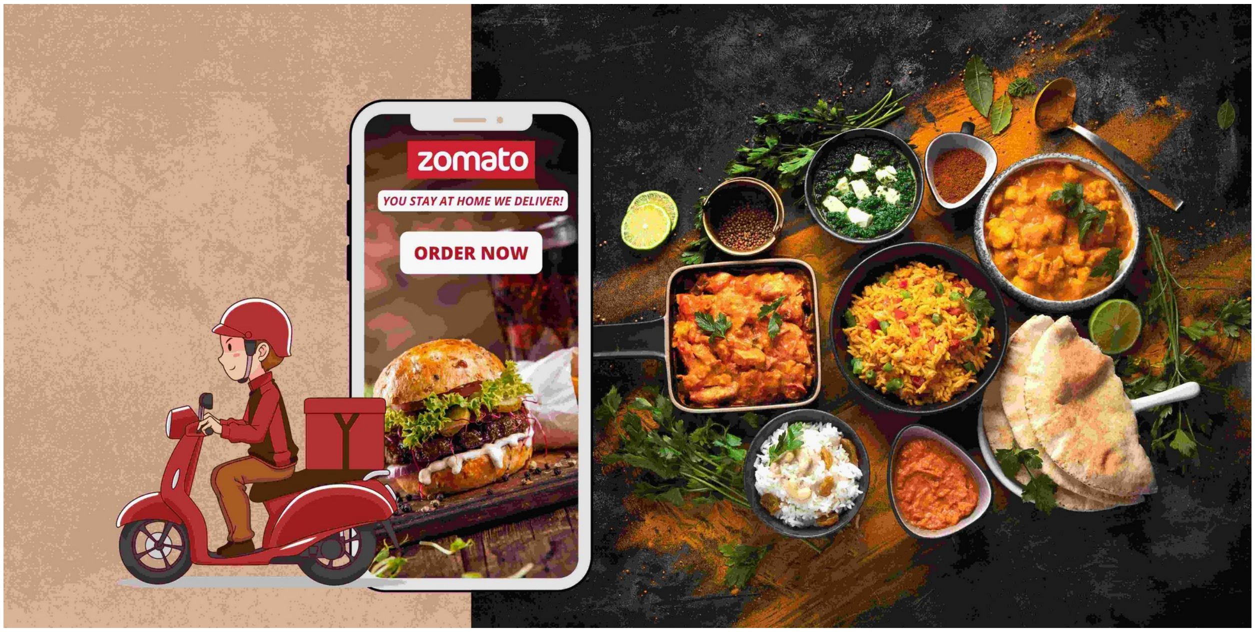


Submitted by: Rohan Negi

Submitted to : Neeraj ma'am

Import Necessary Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Knowing basic composition of data

```
df = pd.read_csv("C:/Users/DELL/Downloads/Indian-Resturants.csv")
df
```

```
         res_id                             name      establishment  \
0       3400299                       Bikanervala     ['Quick Bites']
1       3400005   Mama Chicken Mama Franky House     ['Quick Bites']
2       3401013                     Bhagat Halwai     ['Quick Bites']
3       3400290                     Bhagat Halwai     ['Quick Bites']
4       3401744      The Salt Cafe Kitchen & Bar   ['Casual Dining']
...         ...                              ...               ...
211939  3202251   Kali Mirch Cafe And Restaurant   ['Casual Dining']
211940  3200996                       Raju Omlet     ['Quick Bites']
```

```
211941  18984164                The Grand Thakar  ['Casual Dining']
211942   3201138                          Subway    ['Quick Bites']
211943  18879846     Freshco's - The Health Cafe           ['Café']

                                                      url  \
0       https://www.zomato.com/agra/bikanervala-khanda...
1       https://www.zomato.com/agra/mama-chicken-mama-...
2       https://www.zomato.com/agra/bhagat-halwai-2-sh...
3       https://www.zomato.com/agra/bhagat-halwai-civi...
4       https://www.zomato.com/agra/the-salt-cafe-kitc...
...                                                   ...
211939  https://www.zomato.com/vadodara/kali-mirch-caf...
211940  https://www.zomato.com/vadodara/raju-omlet-kar...
211941  https://www.zomato.com/vadodara/the-grand-thak...
211942  https://www.zomato.com/vadodara/subway-1-akota...
211943  https://www.zomato.com/vadodara/freshcos-the-h...

                                                  address      city
city_id  \
0       Kalyani Point, Near Tulsi Cinema, Bypass Road,...      Agra
34
1             Main Market, Sadar Bazaar, Agra Cantt, Agra      Agra
34
2       62/1, Near Easy Day, West Shivaji Nagar, Goalp...      Agra
34
3       Near Anjana Cinema, Nehru Nagar, Civil Lines, ...      Agra
34
4             1C,3rd Floor, Fatehabad Road, Tajganj, Agra      Agra
34
...                                                   ...       ...
...
211939  Manu Smriti Complex, Near Navrachna School, GI...  Vadodara
32
211940  Mahalaxmi Apartment, Opposite B O B, Karoli Ba...  Vadodara
32
211941  3rd Floor, Shreem Shalini Mall, Opposite Conqu...  Vadodara
32
211942  G-2, Vedant Platina, Near Cosmos, Akota, Vadodara  Vadodara
32
211943  Shop 7, Ground Floor, Opposite Natubhai Circle...  Vadodara
32

           locality   latitude  longitude  ... price_range
currency  \
0          Khandari  27.211450  78.002381  ...           2       Rs.

1        Agra Cantt  27.160569  78.011583  ...           2       Rs.

2          Shahganj  27.182938  77.979684  ...           1       Rs.
```

```
3       Civil Lines  27.205668  78.004799  ...           1       Rs.
4           Tajganj  27.157709  78.052421  ...           3       Rs.
...             ...        ...        ...  ...         ...       ...
211939    Fatehgunj  22.336931  73.192356  ...           2       Rs.
211940   Karelibaug  22.322455  73.197203  ...           1       Rs.
211941     Alkapuri  22.310563  73.171163  ...           2       Rs.
211942        Akota  22.270027  73.143068  ...           2       Rs.
211943     Vadiwadi  22.309935  73.158768  ...           2       Rs.

                                               highlights
aggregate_rating  \
0       ['Lunch', 'Takeaway Available', 'Credit Card',...
4.4
1       ['Delivery', 'No Alcohol Available', 'Dinner',...
4.4
2       ['No Alcohol Available', 'Dinner', 'Takeaway A...
4.2
3       ['Takeaway Available', 'Credit Card', 'Lunch',...
4.3
4       ['Lunch', 'Serves Alcohol', 'Cash', 'Credit Ca...
4.9
...                                                   ...
...
211939  ['Dinner', 'Cash', 'Lunch', 'Delivery', 'Indoo...
4.1
211940  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.1
211941  ['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...
4.0
211942  ['Dinner', 'Delivery', 'Credit Card', 'Lunch',...
3.7
211943  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.0

       rating_text  votes  photo_count opentable_support delivery
takeaway
0        Very Good    814          154               0.0       -1
-1
1        Very Good   1203          161               0.0       -1
-1
2        Very Good    801          107               0.0        1
-1
```

```
3        Very Good    693          157                0.0        1
-1
4        Excellent    470          291                0.0        1
-1
...            ...    ...          ...                ...      ...
...
211939   Very Good    243           40                0.0       -1
-1
211940   Very Good    187           40                0.0        1
-1
211941   Very Good    111           38                0.0       -1
-1
211942        Good    128           34                0.0        1
-1
211943   Very Good     93           53                0.0        1
-1

[211944 rows x 26 columns]
```

```
df.head(10)
```

```
    res_id                             name      establishment  \
0  3400299                      Bikanervala    ['Quick Bites']
1  3400005  Mama Chicken Mama Franky House    ['Quick Bites']
2  3401013                    Bhagat Halwai    ['Quick Bites']
3  3400290                    Bhagat Halwai    ['Quick Bites']
4  3401744     The Salt Cafe Kitchen & Bar  ['Casual Dining']
5  3400275                   Domino's Pizza    ['Quick Bites']
6  3400296              Honeydew Restaurant    ['Quick Bites']
7  3400368                   Domino's Pizza    ['Quick Bites']
8  3401284                       Cake House         ['Bakery']
9  3400838                    Sugar N Thyme           ['Café']

