A PROJECT ON

"RESTAURANT RECOMMENDATION AND RATING PREDICTION SYSTEM"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYSIS

Centre for Development of Advanced Computing

SUBMITTED BY:

Anuj Kumar

UNDER THE GUIDENCE OF:
Mr. Abhijeet das
Project Engineer
CDAC Delhi.

CERTIFICATE

This is to certify that the project work under the title ‘Restaurant Recommendation And Rating Prediction System’ is done by Anuj kumar in partial fulfillment of the requirement for award of Diploma in Big Data Analysis Course.

Mr. Abhijeet das **Project Guide**	**Mr. Ankit Khurana** **Course Co-Ordinator**

Date: 15-Sep-2022

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Ankit khurana (Center Coordinator, CDAC, DELHI) and Project Guide Mr. Abhijeet das.

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Centre for development of advance computing , Delhi for their support.

Anuj kumar
DBDA March 2022
Batch,CDAC Delhi

TABLE OF CONTENTS

1 Introduction

1.1 Introduction And Objectives:

The rapid growth in data collection has led to a new era of a data-driven world. Data is used to create more efficient systems and that’s where recommender systems come in.
Recommendation systems are a type of information filtering systems because they improve the quality of search results and provide elements that are more relevant to the search item and the idea behind the rating prediction is let us assume the online food app, has reached out to you to help them to predict how good or bad a restaurant will turn out in the future. So that, they can take a decision to include the restaurant in their app or remove it.

1.2 Why this problem needs to be Solved?

The rapid growth in data collection has led to a new era of a data-driven world. Data is used to create more efficient systems and that’s where recommender systems come in.

Recommendation systems are a type of information filtering systems because they improve the quality of search results and provide elements that are more relevant to the search item or that are related to the search history of the user.

These are active information filtering systems that personalize the information provided to a user based on their interests, relevance of the information, etc. Recommendation systems are widely used to recommend movies, items, restaurants, places to visit, items to buy, etc.

Due to online recommendation systems comes into picture there is another point to do business with those restaurants who are providing better service to consumers to get maximum profit and usage of online recommending systems, and that’s where rating prediction system come in.

To predict how good or bad a restaurant will turn out in the future. So that, they can take a decision to include the restaurant in their app or remove it.

1.3 Dataset Information.

India all Restaurants details.csv

sno – Serial number
Zomato URL – Zomato URL
Name – Restaurant names
City – City Names
Area–Area name under city
Rating –Ratings for a particular restaurant
Rating count– Total no of Rating count given by customers

Telephone – Contact details
Cuisine- The variety cuisine available in restaurant
Cost for two- Average cost for two people
Price range – The numbers tell that from which cost range restaurant comes in
Address - Restaurant full address
Co-ordinates – Geolocation of restaurant
Timing – Opening and closing time of restaurant
Online order – Online order is available or not
Table reservation- Reservation is available or not
Delivery- Delivery is available or not
Famous food– The food items which are famous and in demand

2 Problem Definition and Algorithm:

2.1 Problem Definition

The problem is quite straightforward. Data from Zomato stores across the India is given, and it giving the detail information about restaurants . The data is already split into a training and a test set, and we want to fit a model to the training data that is able to predict the rating of restaurants as accurately as possible. In fact, our metric of interest will be the and R2 score value. The metric is not very complicated. The further away from the actual outcome our prediction is, the harder it will be punished. Optimally, we exactly predict the rating. This of course is highly unlikely, but we must try to get as close as possible.

2.2 Algorithm Definition

Multiple Linear regression: is one of the very basic forms of machine learning where we train a model to predict the behavior of your data based on some variables. In the case of linear regression as you can see the name suggests Multiple linear that means shows the relationship between the dependent variable and multiple (two or more) independent variables

Random forest: is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Decision Tree: algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).In Decision Trees, for predicting a class label for a record we start from

the root of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

XGBoost: or extreme gradient boosting is one of the well-known gradient boosting techniques (ensemble) having enhanced performance and speed in tree- based (sequential decision trees) machine learning algorithms. XGBoost was created by Tianqi Chen and initially maintained by the Distributed (Deep) Machine Learning Community (DMLC) group. It is the most common algorithm used for applied machine learning in competitions and has gained popularity through winning solutions in structured and tabular data. It is open- source software. Earlier only python and R packages were built for XGBoost but now it has extended to Java, Scala, Julia and other languages as well.

Adaboost: AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique that is used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances. Boosting is used to reduce bias as well as the variance for supervised learning.

KNN : K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set

3 Experimental Evaluation:

3.1 Methodology:

The objective of this project is to recommend the restaurant on the basis of provided specifications of the restaurant and predict the rating of restaurant. The data set is contained from Kaggle which was originally taken out from Zomato Media PVT LTD. and has contains 28 states of Indian cities restaurants data.

