A Mini Project Report

On

**Mood Based Restaurant Recommendation System**

By

**A Srinivasa Sameer**

**1602-17-733-107**

**K Vrinda**

**1602-17-733-119**



**Department of Computer Science & Engineering**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**ACKNOWLEDGEMENT**

We take this opportunity with pride and enormous gratitude, to express the deeply embedded feeling and gratefulness to our respectable guide Ms. M. Sunitha Reddy Mam, Department of Computer Science and engineering, whose guidance was unforgettable. Her constructive suggestions have made the presentation of our project, a grand success.

We are thankful to Dr. T. Adilakshmi Mam, Head of Department (CSE), Vasavi College of Engineering for her help during our course work.

Finally, we express our heart full thanks to the CSE department of our college, Vasavi College of Engineering for providing the necessary guidance and support to complete our project work successively.

**CONTENTS**

[ABSTRACT 5](#_Toc59205355)

[1 INTRODUCTION 6](#_Toc59205356)

[1.1 Problem Statement 6](#_Toc59205357)

[2 SOFTWARE REQUIREMENTS SPECIFICATIONS 7](#_Toc59205358)

[3 SYSTEM ARCHITECTURE (DESIGN) 8](#_Toc59205359)

[3.1 Dataset 8](#_Toc59205360)

[4 IMPLEMENTATION & OUTPUTS 10](#_Toc59205362)

[5 RESULTS AND ANALYSIS 21](#_Toc59205377)

[6 CONCLUSION AND FUTURE WORK 24](#_Toc59205378)

[7 REFERENCES 25](#_Toc59205379)

**LIST OF FIGURES**

Figure 1: Famous Cuisines in New Delhi …………………………………………. 21

Figure 2: Location Wise Restaurant Median Rating in New Delhi ………………. 21

Figure 3: Comfort food and Comfort Food Reasons ……………………………… 22

Figure 4(A): Restaurants List ………...……………………………………………. 23

Figure 4(B): Restaurants List ……………….……………………………………… 23

# **ABSTRACT**

Technology has created an exceptional platform for growth of every kind of businesses. The emerging use of technology urges the need of use of IT is all possible aspects of business. Today hotel and restaurant business are one of the most growing business and has been helping a lot in the economy of the country.

The project analyzes the data of rating and location of a restaurant in the chosen dataset and also the mood of the user and use the data to recommend foods and restaurants to the users. The recommendation is based on the mood of the user and the comfort food corresponding to that particular mood. The recommendation is done on the basis of k-means clustering algorithm.

# **INTRODUCTION**

People from different world like to visit different eateries. The restaurant recommendation system is the most needful recommender system as per now in the modern world where all the families try different eateries to have a peaceful amount of time in their relationships. The restaurant recommendation plays a vital role for a family, a single person, a couple to decide which place to go according to their convenience.

## **1.1 PROBLEM STATEMENT**

User has to tell us his/her mood, we'll recommend him/her a restaurant according to the mood. The moods are stress, laziness, happy, depression, sick, cold weather, hunger, etc. The model works well for age range 15-30 due to the dataset.

# **SOFTWARE REQUIREMENTS SPECIFICATIONS**

Any of the following browsers are supported- Chrome 70+, Safari 10+, Firefox 60+, Opera 60+, IE 9+

Google Colab

Modules used:

* **pandas**
* **numpy**
* **seaborn**
* **nltk**
* **wordcloud**
* **matplotlib.pyplot**
* **Collections**

# **SYSTEM ARCHITECTURE (DESIGN)**

## **3.1 DATASET**

**Dataset Used**: Zomato Restaurants data

### Context

I really get fascinated by good quality food being served in the restaurants and would like to help community find the best cuisines around their area

### Content

Zomato API Analysis is one of the most useful analysis for foodies who want to taste the best cuisines of every part of the world which lies in their budget. This analysis is also for those who want to find the value for money restaurants in various parts of the country for the cuisines. Additionally, this analysis caters the needs of people who are striving to get the best cuisine of the country and which locality of that country serves that cuisines with maximum number of restaurants.