                                                 url  \
0  https://www.zomato.com/agra/bikanervala-khanda...
1  https://www.zomato.com/agra/mama-chicken-mama-...
2  https://www.zomato.com/agra/bhagat-halwai-2-sh...
3  https://www.zomato.com/agra/bhagat-halwai-civi...
4  https://www.zomato.com/agra/the-salt-cafe-kitc...
5  https://www.zomato.com/agra/dominos-pizza-civi...
6  https://www.zomato.com/agra/honeydew-restauran...
7  https://www.zomato.com/agra/dominos-pizza-sika...
8  https://www.zomato.com/agra/cake-house-2-civil...
9  https://www.zomato.com/agra/sugar-n-thyme-tajg...

                                             address  city  city_id  \
0  Kalyani Point, Near Tulsi Cinema, Bypass Road,...  Agra       34
1        Main Market, Sadar Bazaar, Agra Cantt, Agra  Agra       34
2  62/1, Near Easy Day, West Shivaji Nagar, Goalp...  Agra       34
3  Near Anjana Cinema, Nehru Nagar, Civil Lines, ...  Agra       34
```

```
4        1C,3rd Floor, Fatehabad Road, Tajganj, Agra  Agra       34
5  114/23 G, Deep Shikha Complex, Sanjay Place, C...  Agra       34
6       Opposite Soami Bagh Temple, Dayal Bagh, Agra  Agra       34
7              Plot C-1/6, Sector 13, Sikandra, Agra  Agra       34
8  23/301, Wazirpura Rd, Judge Compound Chowraha,...  Agra       34
9  1374 K/1375 K, Ground floor, Dinesh Nagar, Fat...  Agra       34

      locality   latitude  longitude  ... price_range  currency  \
0     Khandari  27.211450  78.002381  ...           2       Rs.
1   Agra Cantt  27.160569  78.011583  ...           2       Rs.
2     Shahganj  27.182938  77.979684  ...           1       Rs.
3  Civil Lines  27.205668  78.004799  ...           1       Rs.
4      Tajganj  27.157709  78.052421  ...           3       Rs.
5  Civil Lines  27.201516  78.007556  ...           2       Rs.
6   Dayal Bagh  27.222175  78.010174  ...           2       Rs.
7     Sikandra  27.203930  77.954260  ...           2       Rs.
8  Civil Lines  27.204148  78.009025  ...           2       Rs.
9      Tajganj  27.158243  78.045591  ...           3       Rs.

                                          highlights aggregate_rating
\
0  ['Lunch', 'Takeaway Available', 'Credit Card',...              4.4

1  ['Delivery', 'No Alcohol Available', 'Dinner',...              4.4

2  ['No Alcohol Available', 'Dinner', 'Takeaway A...              4.2

3  ['Takeaway Available', 'Credit Card', 'Lunch',...              4.3

4  ['Lunch', 'Serves Alcohol', 'Cash', 'Credit Ca...              4.9

5  ['Credit Card', 'Lunch', 'Delivery', 'Dinner',...              4.0

6  ['Dinner', 'Delivery', 'Lunch', 'Cash', 'Takea...              4.2

7  ['Lunch', 'Delivery', 'Credit Card', 'No Alcoh...              3.8

8  ['Takeaway Available', 'Cash', 'Indoor Seating...              3.4

9  ['No Alcohol Available', 'Dinner', 'Delivery',...              4.4


  rating_text  votes  photo_count opentable_support delivery  takeaway

0   Very Good    814          154               0.0       -1        -1

1   Very Good   1203          161               0.0       -1        -1

2   Very Good    801          107               0.0        1        -1

3   Very Good    693          157               0.0        1        -1
```

```
4   Excellent    470          291                   0.0         1         -1
5   Very Good    707           62                   0.0        -1         -1
6   Very Good    647           46                   0.0         1         -1
7        Good    617           18                   0.0        -1         -1
8     Average    322           14                   0.0         1         -1
9   Very Good    289          324                   0.0         1         -1

[10 rows x 26 columns]
```

```
df.tail(10)
```

```
          res_id                              name      establishment  \
211934   3200763                              Swad    ['Quick Bites']
211935   3201351                       Mummys Pizza  ['Casual Dining']
211936   3202169                     Red Dot Nation  ['Casual Dining']
211937  18855810                Biryani aur Baatein  ['Casual Dining']
211938  18662583                        Wok On Fire  ['Casual Dining']
211939   3202251  Kali Mirch Cafe And Restaurant  ['Casual Dining']
211940   3200996                         Raju Omlet    ['Quick Bites']
211941  18984164                   The Grand Thakar  ['Casual Dining']
211942   3201138                             Subway    ['Quick Bites']
211943  18879846     Freshco's - The Health Cafe           ['Café']

                                                      url  \
211934  https://www.zomato.com/vadodara/swad-karelibau...
211935  https://www.zomato.com/vadodara/mummys-pizza-d...
211936  https://www.zomato.com/vadodara/red-dot-nation...
211937  https://www.zomato.com/vadodara/biryani-aur-ba...
211938  https://www.zomato.com/vadodara/wok-on-fire-fa...
211939  https://www.zomato.com/vadodara/kali-mirch-caf...
211940  https://www.zomato.com/vadodara/raju-omlet-kar...
211941  https://www.zomato.com/vadodara/the-grand-thak...
211942  https://www.zomato.com/vadodara/subway-1-akota...
211943  https://www.zomato.com/vadodara/freshcos-the-h...

                                                  address      city
city_id  \
211934  G-3, Status Complex, Opposite Amrapali Complex...  Vadodara
32
211935  Top Floor 323 - 327, Southwest Central Mall, D...  Vadodara
32
211936  Vinyak Heights, Beside Bharat Petrol Pump, Wag...  Vadodara
32
211937  Shop 14, Atlantis K-10, A Wing, Genda Circle R...  Vadodara
```

```
32
211938  Ground Floor 1, Rossette Building, Opposite Se...  Vadodara
32
211939  Manu Smriti Complex, Near Navrachna School, GI...  Vadodara
32
211940  Mahalaxmi Apartment, Opposite B O B, Karoli Ba...  Vadodara
32
211941  3rd Floor, Shreem Shalini Mall, Opposite Conqu...  Vadodara
32
211942  G-2, Vedant Platina, Near Cosmos, Akota, Vadodara  Vadodara
32
211943  Shop 7, Ground Floor, Opposite Natubhai Circle...  Vadodara
32