Loading in raw data

```
ind_hotels = pd.read_csv('./india_all_restaurants_details.csv')
        ind_hotels.columns
        ind_hotels.info()
        ind_hotels.describe
```

Preprocessing:

Checking whether the dataset having unique cities and ratings

```
ind_hotels['city'].unique()
ind_hotels['rating'].unique()

# Dropping the rows in rating columns which had the unique values such as ’NEW’ and ‘Nove’.
ind_hotels = ind_hotels.loc[ind_hotels.rating !='NEW'].reset_index(drop=True)
ind_hotels = ind_hotels.loc[ind_hotels.rating !='Nové'].reset_index(drop=True)
ind_hotels['rating'] = ind_hotels['rating'].astype(float)
ind_hotels['rating'].unique()

# Removing commas, full stops in ‘Cost’ column and turning the column into a data type float.
ind_hotels['cost'] = ind_hotels['cost'].astype(str) #Changing into string
ind_hotels['cost'] = ind_hotels['cost'].apply(lambda i: i.replace(',','.')) #replace ','
ind_hotels['cost'] = ind_hotels['cost'].astype(float) # Changing from str to Float

ind_hotels['price_range'] = ind_hotels['price_range'].astype(int) # Changing from float to int
ind_hotels['rating_count'] = ind_hotels['rating_count'].astype(int) # Changing from float to int
ind_hotels = ind_hotels.drop(ind_hotels[ind_hotels.cost < 20].index)
ind_hotels = ind_hotels.drop(ind_hotels[ind_hotels.rating_count==0].index)
ind_hotels = ind_hotels.drop(ind_hotels[ind_hotels.rating==0].index)


# Replacing true false with Yes NO only for recommendation system
ind_hotels.online_order.replace(('Yes','No'),(True, False),inplace=True)
ind_hotels.table_reservation.replace(('Yes','No'),(True, False),inplace=True)
ind_hotels.info()
```

The data had several missing values and needed to be cleaned. Since data is very bigger in size so we directly dropped the rows

```
ind_hotels.drop_duplicates(inplace=True)
ind_hotels.duplicated().sum()

ind_hotels.isnull().sum()
ind_hotels.dropna(how='any',inplace=True)
```

Replacing outliers for 'Average Cost for two'

```
ind_hotels['cost'][ind_hotels['cost']<15000].sort_values(ascending=False)
# Replacing outliers with nearest possible value
ind_hotels['cost'][ind_hotels['cost']>15000] = 12000
```

Replacing outliers for 'rating count'

```
# Finding nearest values to 4000 mark
ind_hotels['rating_count'][ind_hotels['rating_count']<20000].sort_values(ascending=False)
# Replacing outliers with nearest possible value
ind_hotels['rating_count'][ind_hotels['rating_count']>20000] =15653


# Replacing Yes NO with 1’s 0’s only for rating prediction system

from sklearn.preprocessing import LabelEncoder

# create an encoder
encoder = LabelEncoder()

ind_hotels['online_order'] = encoder.fit_transform(ind_hotels['online_order'])
ind_hotels['table_reservation'] = encoder.fit_transform(ind_hotels['table_reservation'])
ind_hotels['delivery_only'] = encoder.fit_transform(ind_hotels['delivery_only'])
ind_hotels
```