Data Collection:  
Data collected can be seen as a raw .json file here

Data Storage:  
The collected data has been stored in the Comma Separated Value file Zomato.csv. Each restaurant in the dataset is uniquely identified by its Restaurant Id. Every Restaurant contains the following variables:

• Restaurant Id: Unique id of every restaurant across various cities of the world  
• Restaurant Name: Name of the restaurant  
• Country Code: Country in which restaurant is located  
• City: City in which restaurant is located  
• Address: Address of the restaurant  
• Locality: Location in the city  
• Locality Verbose: Detailed description of the locality  
• Longitude: Longitude coordinate of the restaurant's location  
• Latitude: Latitude coordinate of the restaurant's location  
• Cuisines: Cuisines offered by the restaurant  
• Average Cost for two: Cost for two people in different currencies   
• Currency: Currency of the country  
• Has Table booking: yes/no  
• Has Online delivery: yes/ no  
• Is delivering: yes/ no  
• Switch to order menu: yes/no  
• Price range: range of price of food  
• Aggregate Rating: Average rating out of 5  
• Rating color: depending upon the average rating color  
• Rating text: text on the basis of rating of rating  
• Votes: Number of ratings casted by

**Dataset Used**: Food Choices

This dataset includes information on food choices, nutrition, preferences, childhood favorites, and other information from college students. There are 126 responses from students. Data is raw and uncleaned. Cleaning is in the process and as soon as that is done, additional versions of the data will be posted.

Inspiration

How important is nutrition information for today's college kids? Is their taste in food defined by their food preferences when they were children? Are kids of parents who cook more likely to make better food choices than others? Are these kids likely to have a different taste compared to others? There a number of open-ended questions included in this dataset such as: What is your favorite comfort food? What is your favorite cuisine? that could work well for natural language processing.

# **IMPLEMENTATION & OUTPUT**

**Recommending Restaurants based on User Moods**

User has to enter his/her mood.

According to the mood entered, restaurants having the comfort food in the chosen city are displayed. The moods are stress, laziness, happy, depression, sick, cold weather, hunger, etc. The model works well for age range 15-30 due to the dataset.

**### About the Dataset**

We are using two datasets. First is Zomato Restaurants Dataset and Second is Food Choices of College Students Dataset.

import nltk

nltk.download('stopwords')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from wordcloud import WordCloud, STOPWORDS

stopwords = set(STOPWORDS)

import seaborn as sns

from plotly.offline import init\_notebook\_mode, iplot

init\_notebook\_mode()

from collections import Counter

from nltk.corpus import stopwords

stop = set(stopwords.words('english'))

stop.update(['.', ',', '"', "'", '?', '!', ':', ';', '(', ')', '[', ']', '{', '}',''])

from nltk.stem import WordNetLemmatizer

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

**### Zomato Restaurants Dataset Analysis (in New Delhi)**

res\_data = pd.read\_csv('/content/zomato.csv', encoding='latin-1')

countryCode\_toName = {

    1: "India",

    14: "Australia",

    30: "Brazil",

    37: "Canada",

    94: "Indonesia",

    148: "New Zealand",

    162: "Phillipines",

    166: "Qatar",

    184: "Singapore",

    189: "South Africa",

    191: "Sri Lanka",

    208: "Turkey",

    214: "UAE",

    215: "United Kingdom",

    216: "United States",

}

res\_data['Country'] = res\_data['Country Code'].apply(lambda x: countryCode\_toName[x])

res\_data = res\_data.loc[(res\_data['Country Code'] == 1) & (res\_data['City'] == 'New Delhi'), :]

res\_data = res\_data.loc[res\_data['Longitude'] != 0, :]

res\_data = res\_data.loc[res\_data['Latitude'] != 0, :]

res\_data = res\_data.loc[res\_data['Latitude'] < 29] # clearing out invalid outlier

res\_data = res\_data.loc[res\_data['Rating text'] != 'Not rated']

res\_data['Cuisines'] = res\_data['Cuisines'].astype(str)

res\_data['fusion\_num'] = res\_data['Cuisines'].apply(lambda x: len(x.split(',')))

res\_data.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 3975 entries, 2561 to 8009

Data columns (total 23 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Restaurant ID 3975 non-null int64

