facecolor('green')
plt.style.use('grayscale')

plt.subplot(121)
sns.distplot(dt['votes'],kde_kws={"shade": True})

#Linear Relationship between rate and votes shown below:
sns.lmplot(x="rate",y="votes", data=dt);

print('No of restaurants having book table facility:',(dt['book_table']=='yes').sum())
print('No of restaurants having book table facility:',(dt['book_table']=='no').sum())
```

```
z=dt["book_table"].value_counts()
labels = 'Yes', 'No'
sizes = [z.yes, z.no]
colors = ['red', 'blue']
explode = (0.1, 0,)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)
plt.title("Restaurents Providing Table Booking")
plt.axis('equal')
plt.show()

# relation between table booking option and rating of the restaurant
plt.rcParams['figure.figsize'] = (7,3)
Y = pd.crosstab(dt['rate'], dt['book_table'])
Y.div(Y.sum(1).astype(float), axis = 0).plot(kind = 'bar', stacked = True,color=['red','blue'])
plt.title('table booking vs Normal rate', fontweight = 30, fontsize = 20)
plt.legend(loc="upper right")
plt.show()


plt.style.use('fivethirtyeight')
pd.crosstab(dt.rate,dt.book_table).plot(kind='line',marker='o',figsize=(7,3));
plt.title("Ratings vs bookTable");

# OnlineOrder Vs ApproxCost wrt book Table
fig = plt.figure(figsize=(15,5))
fig.patch.set_facecolor('mediumorchid')
plt.style.use('fivethirtyeight')

plt.subplot(122)
sns.boxenplot(data=dt,x='online_order',y='approx_cost(for two people)',hue='book_table');
plt.title("OnlineOrder Vs ApproxCost wrt book Table",fontweight='bold',fontsize=15);

# Lets look at distribution of Location Variable
g = sns.countplot(x="location",data=dt, palette = "Set1",order =
dt['location'].value_counts()[:10].index)
g.set_xticklabels(g.get_xticklabels(), rotation=90, ha="right")
g
plt.title('locality',size = 10)
fig = plt.gcf()
fig.set_size_inches(10,5)

plt.figure(figsize=(6,6))
names = dt.location.value_counts()[:15].index
```

```
values = dt.location.value_counts()[:15].values
explode = [0.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

plt.pie(values, explode=explode, autopct='%0.1f%%',  labels = names)
plt.title("Percentage of restaurants present in that location")
plt.show()

#location and rating
dt.groupby('location')['rate'].mean().sort_values(ascending=False).head(10)

loc_plot=pd.crosstab(dt['rate'],dt['listed_in(city)'])
loc_plot.plot(kind='bar',stacked=True);
plt.title('Location - Rating',fontsize=15,fontweight='bold')
plt.xlabel('Rate',fontsize=10,fontweight='bold')
plt.xticks(fontsize=10,fontweight='bold')
plt.yticks(fontsize=10,fontweight='bold');
plt.legend().remove();

pd.DataFrame(dt.groupby('location')['rate'].mean().sort_values(ascending=True).head(10))
dt.groupby("location")['rate'].median().sort_values(ascending=False)

a=pd.DataFrame(dt.groupby('location')['approx_cost(for two people)'].mean().sort_values())
a
#analysing cuisines
plt.figure(figsize=(10,5))
a=dt.cuisines.value_counts()[:15]
sns.barplot(x=a,y=a.index)
plt.title('Most liked cuisins')
plt.xlabel('cuisines_count')

pd.DataFrame(dt.groupby('cuisines')['cuisines'].agg('count').sort_values(ascending=False).he
ad(10))

fig=plt.figure(figsize=(13,14))
ax4=fig.add_subplot(3,2,1)
ax5=fig.add_subplot(3,2,2)
sns.distplot(dt['approx_cost(for two people)'],ax=ax4)
sns.boxplot(dt['approx_cost(for two people)'],ax=ax5)
plt.show()

#Top 5 costliest location in bangalore
dt.groupby('location')['approx_cost(for two
people)'].mean().sort_values(ascending=False).head(5)
```

```
pd.DataFrame(dt.groupby('location')['approx_cost(for two
people)'].mean().sort_values(ascending=True).head(5))

sns.countplot(dt['approx_cost(for two people)'])
```

To dtermine there is a relation between online order and ratings¶

H0: There is no difference in mean rating of restaurasnts that provide online order facility and of that restaurants that dont provide online book facility

H1:There is difference in mean rating of restaurasnts that provide online order facility and of that restaurants that dont provide online book facility

```
from scipy import stats

yes=dt[dt['online_order']=='yes']

no=dt[dt['online_order']=='no']

stats.ttest_ind(yes['rate'],no['rate'])
```

H0: There is no difference in mean rating of restaurasnts that provide online order facility and of that restaurants that dont provide online book facility

H1:There is difference in mean rating of restaurasnts that provide online order facility and of that restaurants that dont provide online book facility