Flow Diagram : Fig.1.Flowchart for Rating Prediction

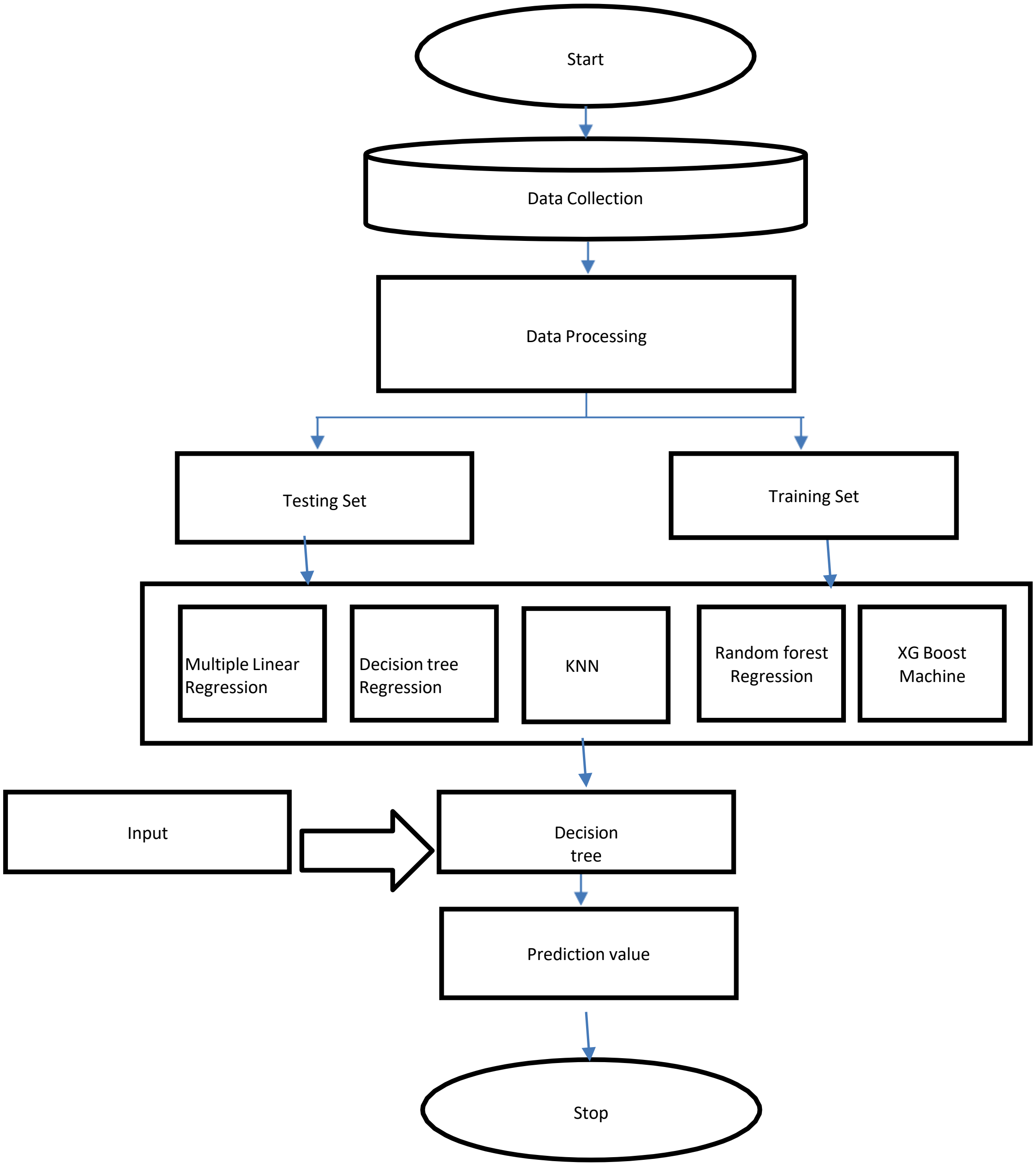
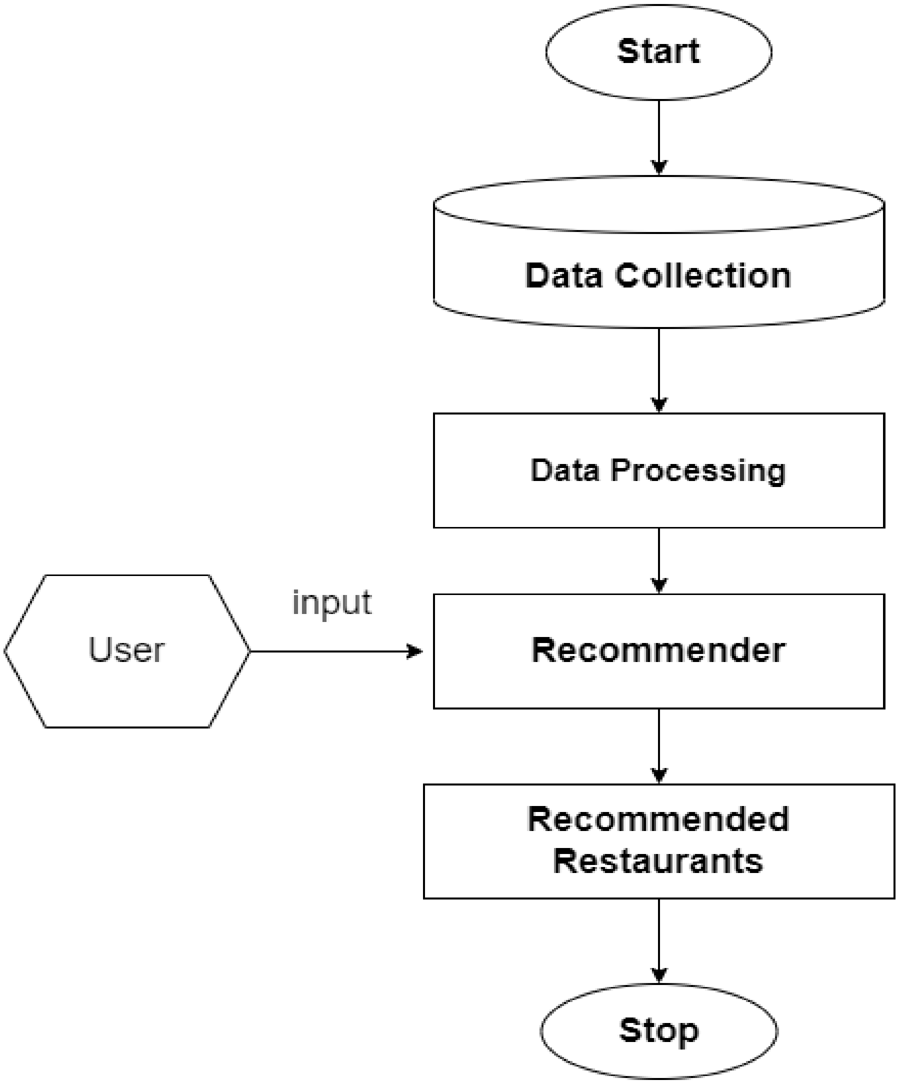


Fig.2.Flowchart for Recommendation system



3.2 Exploratory Data Analysis

The popularity of cuisines is plot with the help of a bar chart (fig 3). From the figure we can infer that North Indian food has the highest popularity after that comes Chinese Food , hyderabadi and wraps has the least popularity.