1 Restaurant Name 3975 non-null object

2 Country Code 3975 non-null int64

3 City 3975 non-null object

4 Address 3975 non-null object

5 Locality 3975 non-null object

6 Locality Verbose 3975 non-null object

7 Longitude 3975 non-null float64

8 Latitude 3975 non-null float64

9 Cuisines 3975 non-null object

10 Average Cost for two 3975 non-null int64

11 Currency 3975 non-null object

12 Has Table booking 3975 non-null object

13 Has Online delivery 3975 non-null object

14 Is delivering now 3975 non-null object

15 Switch to order menu 3975 non-null object

16 Price range 3975 non-null int64

17 Aggregate rating 3975 non-null float64

18 Rating color 3975 non-null object

19 Rating text 3975 non-null object

20 Votes 3975 non-null int64

21 Country 3975 non-null object

22 fusion\_num 3975 non-null int64

dtypes: float64(3), int64(6), object(14)

memory usage: 745.3+ KB

**### What are the most famous cuisines in CHOOSEN CITY?**

lst\_cuisine = set()

Cnt\_cuisine = Counter()

for cu\_lst in res\_data['Cuisines']:

    cu\_lst = cu\_lst.split(',')

    lst\_cuisine.update([cu.strip() for cu in cu\_lst])

    for cu in cu\_lst:

        Cnt\_cuisine[cu.strip()] += 1

cnt = pd.DataFrame.from\_dict(Cnt\_cuisine, orient = 'index')

cnt.sort\_values(0, ascending = False, inplace = True)

tmp\_cnt = cnt.head(10)

tmp\_cnt.rename(columns = {0:'cnt'}, inplace = True)

with plt.style.context('bmh'):

    f = plt.figure(figsize = (12, 8))

    ax = plt.subplot2grid((2,2), (0,0))

    sns.barplot(x = tmp\_cnt.index, y = 'cnt', data = tmp\_cnt, ax = ax, palette = sns.color\_palette('Blues\_d', 10))

    ax.set\_title('# Cuisine')

    ax.tick\_params(axis='x', rotation=70)

    ax = plt.subplot2grid((2,2), (0,1))

    sns.countplot(res\_data['fusion\_num'], ax=ax, palette = sns.color\_palette('Blues\_d', res\_data.fusion\_num.nunique()))

    ax.set\_title('# Cuisine Provided')

    ax.set\_ylabel('')

    plt.show()

print('# Unique Cuisine: ', len(lst\_cuisine))

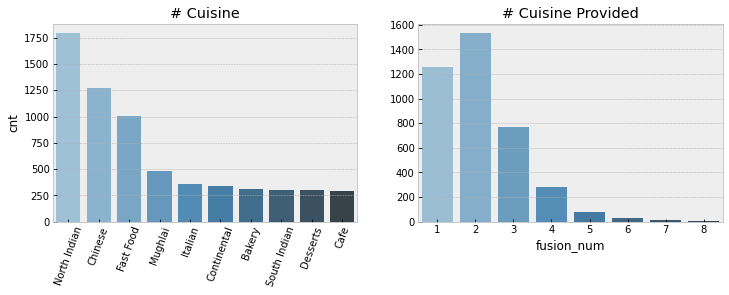
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4308: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy>

/usr/local/lib/python3.6/dist-packages/seaborn/\_decorators.py:43: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.



# Unique Cuisine: 78

**### K-Means Clustering - Where are high rated restaurants located?**

res\_data['Rating category'] = res\_data['Rating text'].map({'Not rated': -1, 'Poor':0, 'Average':2, 'Good':3, 'Very Good':4, 'Excellent':5})

tmp = res\_data['Aggregate rating'].map(np.round)

a = np.full(tmp.shape[0], False, dtype = bool)

((tmp - res\_data['Rating category']).map(np.round)).value\_counts()

sys\_check = res\_data[['Aggregate rating', 'Rating category', 'Votes']].copy()

sys\_check['distorted'] = (res\_data['Aggregate rating'] - res\_data['Rating category']).map(np.round)

sys\_check['diff'] = sys\_check['Aggregate rating'] - sys\_check['Rating category']

res\_data = res\_data.loc[sys\_check['distorted'] != 2, :]

res\_data['Rating category'] = res\_data['Aggregate rating'].round(0).astype(int)