          locality   latitude  longitude  ... price_range  currency  \
211934  Karelibaug  22.320823  73.199167  ...           1       Rs.
211935  Diwalipura  22.280378  73.149108  ...           2       Rs.
211936  Suryanagar  22.281816  73.232252  ...           2       Rs.
211937    Alkapuri  22.317746  73.168043  ...           2       Rs.
211938   Fatehgunj  22.323357  73.187461  ...           3       Rs.
211939   Fatehgunj  22.336931  73.192356  ...           2       Rs.
211940  Karelibaug  22.322455  73.197203  ...           1       Rs.
211941    Alkapuri  22.310563  73.171163  ...           2       Rs.
211942       Akota  22.270027  73.143068  ...           2       Rs.
211943    Vadiwadi  22.309935  73.158768  ...           2       Rs.

                                                highlights
aggregate_rating  \
211934  ['Dinner', 'Takeaway Available', 'Delivery', '...
4.0
211935  ['Dinner', 'Cash', 'Takeaway Available', 'Lunc...
4.3
211936  ['Cash', 'Delivery', 'Credit Card', 'Dinner', ...
3.6
211937  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.1
211938  ['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...
4.0
211939  ['Dinner', 'Cash', 'Lunch', 'Delivery', 'Indoo...
4.1
211940  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.1
211941  ['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...
4.0
211942  ['Dinner', 'Delivery', 'Credit Card', 'Lunch',...
3.7
211943  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.0

       rating_text  votes  photo_count opentable_support delivery
```

```
takeaway
211934   Very Good    365             9                 0.0        -1
-1
211935   Very Good    344            86                 0.0         1
-1
211936        Good    381            19                 0.0        -1
-1
211937   Very Good    154            96                 0.0        -1
-1
211938   Very Good    301           126                 0.0         1
-1
211939   Very Good    243            40                 0.0        -1
-1
211940   Very Good    187            40                 0.0         1
-1
211941   Very Good    111            38                 0.0        -1
-1
211942        Good    128            34                 0.0         1
-1
211943   Very Good     93            53                 0.0         1
-1

[10 rows x 26 columns]
```

```
df.shape
```

```
(211944, 26)
```

```
df.size
```

```
5510544
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   res_id                  211944 non-null  int64
 1   name                    211944 non-null  object
 2   establishment           211944 non-null  object
 3   url                     211944 non-null  object
 4   address                 211810 non-null  object
 5   city                    211944 non-null  object
 6   city_id                 211944 non-null  int64
 7   locality                211944 non-null  object
 8   latitude                211944 non-null  float64
 9   longitude               211944 non-null  float64
 10  zipcode                 48757 non-null   object
 11  country_id              211944 non-null  int64
```

```
 12  locality_verbose      211944 non-null  object 
 13  cuisines              210553 non-null  object 
 14  timings               208070 non-null  object 
 15  average_cost_for_two  211944 non-null  int64  
 16  price_range           211944 non-null  int64  
 17  currency              211944 non-null  object 
 18  highlights            211944 non-null  object 
 19  aggregate_rating      211944 non-null  float64
 20  rating_text           211944 non-null  object 
 21  votes                 211944 non-null  int64  
 22  photo_count           211944 non-null  int64  
 23  opentable_support     211896 non-null  float64
 24  delivery              211944 non-null  int64  
 25  takeaway              211944 non-null  int64  
dtypes: float64(4), int64(9), object(13)
memory usage: 42.0+ MB

df.describe().T

                         count          mean           std   min  \
res_id                211944.0  1.349411e+07  7.883722e+06  50.0   
city_id               211944.0  4.746785e+03  5.568766e+03   1.0   
latitude              211944.0  2.149976e+01  2.278133e+01   0.0   
longitude             211944.0  7.761528e+01  7.500104e+00   0.0   
country_id            211944.0  1.000000e+00  0.000000e+00   1.0   
average_cost_for_two  211944.0  5.958122e+02  6.062394e+02   0.0   
price_range           211944.0  1.882535e+00  8.929891e-01   1.0   
aggregate_rating      211944.0  3.395937e+00  1.283642e+00   0.0   
votes                 211944.0  3.780019e+02  9.253334e+02 -18.0   
photo_count           211944.0  2.569712e+02  8.676689e+02   0.0   
opentable_support     211896.0  0.000000e+00  0.000000e+00   0.0   
delivery              211944.0 -2.559072e-01  9.641721e-01  -1.0   
takeaway              211944.0 -1.000000e+00  0.000000e+00  -1.0   

                               25%           50%           75%  
max  
res_id                3.301027e+06  1.869573e+07  1.881297e+07  
1.915979e+07  
city_id               1.100000e+01  3.400000e+01  1.130600e+04  
1.135400e+04  
latitude              1.549607e+01  2.251449e+01  2.684167e+01  
1.000000e+04  
longitude             7.487796e+01  7.742597e+01  8.021932e+01  
9.183277e+01  
country_id            1.000000e+00  1.000000e+00  1.000000e+00  
1.000000e+00  
average_cost_for_two  2.500000e+02  4.000000e+02  7.000000e+02  
3.000000e+04  
price_range           1.000000e+00  2.000000e+00  2.000000e+00  
4.000000e+00  
```

```
aggregate_rating      3.300000e+00  3.800000e+00  4.100000e+00
4.900000e+00
votes                 1.600000e+01  1.000000e+02  3.620000e+02
4.253900e+04
photo_count           3.000000e+00  1.800000e+01  1.280000e+02
1.770200e+04
opentable_support     0.000000e+00  0.000000e+00  0.000000e+00
0.000000e+00
delivery             -1.000000e+00 -1.000000e+00  1.000000e+00
1.000000e+00
takeaway             -1.000000e+00 -1.000000e+00 -1.000000e+00 -
1.000000e+00
```