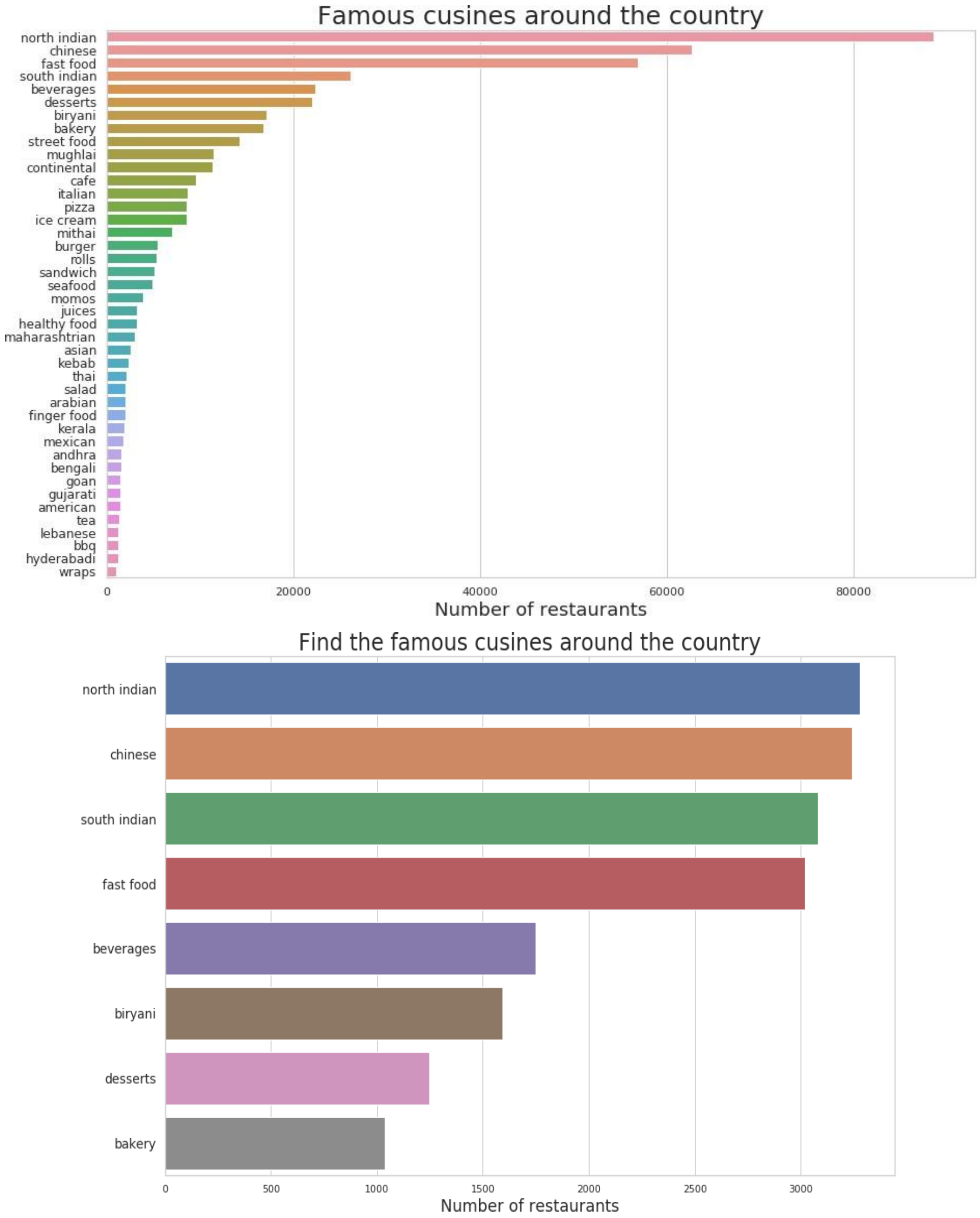


Fig.3 Famous cuisines around the country

the Indian restaurants around the country plotted with the help of heatmap (fig.4)From fig we can say that almost from each part of the Indian states has been covered in the given dataset