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=7, random\_state=0).fit(res\_data[['Longitude', 'Latitude']])

res\_data['pos'] = kmeans.labels\_

pop\_local = res\_data.groupby('pos')['Longitude', 'Latitude', 'Aggregate rating'].agg({'Longitude':np.mean, 'Latitude':np.mean, 'Aggregate rating':np.median}).reset\_index()

with plt.style.context('bmh', after\_reset=True):

    pal = sns.color\_palette('Spectral', 7)

    plt.figure(figsize = (8,6))

    for i in range(7):

        ix = res\_data.pos == i

        plt.scatter(res\_data.loc[ix, 'Latitude'], res\_data.loc[ix, 'Longitude'], color = pal[i], label = str(i))

        plt.text(pop\_local.loc[i, 'Latitude'], pop\_local.loc[i, 'Longitude'], str(i) + ': '+str(pop\_local.loc[i, 'Aggregate rating'].round(2)), fontsize = 14, color = 'black')

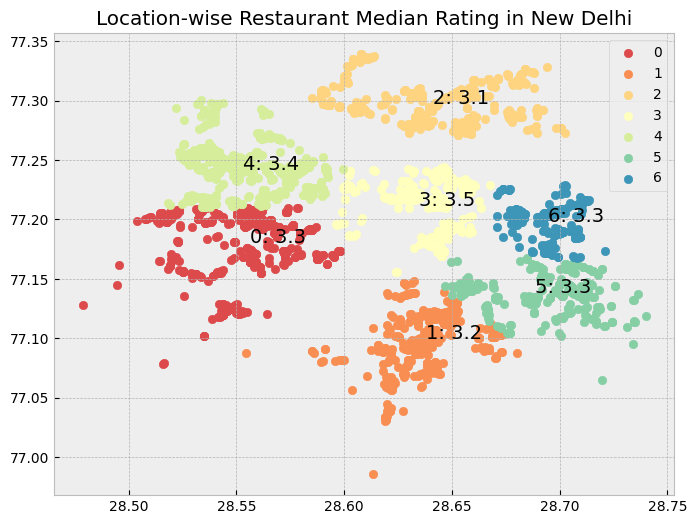
    plt.title('Location-wise Restaurant Median Rating in New Delhi')

    plt.legend()

    plt.show()

/usr/local/lib/python3.6/dist-packages/ipykernel\_launcher.py:4: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.



**### Food Choices Dataset Analysis**

food\_data = pd.read\_csv('/content/food\_choices.csv')

food\_data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 125 entries, 0 to 124

Data columns (total 61 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 GPA 123 non-null object

1 Gender 125 non-null int64

2 breakfast 125 non-null int64

3 calories\_chicken 125 non-null int64

4 calories\_day 106 non-null float64

5 calories\_scone 124 non-null float64

6 coffee 125 non-null int64

7 comfort\_food 124 non-null object

8 comfort\_food\_reasons 124 non-null object

9 comfort\_food\_reasons\_coded 106 non-null float64

10 cook 122 non-null float64

11 comfort\_food\_reasons\_coded.1 125 non-null int64

12 cuisine 108 non-null float64

13 diet\_current 124 non-null object

14 diet\_current\_coded 125 non-null int64

15 drink 123 non-null float64

16 eating\_changes 122 non-null object

17 eating\_changes\_coded 125 non-null int64

18 eating\_changes\_coded1 125 non-null int64

19 eating\_out 125 non-null int64

20 employment 116 non-null float64

21 ethnic\_food 125 non-null int64

22 exercise 112 non-null float64

23 father\_education 124 non-null float64

24 father\_profession 122 non-null object

25 fav\_cuisine 123 non-null object

26 fav\_cuisine\_coded 125 non-null int64

27 fav\_food 123 non-null float64

28 food\_childhood 124 non-null object

29 fries 125 non-null int64

30 fruit\_day 125 non-null int64

31 grade\_level 125 non-null int64

32 greek\_food 125 non-null int64

33 healthy\_feeling 125 non-null int64

34 healthy\_meal 124 non-null object

35 ideal\_diet 124 non-null object

36 ideal\_diet\_coded 125 non-null int64

37 income 124 non-null float64

38 indian\_food 125 non-null int64

39 italian\_food 125 non-null int64

40 life\_rewarding 124 non-null float64

41 marital\_status 124 non-null float64

42 meals\_dinner\_friend 122 non-null object

43 mother\_education 122 non-null float64

44 mother\_profession 123 non-null object

45 nutritional\_check 125 non-null int64

46 on\_off\_campus 124 non-null float64

47 parents\_cook 125 non-null int64

48 pay\_meal\_out 125 non-null int64

49 persian\_food 124 non-null float64

50 self\_perception\_weight 124 non-null float64

51 soup 124 non-null float64

52 sports 123 non-null float64

53 thai\_food 125 non-null int64

54 tortilla\_calories 124 non-null float64

55 turkey\_calories 125 non-null int64

56 type\_sports 104 non-null object

57 veggies\_day 125 non-null int64

58 vitamins 125 non-null int64

59 waffle\_calories 125 non-null int64

60 weight 123 non-null object

dtypes: float64(20), int64(27), object(14)