Removing duplicates

```
df.duplicated().sum()
```

```
151527
```

```
duplicated = df[df.duplicated()]
print(duplicated)
```

```
          res_id                            name      establishment  \
101      3400059            Peshawri - ITC Mughal    ['Fine Dining']
116      3400060            Taj Bano - ITC Mughal    ['Fine Dining']
140      3400017                   Pinch Of Spice  ['Casual Dining']
141      3400018                   Pinch Of Spice  ['Casual Dining']
142      3400850                       Urban Deck  ['Casual Dining']
...          ...                              ...                ...
211937  18855810               Biryani aur Baatein  ['Casual Dining']
211938  18662583                      Wok On Fire  ['Casual Dining']
211939   3202251  Kali Mirch Cafe And Restaurant  ['Casual Dining']
211941  18984164                 The Grand Thakar  ['Casual Dining']
211943  18879846     Freshco's - The Health Cafe          ['Café']

                                                      url  \
101     https://www.zomato.com/agra/peshawri-itc-mugha...
116     https://www.zomato.com/agra/taj-bano-itc-mugha...
140     https://www.zomato.com/agra/pinch-of-spice-civ...
141     https://www.zomato.com/agra/pinch-of-spice-taj...
142     https://www.zomato.com/agra/urban-deck-2-civil...
...                                                   ...
211937  https://www.zomato.com/vadodara/biryani-aur-ba...
211938  https://www.zomato.com/vadodara/wok-on-fire-fa...
211939  https://www.zomato.com/vadodara/kali-mirch-caf...
211941  https://www.zomato.com/vadodara/the-grand-thak...
211943  https://www.zomato.com/vadodara/freshcos-the-h...

                                                  address      city
city_id  \
```

```
101                ITC Mughal, Fatehabad Road, Tajganj, Agra      Agra
34
116                ITC Mughal, Fatehabad Road, Tajganj, Agra      Agra
34
140     23/453, Opposite Sanjay Cinema, Wazipura Road,...      Agra
34
141                    1076/2, Fatehabad Road, Tajganj, Agra      Agra
34
142     5th Floor, The P L Palace Hotel, MG Road, Sanj...      Agra
34
...                                                    ...       ...
...
211937  Shop 14, Atlantis K-10, A Wing, Genda Circle R...  Vadodara
32
211938  Ground Floor 1, Rossette Building, Opposite Se...  Vadodara
32
211939  Manu Smriti Complex, Near Navrachna School, GI...  Vadodara
32
211941  3rd Floor, Shreem Shalini Mall, Opposite Conqu...  Vadodara
32
211943  Shop 7, Ground Floor, Opposite Natubhai Circle...  Vadodara
32

                     locality   latitude  longitude  ... price_range
currency  \
101     ITC Mughal, Tajganj  27.161150  78.043993  ...           4
Rs.
116     ITC Mughal, Tajganj  27.161132  78.044022  ...           4
Rs.
140             Civil Lines  27.201735  78.007625  ...           4
Rs.
141                 Tajganj  27.159649  78.043304  ...           4
Rs.
142             Civil Lines  27.199573  78.003699  ...           4
Rs.
...                     ...        ...        ...  ...         ...
...
211937             Alkapuri  22.317746  73.168043  ...           2
Rs.
211938           Fatehgunj  22.323357  73.187461  ...           3
Rs.
211939           Fatehgunj  22.336931  73.192356  ...           2
Rs.
211941             Alkapuri  22.310563  73.171163  ...           2
Rs.
211943             Vadiwadi  22.309935  73.158768  ...           2
Rs.

                                               highlights
```

```
aggregate_rating  \
101     ['Lunch', 'Cash', 'Credit Card', 'Dinner', 'De...
4.4
116     ['Credit Card', 'Lunch', 'Cash', 'Debit Card',...
4.3
140     ['Lunch', 'Delivery', 'Credit Card', 'Dinner',...
4.6
141     ['Delivery', 'Dinner', 'Cash', 'Credit Card', ...
4.6
142     ['Dinner', 'Cash', 'Debit Card', 'Takeaway Ava...
4.3
...                                                   ...
...
211937  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.1
211938  ['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...
4.0
211939  ['Dinner', 'Cash', 'Lunch', 'Delivery', 'Indoo...
4.1
211941  ['Dinner', 'Cash', 'Debit Card', 'Lunch', 'Tak...
4.0
211943  ['Dinner', 'Cash', 'Takeaway Available', 'Debi...
4.0

       rating_text  votes  photo_count opentable_support delivery
takeaway
101      Very Good    353          154               0.0       -1
-1
116      Very Good     96          205               0.0       -1
-1
140      Excellent    915          105               0.0        1
-1
141      Excellent    965          690               0.0        1
-1
142      Very Good    672          192               0.0        1
-1
...            ...    ...          ...               ...      ...
...
211937   Very Good    154           96               0.0       -1
-1
211938   Very Good    301          126               0.0        1
-1
211939   Very Good    243           40               0.0       -1
-1
211941   Very Good    111           38               0.0       -1
-1
211943   Very Good     93           53               0.0        1
-1
```

```
[151527 rows x 26 columns]
```