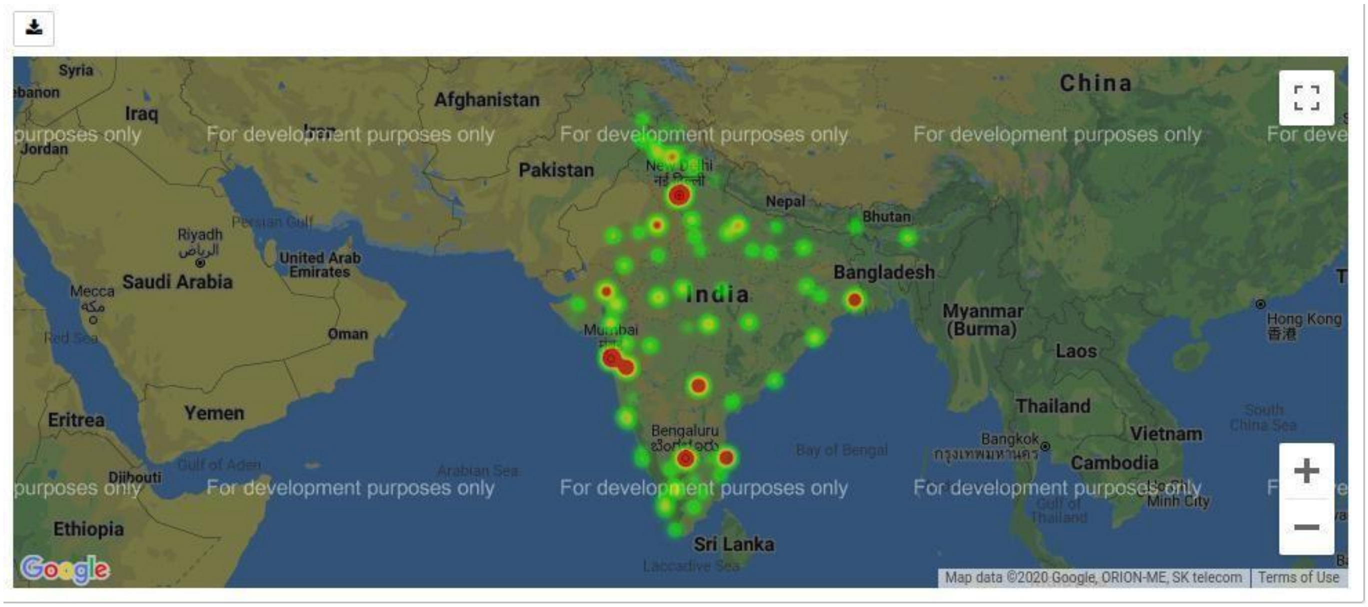


Fig.4 Heatmap of number of restaurants around the country

Restaurants distribution of top cities in India has plotted with the help of pie chart (Fig.5) We can say that Delhi NCR having total 25.8% of restaurants, and least is Chandigarh having 2.9%

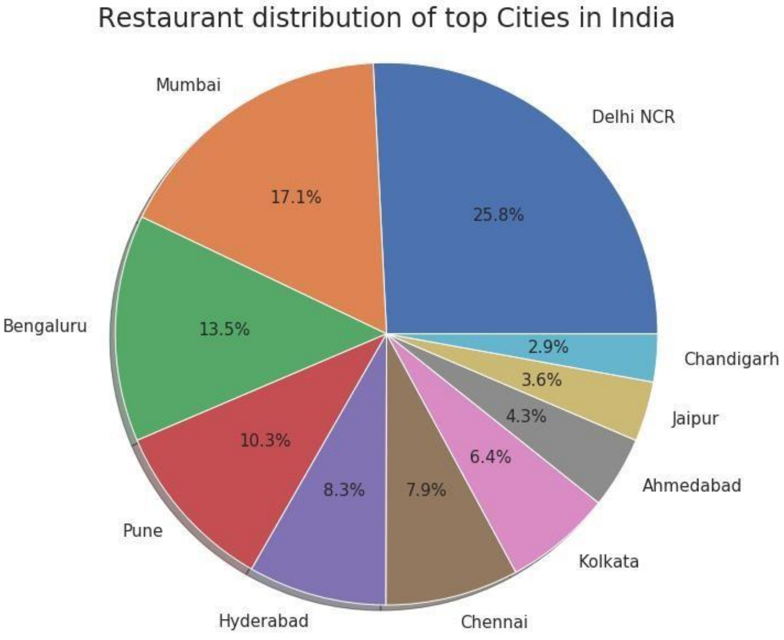


Fig.5

Famous restaurants chains in India as per the number of outlets in India plotted with the help of bar chart (Fig.6) We can say Café coffee Day having maximum outlets chaining present in the country and O’Biryani having least outlets across the country.