```
yes=dt[dt['book_table']=='yes']
no=dt[dt['book_table']=='no']
stats.ttest_ind(yes['rate'],no['rate'])

#split to train test
x=dt.iloc[:,[0,1,2,4,5,6,7,8,9,10,11,12,13]]
y=dt.iloc[:,3]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)

#Bow
from sklearn.feature_extraction.text import CountVectorizer
vec=CountVectorizer()

#Book order
x_train_order=vec.fit_transform(x_train['online_order'].values)
```

```
x_test_order=vec.transform(x_test['online_order'].values)
print(x_train_order.shape)
print(x_test_order.shape)

#Book table

x_train_booktable=vec.fit_transform(x_train['book_table'].values)
x_test_booktable=vec.fit_transform(x_test['book_table'].values)

print(x_train_booktable.shape)
print(x_test_booktable.shape)

#location
vec=CountVectorizer()
x_train_location=vec.fit_transform(x_train['location'].values)
x_test_location=vec.transform(x_test['location'].values)
print(x_train_location.shape)
print(x_test_location.shape)

#rest_type
vec=CountVectorizer()
vec.fit(x_train['rest_type'].values)
x_train_resttype=vec.transform(x_train['rest_type'].values)
x_test_resttype=vec.transform(x_test['rest_type'].values)
print(x_train_resttype.shape)
print(x_test_resttype.shape)

#dish liked
vec=CountVectorizer()
x_train_dishliked=vec.fit_transform(x_train['dish_liked'].values)
x_test_dishliked=vec.transform(x_test['dish_liked'].values)
print(x_train_dishliked.shape)
print(x_test_dishliked.shape)

#cuisines
vec=CountVectorizer()
x_train_cuisines=vec.fit_transform(x_train['cuisines'].values)
x_test_cuisines=vec.transform(x_test['cuisines'].values)
print(x_train_cuisines.shape)
print(x_test_cuisines.shape)

#menu_item
vec=CountVectorizer()
vec.fit(x_train['menu_item'].values)
```

```
x_train_menuitem=vec.transform(x_train['menu_item'].values)
x_test_menuitem=vec.transform(x_test['menu_item'].values)
print(x_train_menuitem.shape)
print(x_test_menuitem.shape)

#listed_in(type)
x_train_listedtype=vec.fit_transform(x_train['listed_in(type)'].values)
x_test_listedtype=vec.transform(x_test['listed_in(type)'].values)

#listed_in(city)
x_train_listedcity=vec.fit_transform(x_train['listed_in(city)'].values)
x_test_listedcity=vec.transform(x_test['listed_in(city)'].values)

#dish_liked
x_train_reviewlist=vec.fit_transform(x_train['dish_liked'].values)
x_test_reviewlist=vec.transform(x_test['dish_liked'].values)


#normalization
from sklearn.preprocessing import StandardScaler
scale=StandardScaler()

# votes
#finnding mean and stdvtn using train data
scale.fit(x_train['votes'].values.reshape(-1,1))

#normalising testand train by estimates
x_train_votes=scale.transform(x_train['votes'].values.reshape(-1,1))
x_test_votes=scale.transform(x_test['votes'].values.reshape(-1,1))
print(x_train_votes.shape)

#aprox_cost
#finding mean and stddvtn using test data
scale.fit(x_train['approx_cost(for two people)'].values.reshape(-1,1))
#standardizing bot test and train using paramters estimated from train
x_train_cost=scale.transform(x_train['approx_cost(for two people)'].values.reshape(-1,1))
x_test_cost=scale.transform(x_test['approx_cost(for two people)'].values.reshape(-1,1))
print(x_test_cost.shape)

#Joining all features
from scipy.sparse import hstack
x_trn=hstack((x_train_order,x_train_booktable,x_train_location,x_train_resttype,x_train_di
shliked,x_train_cuisines,x_train_menuitem,x_train_listedtype,x_train_listedcity,x_train_revi
ewlist)).tocsr()
```

```
x_tst=hstack((x_test_order,x_test_booktable,x_test_location,x_test_resttype,x_test_dishlik
ed,x_test_cuisines,x_test_menuitem,x_test_listedtype,x_test_listedcity,x_test_reviewlist)).t
ocsr()
print(x_trn.shape , y_train.shape)
print(x_tst.shape,y_test.shape)

#Model building

#Lineear regression
lr=LinearRegression()
lr.fit(x_trn,y_train)
y_predct=lr.predict(x_tst)
print(r2_score(y_test,y_predct,multioutput='uniform_average'))

#Decision tree regressor
dcsn=DecisionTreeRegressor()
dcsn.fit(x_trn,y_train)
dcsn_predict=dcsn.predict(x_tst)
print(r2_score(y_test,dcsn_predict,multioutput='uniform_average'))

#Randomforest
rnd=RandomForestRegressor(random_state=0 , n_estimators=100)
rnd.fit(x_trn,y_train)
random_prdct=rnd.predict(x_tst)
print(r2_score(y_test,random_prdct,multioutput='uniform_average'))
```