Removing duplicates across all columns

```
df.drop_duplicates(inplace=True)

df.duplicated().sum()

0
```

Dealing with missing values

```
df[df['address']==""]

Empty DataFrame
Columns: [res_id, name, establishment, url, address, city, city_id,
locality, latitude, longitude, zipcode, country_id, locality_verbose,
cuisines, timings, average_cost_for_two, price_range, currency,
highlights, aggregate_rating, rating_text, votes, photo_count,
opentable_support, delivery, takeaway]
Index: []

[0 rows x 26 columns]

df.isna().sum()

res_id                      0
name                        0
establishment               0
url                         0
address                    18
city                        0
city_id                     0
locality                    0
latitude                    0
longitude                   0
zipcode                 47869
country_id                  0
locality_verbose            0
cuisines                  470
timings                  1070
average_cost_for_two        0
price_range                 0
currency                    0
highlights                  0
aggregate_rating            0
rating_text                 0
votes                       0
photo_count                 0
```

```
opentable_support         19
delivery                   0
takeaway                   0
dtype: int64
```

Basic Statistics

Average Rating

```
print(f"Average Rating: {df['aggregate_rating'].mean()}")
```

```
Average Rating: 3.032868232451132
```

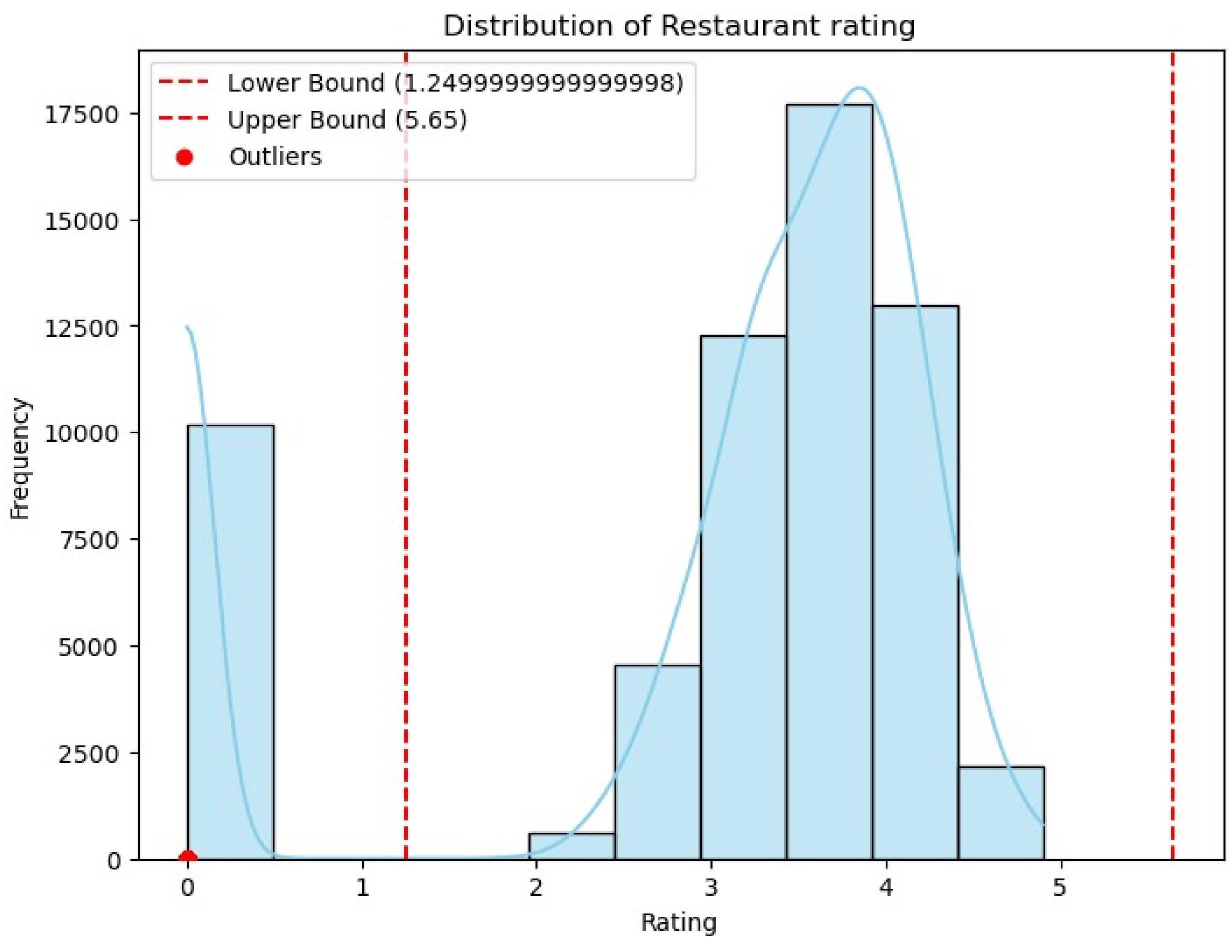
Distribution of ratings

```
# calculate the IQR
Q1=df['aggregate_rating'].quantile(0.25)
Q3=df['aggregate_rating'].quantile(0.75)
IQR=Q3-Q1

# define outlier range
lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR

# Identify outliers
outliers = df[(df['aggregate_rating'] < lower_bound) |
(df['aggregate_rating'] > upper_bound)]

# Display the calculated values
print(f"Q1: {Q1}")
print(f"Q3: {Q3}")
print(f"IQR: {IQR}")
print(f"lower: {lower_bound}")
print(f"upper: {upper_bound}")
```

```
Q1: 2.9
Q3: 4.0
IQR: 1.1
lower: 1.2499999999999998
upper: 5.65
```

```
# Create a histogram to visualize the distribution of the data
plt.figure(figsize=(8, 6))
sns.histplot(df['aggregate_rating'], bins=10, kde=True,
color='skyblue', edgecolor='black')

# Add lines for the lower and upper bounds
plt.axvline(x=lower_bound, color='red', linestyle='--', label=f'Lower
Bound ({lower_bound})')
plt.axvline(x=upper_bound, color='red', linestyle='--', label=f'Upper
```

```
Bound ({upper_bound})')

# Highlight the outliers
outlier_values = df[(df['aggregate_rating'] < lower_bound) |
(df['aggregate_rating'] > upper_bound)]['aggregate_rating']
plt.scatter(outlier_values, np.zeros_like(outlier_values),
color='red', label='Outliers', zorder=5)

# Add title and labels
plt.title('Distribution of Restaurant rating')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.legend()

# Show the plot
plt.show()
```