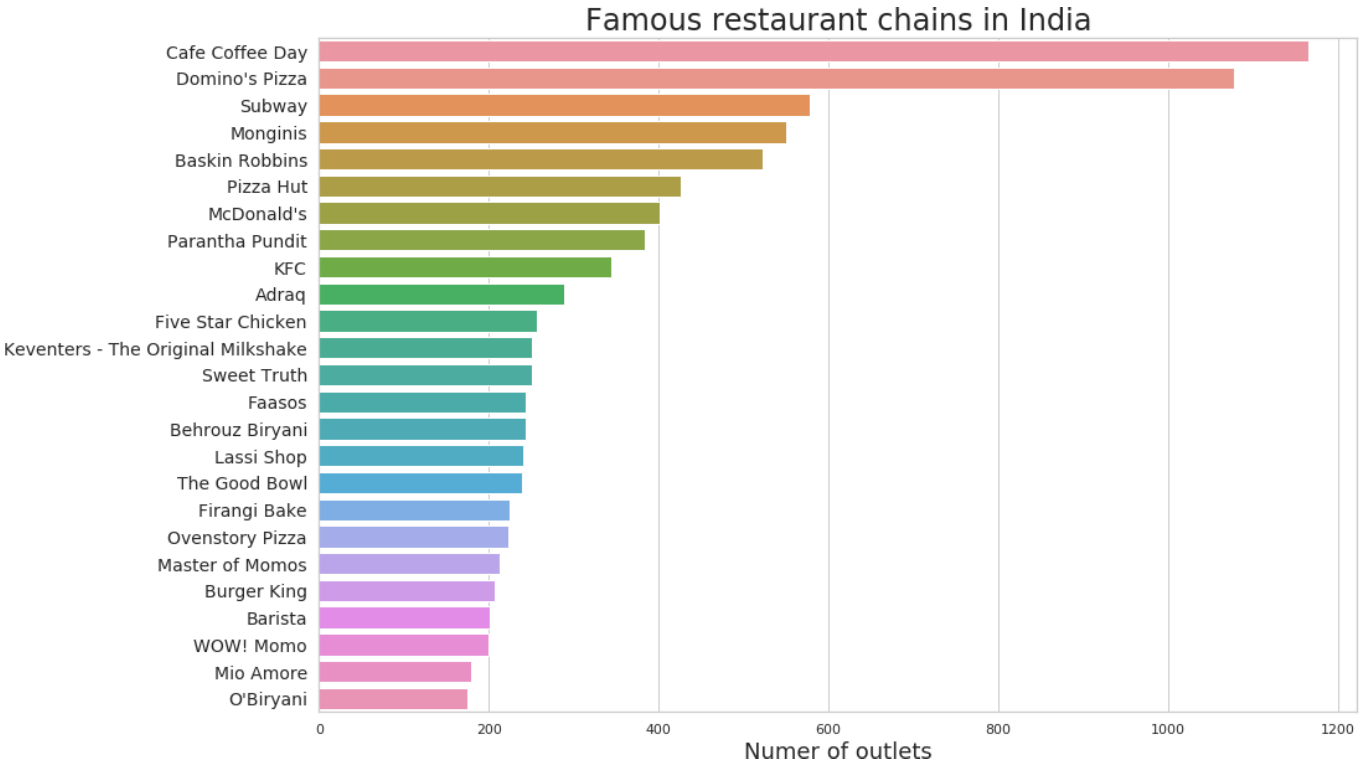


Fig.6 Famous restaurants chains in India as per the no of outlets

Rating distribution across the country plotted as per the (Fig.7) We can say we have normal distribution of ratings across the country

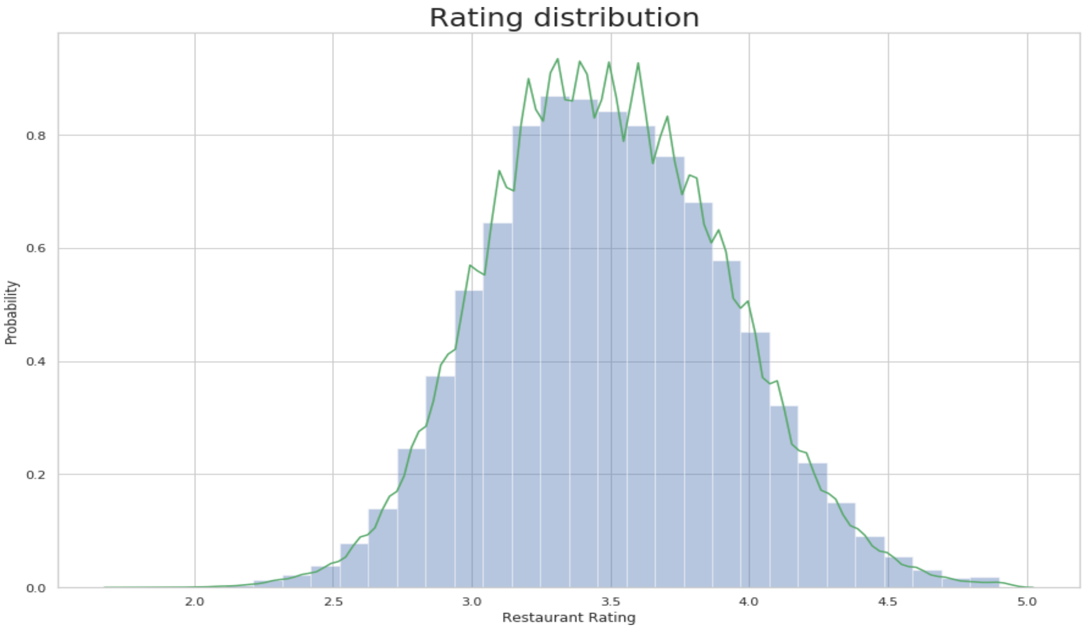


Fig. 7 Rating distribution across the country

Restaurants accepting table reservation or not plotted with the help of pie chart.(Fig.8). We can say that 97.1% of restaurants having table reservation, which is quite high number.