memory usage: 59.7+ KB

**### What are some comfort foods in various situations such as stress, boredom, hunger, happiness?**

food\_data[['comfort\_food\_reasons', 'comfort\_food']]

|  | **comfort\_food\_reasons** | **comfort\_food** |
| --- | --- | --- |
| **0** | we dont have comfort | none |
| **1** | Stress, bored, anger | chocolate, chips, ice cream |
| **2** | stress, sadness | frozen yogurt, pizza, fast food |
| **3** | Boredom | Pizza, Mac and cheese, ice cream |
| **4** | Stress, boredom, cravings | Ice cream, chocolate, chips |
| **...** | ... | ... |
| **120** | boredom and sadness | wine. mac and cheese, pizza, ice cream |
| **121** | Loneliness / Homesick / Sadness | Pizza / Wings / Cheesecake |
| **122** | sadness | rice, potato, seaweed soup |
| **123** | happiness, they are some of my favorite foods | Mac n Cheese, Lasagna, Pizza |
| **124** | hormones, Premenstrual syndrome. | Chocolates, pizza, and Ritz. |

125 rows × 2 columns

def search\_comfort(mood):

    lemmatizer = WordNetLemmatizer()

    foodcount = {}

    for i in range(124):

        temp = [temps.strip().replace('.','').replace(',','').lower() for temps in str(food\_data["comfort\_food\_reasons"][i]).split(' ') if temps.strip() not in stop ]

        if mood in temp:

            foodtemp = [lemmatizer.lemmatize(temps.strip().replace('.','').replace(',','').lower()) for temps in str(food\_data["comfort\_food"][i]).split(',') if temps.strip() not in stop ]

            for a in foodtemp:

                if a not in foodcount.keys():

                    foodcount[a] = 1

                else:

                    foodcount[a] += 1

    sorted\_food = []

    sorted\_food = sorted(foodcount, key=foodcount.get, reverse=True)

    return sorted\_food

def find\_my\_comfort\_food(mood):

    topn = []

    topn = search\_comfort(mood) #function create dictionary only for particular mood

    print("5 Popular Comfort Foods in %s are:"%(mood))

    for i in range(5):

      print(topn[i])

import nltk

nltk.download('wordnet')

find\_my\_comfort\_food('happy')

[nltk\_data] Downloading package wordnet to /root/nltk\_data...

[nltk\_data] Package wordnet is already up-to-date!