Observations:

The distribution of restaurant ratings is right-skewed, with a majority of ratings falling between 3 and 4. There are also some outliers below the lower bound, indicating very low ratings.

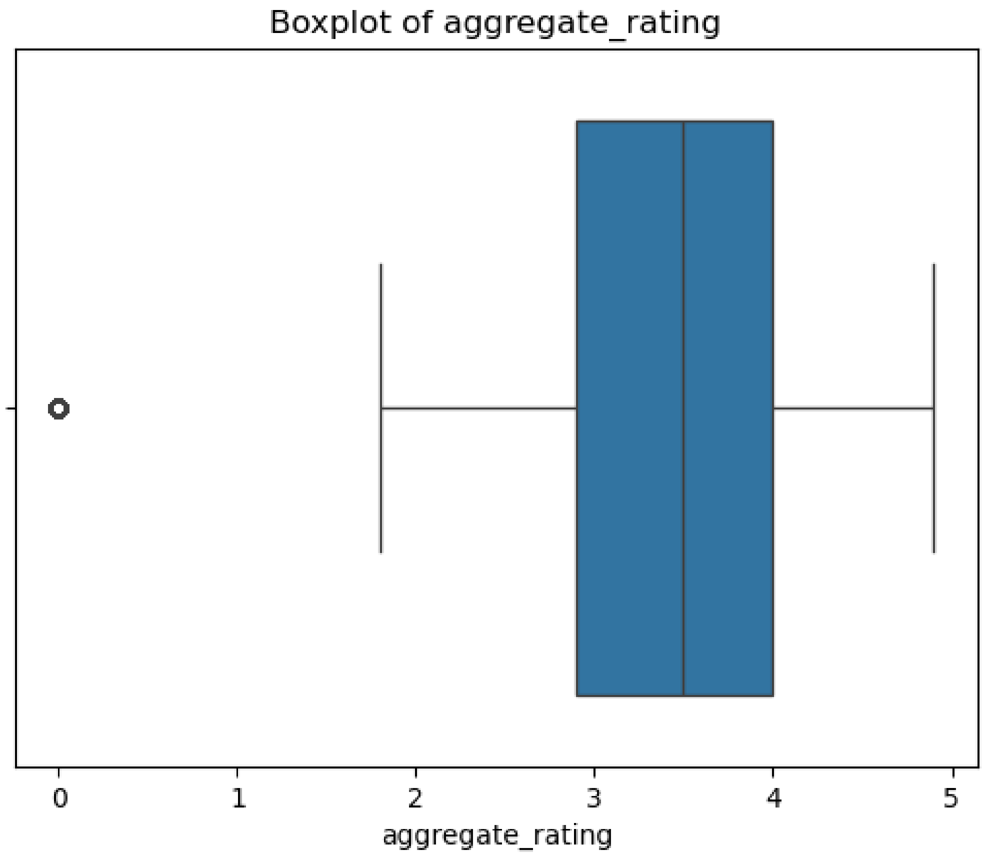
Recommendations:

Focus on High-Rated Restaurants: Prioritize marketing and promotions for restaurants with high ratings (4 and above) to attract more customers.

Address Low-Rated Restaurants: Identify the reasons for low ratings and take corrective actions, such as improving service quality, food quality, or ambiance.

Customer Feedback Analysis: Regularly analyze customer feedback and reviews to identify areas for improvement and implement necessary changes.

```python
sns.boxplot(x='aggregate_rating', data=df)
plt.title('Boxplot of aggregate_rating')
plt.show()

# Calculate quartiles
Q1 = df['aggregate_rating'].quantile(0.25)
Q3 = df['aggregate_rating'].quantile(0.75)
IQR = Q3 - Q1

# Define threshold for outliers
threshold = 1.5 * IQR

# Identify outliers
outliers = df[(df['aggregate_rating'] < Q1 - threshold) |
(df['aggregate_rating'] > Q3 + threshold)]
print(outliers)
```