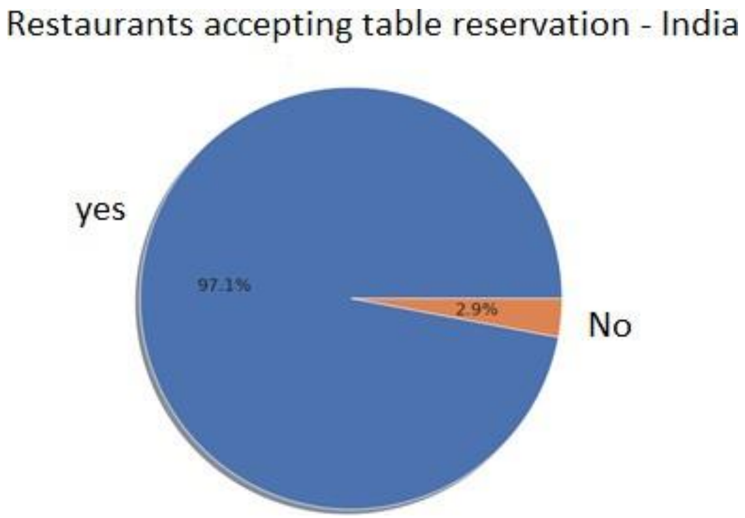


Fig.8

Restaurants accepting Online delivery or not plotted with the help of pie chart.(Fig.9). We can say that 93.7% of restaurants having Online delivery, and rest of the 6.3% doesn't provides delivery facility.

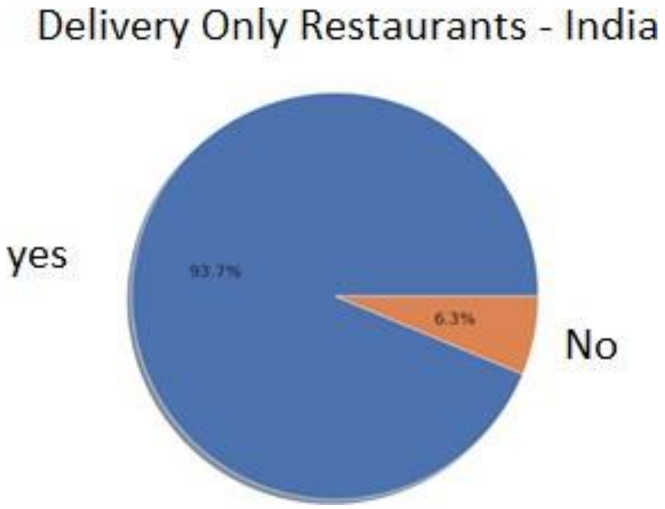


Fig.9

Type of restaurants available after midnight till 5 am plotted with the help of bar chart fig.10 We can say that the maximum number of north Indian restaurants are available midnight till am.