3 Popular Comfort Foods in happy are:

pizza

ice cream

chicken wings

pretzel

fruit snacks

**###Suggesting Restaurants based on User Moods**

**##### Under Following Moods**

\* stress

\* boredom

\* depression/sadness

\* hunger

\* laziness

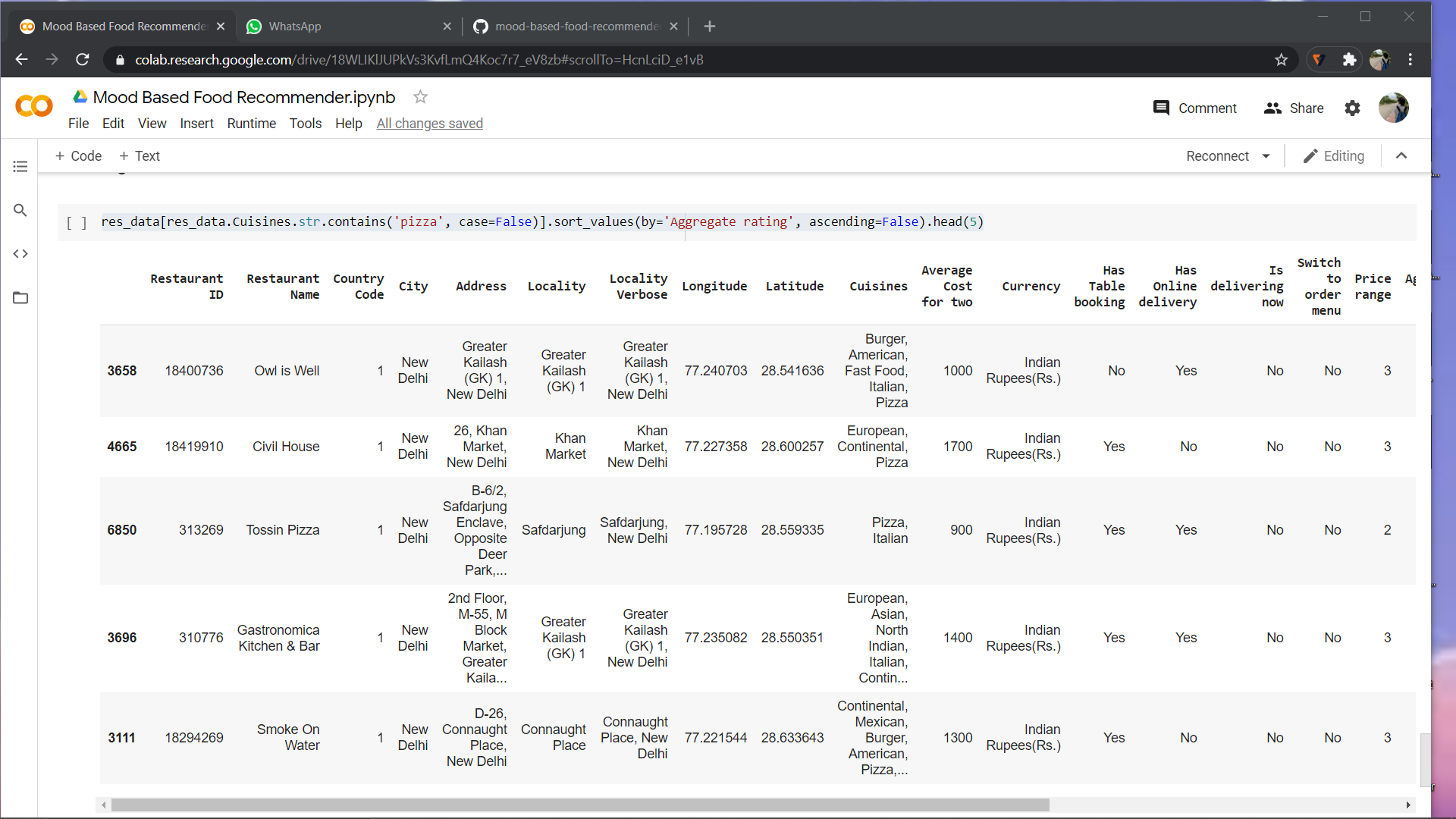
\* cold weather

\* happiness

\* watching tv

**### Finding Restaurants based on cuisines**

res\_data[res\_data.Cuisines.str.contains('pizza', case=False)].sort\_values(by='Aggregate rating', ascending=False).head(5)