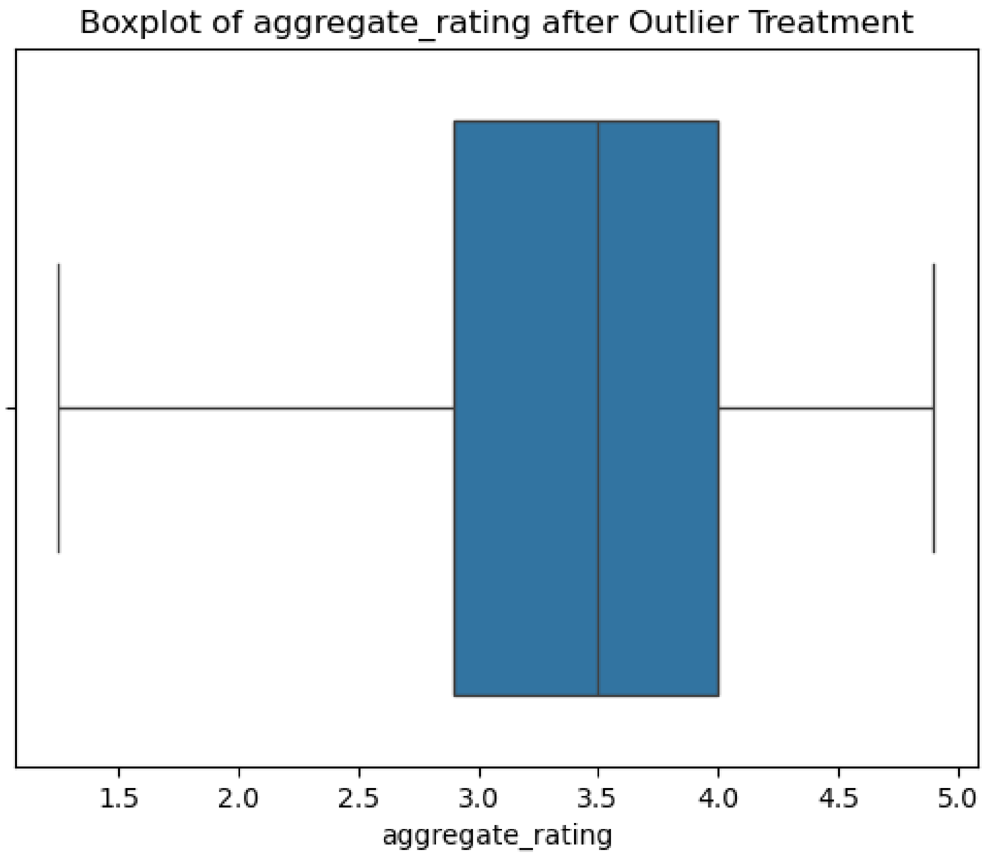



```
          res_id                                    name  
establishment  \
103      3400560            Skydeck - The Gateway Hotel    ['Fine 
Dining']
114      3401117  The Roof Top - Four Points By Sheraton    ['Fine 
Dining']
132      3400120          The Tequila - Mansingh Palace   
['Bar']
133      3400115                The Bar - Trident Hotel   
['Bar']
134      3400427                             Shahenshah  ['Casual 
Dining']
...          ...                                    ...   
...
210590  19148927                            K.G.N Hotel   
['Bhojanalya']
210592  18755421                       Super Rajni Dosa   
['Bhojanalya']
210593  18891247                 Jay Jalaram Bhojanalay   
['Bhojanalya']
210594  18929553                 Jay Jalaram Bhojanalay   
['Bhojanalya']
210802  18725986               Shiv laheri tea and food    ['Quick 
```

```
Bites']

                                                      url  \
103     https://www.zomato.com/agra/skydeck-the-gatewa...
114     https://www.zomato.com/agra/the-roof-top-four-...
132     https://www.zomato.com/agra/the-tequila-mansin...
133     https://www.zomato.com/agra/the-bar-trident-ho...
134     https://www.zomato.com/agra/shahenshah-rakabga...
...                                                   ...
210590  https://www.zomato.com/vadodara/k-g-n-hotel-ka...
210592  https://www.zomato.com/vadodara/super-rajni-do...
210593  https://www.zomato.com/vadodara/jay-jalaram-bh...
210594  https://www.zomato.com/vadodara/jay-jalaram-bh...
210802  https://www.zomato.com/vadodara/shiv-laheri-te...

                                                  address      city
city_id  \
103      The Gateway Hotel, Fatehabad Road, Tajganj, Agra      Agra
34
114     Four Points by Sheraton, Tin ka Nagla Road, Ta...      Agra
34
132        Mansingh Palace, Fatehabad Road, Tajganj, Agra      Agra
34
133          Trident Hotel, Fatehabad Road, Tajganj, Agra      Agra
34
134     32/107 A, Hotel Grand Imperial, Opposite D M C...      Agra
34
...                                                   ...       ...
...
210590  Opposite Nishant Complex, Karelibaug Road, Naw...  Vadodara
32
210592  SB 27, Race Course Towers, Near Natubhai Circl...  Vadodara
32
210593  Opposite Kukum Marriage Hall, Upasana Society,...  Vadodara
32
210594  Opposite Kumkum Party Plot, Near Chhani Jakat ...  Vadodara
32
210802  Shop GF-19, Block A, Signet Plaza, Arunachal G...  Vadodara
32

                                     locality   latitude
longitude  ...  \
103                 The Gateway Hotel, Tajganj  27.157372
78.037444  ...
114     Four Points by Sheraton Agra, Tajganj  27.158822
78.054014  ...
132                   Mansingh Palace, Tajganj  27.161227
78.035364  ...
133                     Trident Hotel, Tajganj  27.159558
78.059922  ...
```

```
134                                   Rakabganj  27.173421
78.009467  ...
...                                         ...        ...        ...  .
..
210590                               Karelibaug  22.305794
73.198954  ...
210592                                 Vadiwadi  22.308788
73.159577  ...
210593                                Nizampura  22.346887
73.175496  ...
210594                                     Sama  22.347004
73.175646  ...
210802                                    Gotri  22.320567
73.141505  ...

       price_range  currency  \
103              4       Rs.
114              4       Rs.
132              4       Rs.
133              4       Rs.
134              4       Rs.
...            ...       ...
210590           1       Rs.
210592           1       Rs.
210593           1       Rs.
210594           1       Rs.
210802           1       Rs.

                                                 highlights
aggregate_rating  \
103     ['Dinner', 'Lunch', 'Credit Card', 'Breakfast'...
0.0
114     ['Debit Card', 'Credit Card', 'Dinner', 'Cash'...
0.0
132     ['Cash', 'Lunch', 'Serves Alcohol', 'Debit Car...
0.0
133     ['Serves Alcohol', 'Debit Card', 'Cash', 'Cred...
0.0
134     ['Lunch', 'Dinner', 'Cash', 'Takeaway Availabl...
0.0
...                                                   ...
...
210590  ['Dinner', 'Cash', 'Lunch', 'Takeaway Availabl...
0.0
210592  ['Lunch', 'Breakfast', 'Takeaway Available', '...
0.0
210593  ['Dinner', 'Takeaway Available', 'Lunch', 'Cas...
0.0
210594  ['Takeaway Available', 'Cash', 'Lunch', 'Dinne...
```

```
0.0
210802  ['Dinner', 'Cash', 'Breakfast', 'Lunch', 'Lunc...
0.0

       rating_text  votes  photo_count opentable_support delivery
takeaway
103      Not rated      3            4               0.0       -1
-1
114      Not rated      0            2               0.0       -1
-1
132      Not rated      2            5               0.0       -1
-1
133      Not rated      3            0               0.0       -1
-1
134      Not rated      1            0               0.0       -1
-1
...            ...    ...          ...               ...      ...
...
210590   Not rated      1            1               0.0       -1
-1
210592   Not rated      2            0               0.0       -1
-1
210593   Not rated      0            0               0.0       -1
-1
210594   Not rated      0            0               0.0       -1
-1
210802   Not rated      3            5               0.0       -1
-1

[10159 rows x 26 columns]

# Handle the outliers at the threshold values
df['aggregate_rating'] = df['aggregate_rating'].clip(lower=Q1 -
threshold, upper=Q3 + threshold)

# Recheck the boxplot
sns.boxplot(x='aggregate_rating', data=df)
plt.title('Boxplot of aggregate_rating after Outlier Treatment')
plt.show()
```