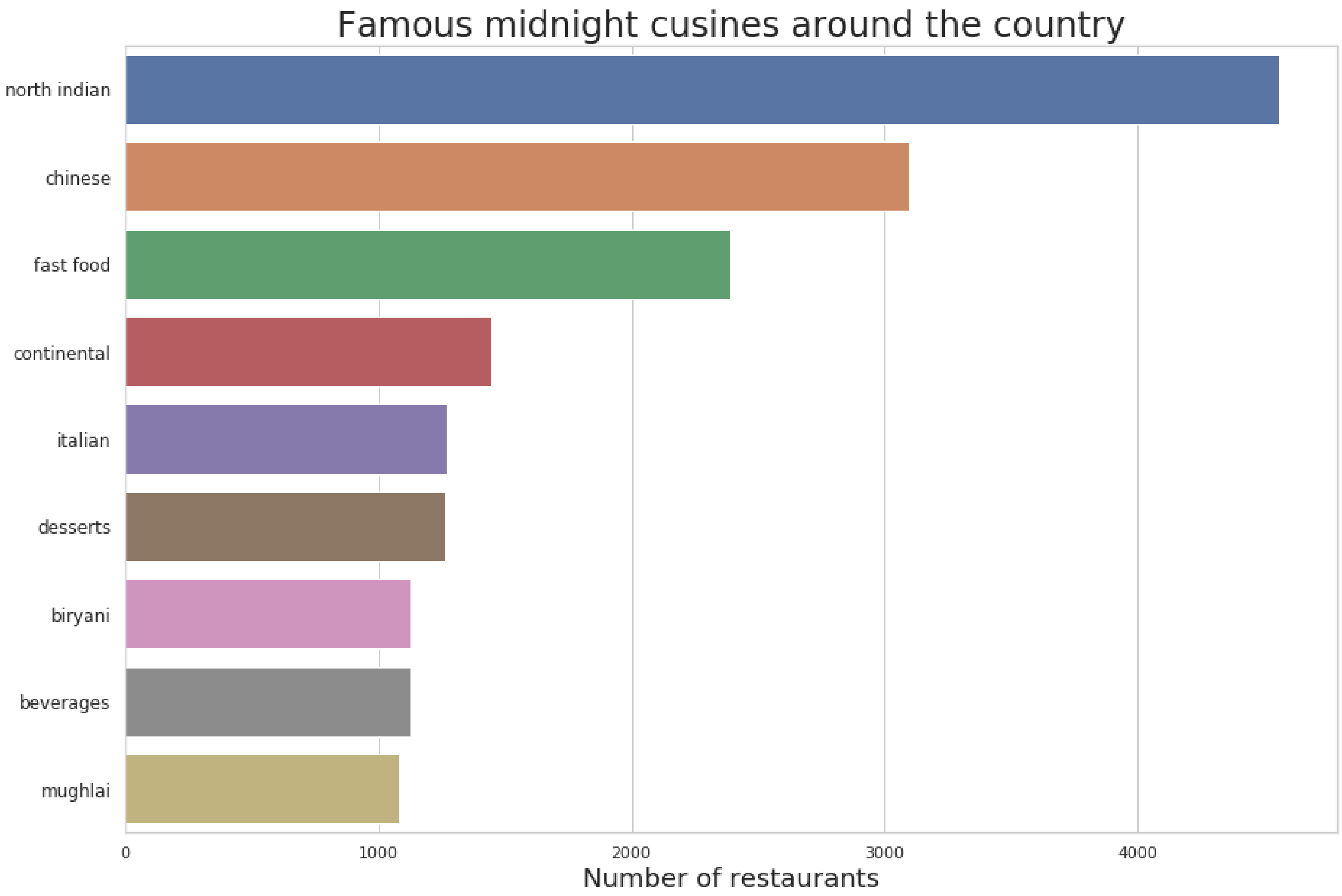


Fig.10

4. Results and discussion:

Multiple Linear regression, random forest, decision tree , KNN and gradient boosting machine algorithm were used to predict the ratings of restaurant. Among the given algorithms Decision Tree algorithm was the best performing one as it provided the best average accuracy 93.54

```
from sklearn.tree import DecisionTreeRegressor
RegModel = DecisionTreeRegressor(max_depth=6,criterion='mse')

print(RegModel)

DT=RegModel.fit(X_train,y_train)
prediction=DT.predict(X_test)

from sklearn import metrics

        R2 = r2_score(y_test, y_prediction)
        print(f"R2 = {R2}")

TestingDataResults=pd.DataFrame(data=X_test, columns=Predictors)
TestingDataResults[TargetVariable]=y_test
TestingDataResults[('Predicted'+TargetVariable)]=np.round(prediction)

# Printing sample prediction values
print(TestingDataResults[[TargetVariable,'Predicted'+TargetVariable]].head())

# Calculating the error for each row
TestingDataResults['APE']=100 * ((abs(
TestingDataResults['rating']-
TestingDataResults['Predictedrating']))/TestingDataResults['rating'])

MAPE=np.mean(TestingDataResults['APE'])
MedianMAPE=np.median(TestingDataResults['APE'])

Accuracy =100 - MAPE
MedianAccuracy=100- MedianMAPE
print('Mean Accuracy on test data:', Accuracy) # Can be negative sometimes due to outlier
print('Median Accuracy on test data:', MedianAccuracy)
```

```
# Defining a custom function to calculate accuracy
# Make sure there are no zeros in the Target variable if you are using MAPE
def Accuracy_Score(orig,pred):
    MAPE = np.mean(100 * (np.abs(orig-pred)/orig))
    #print('#'*70,'Accuracy:', 100-MAPE)
    return(100-MAPE)

# Custom Scoring MAPE calculation
from sklearn.metrics import make_scorer
custom_Scoring=make_scorer(Accuracy_Score, greater_is_better=True)

# Importing cross validation function from sklearn
from sklearn.model_selection import cross_val_score

# Running 10-Fold Cross validation on a given algorithm
# Passing full data X and y because the K-fold will split the data and automatically choose
train/test
Accuracy_Values=cross_val_score(RegModel, X , y, cv=10, scoring=custom_Scoring)
print('\nAccuracy values for 10-fold Cross Validation:\n',Accuracy_Values)
print('\nFinal Average Accuracy of the model:', round(Accuracy_Values.mean(),2))
```

```
Final Average Accuracy of the model: 93.54
```

5 GUI:

GUI is made using Flask framework. **Flask** is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools

6 Future work And Conclusion

6.1 Future Work:

Due to memory constraints, we have only considered 45 cities of the restaurants data. In the expansion of our project, we can add and use more cities of data. The memory and the response time can both be addressed at the same time big data concepts we wish to recreate the project using Hadoop-Mahoot or spark with the mlLib library in the future.

Also In future by using DBMS connectivity and geolocations from advanced sql we can extract restaurants data from selected area and then send it to the system to perform operations.

6.2 Conclusion:

- Cuisines of restaurants is a major contributing factor for the higher rating of restaurants.
- City, cost and ratings are important factors for recommendation system.
- Due to pandemic restaurants more likely focused on to give facilities like online delivery and table reservation.
- Among the trained models for predicting the ratings of restaurants Decision tree Machine learning algorithm performs the best.