# **RESULTS AND ANALYSIS**

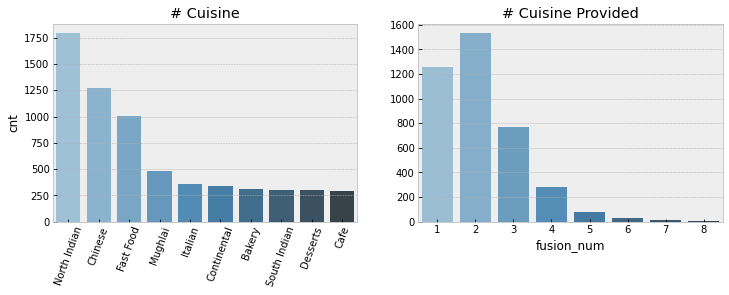


Figure 1: Famous Cuisines in New Delhi

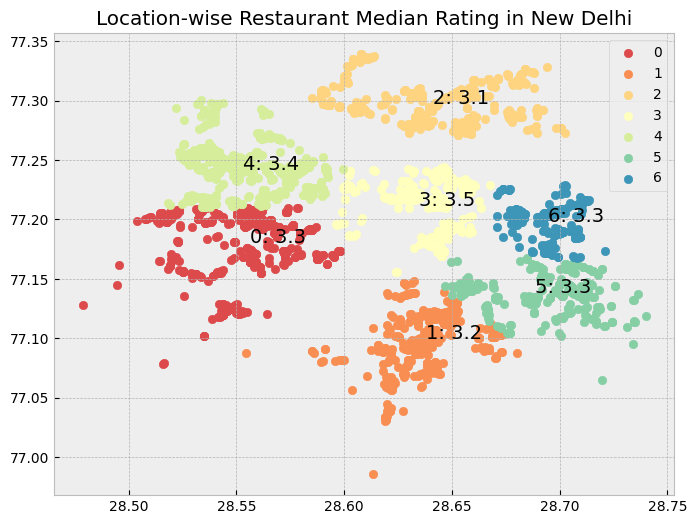


Figure 2: Location wise Restaurant Median Rating in New Delhi



Figure 3: Comfort food and Comfort Food Reasons

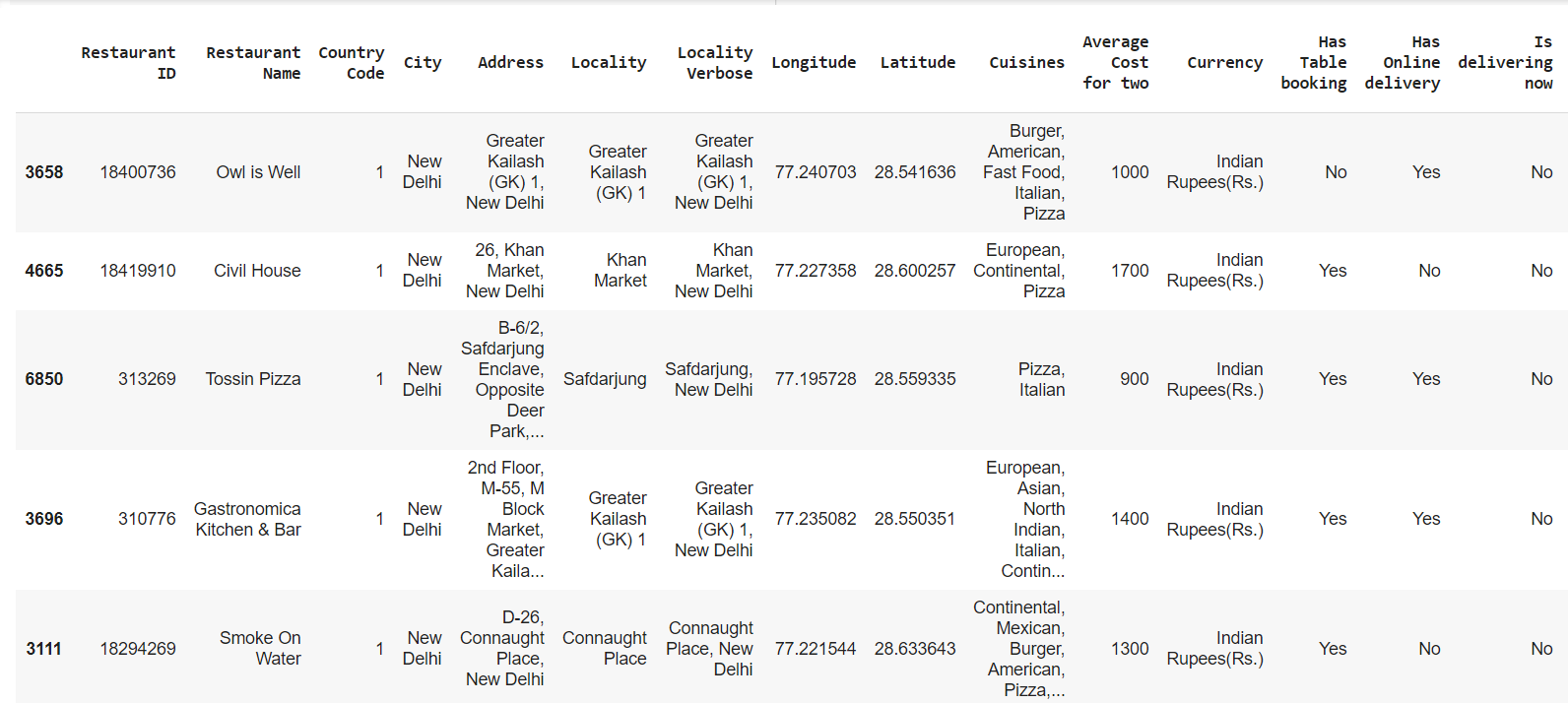


Figure 4(A): Restaurants List

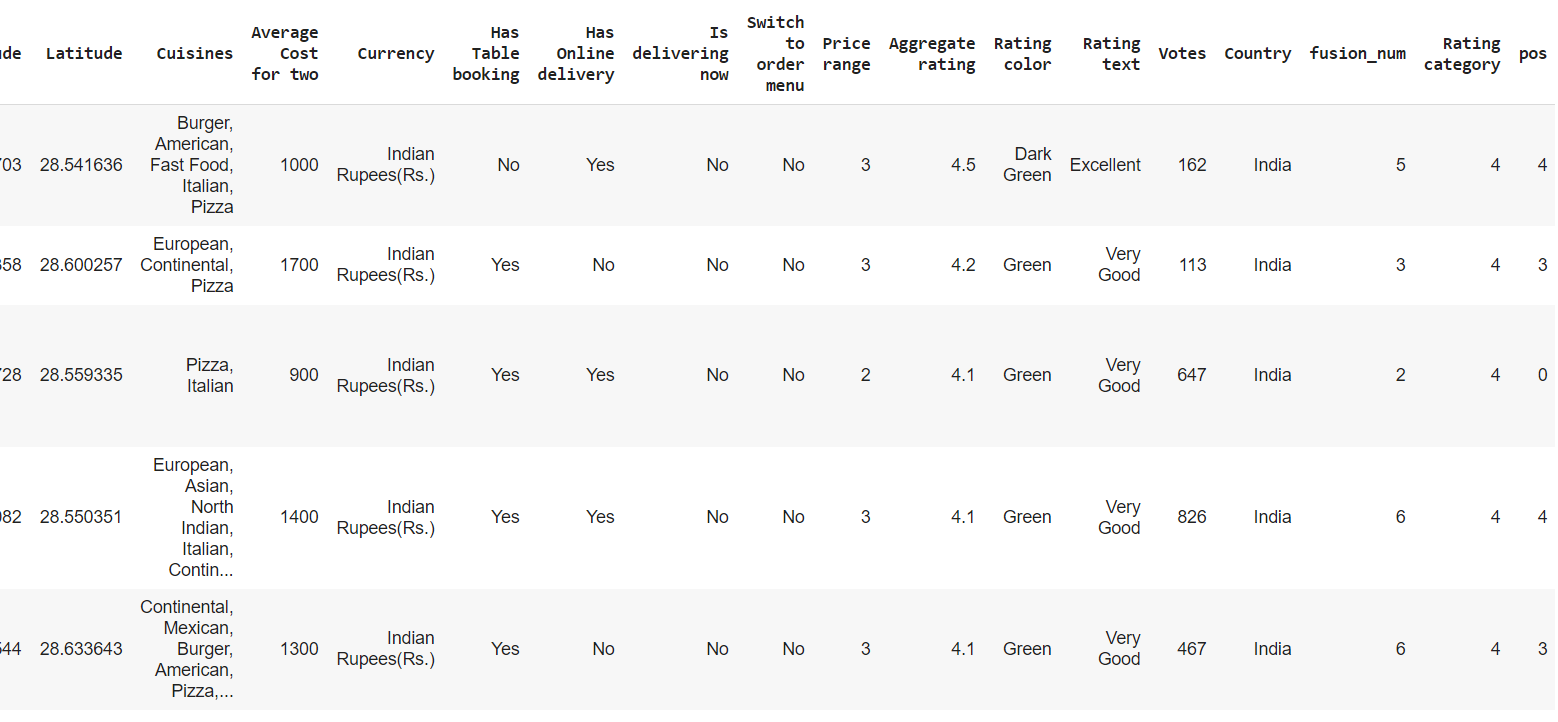


Figure 4(B): Restaurants List

# **CONCLUSION AND FUTURE WORK**

This project can be made into a fully functional website.

We can implement the following functionalities:

* Automatically detecting user location
* GUI input for user mood
* City/location input from user
* Enabling restaurant owners to add their restaurant to the list.

# **REFERENCES**

[1] <https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm>

[2] <https://matplotlib.org/>

[3] <https://www.youtube.com/watch?v=i-HnvsehuSw>

[4] <https://www.kaggle.com/shrutimehta/zomato-restaurants-data>

[5] <https://www.kaggle.com/borapajo/food-choices>