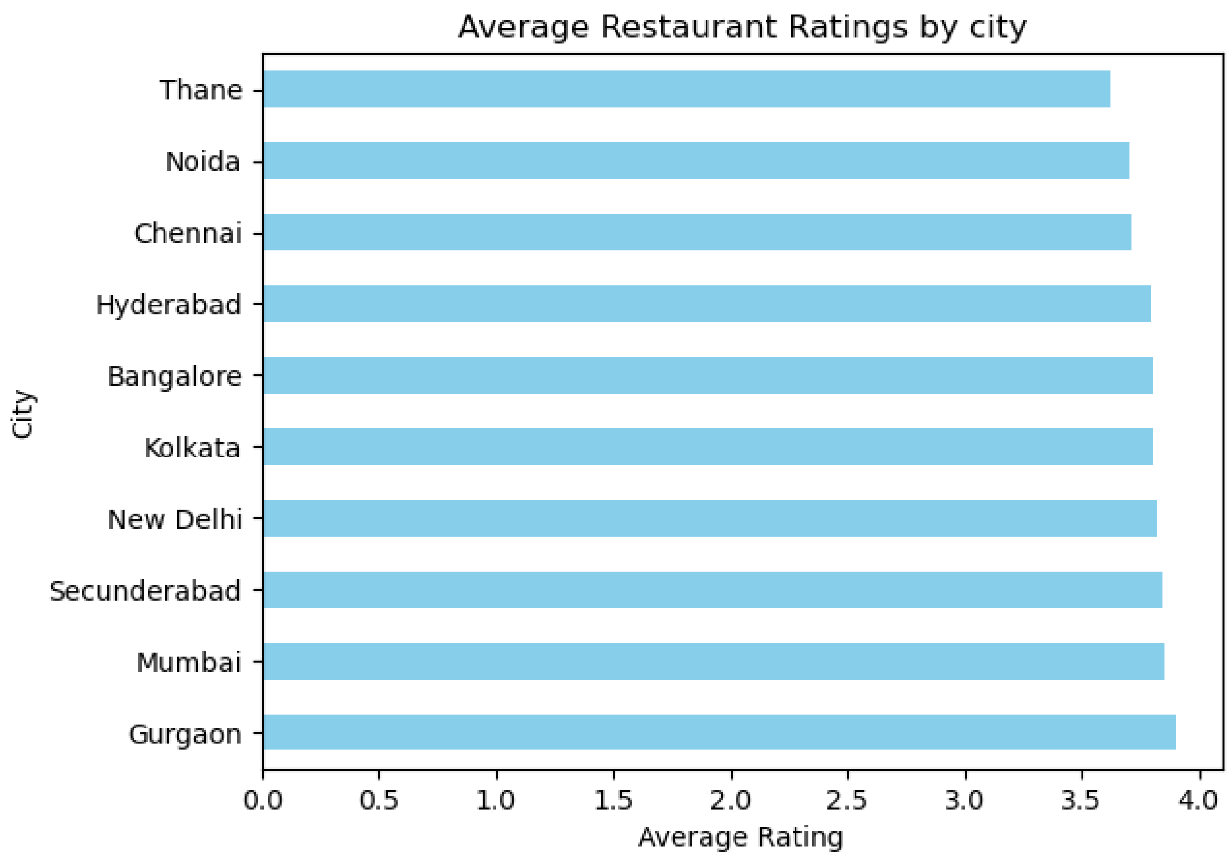
Location Analysis

City with the highest concentration of restaurants

```
df['city'].value_counts()

city
Chennai          2612
Mumbai           2538
Bangalore        2365
Pune             1911
New Delhi        1847
                 ...
Udupi              61
Howrah             50
Neemrana           26
Greater Noida      22
Nayagaon           15
Name: count, Length: 99, dtype: int64
```

Visualize restaurant rating by city

```
city_counts = df['city'].value_counts()

city_ratings = df.groupby('city')['aggregate_rating'].mean()
city_ratings.sort_values(ascending=False).head(10).plot(kind='barh',co
lor='skyblue')
plt.title('Average Restaurant Ratings by city')
plt.xlabel('Average Rating')
plt.ylabel('City')
plt.show()
```



Observations:

The average restaurant ratings are relatively high across all cities, with Gurgaon having the highest average rating. There is not a significant difference in ratings between cities.

Recommendations:

Maintain High Standards: Zomato should continue to maintain high standards for restaurant partners to ensure consistent quality across all cities.

Targeted Marketing: While all cities have high ratings, targeted marketing campaigns can be implemented to highlight specific cuisines, restaurants, or promotions in each city to drive sales.

Customer Feedback Analysis: Regularly analyze customer feedback and reviews to identify areas for improvement and implement necessary changes in specific cities.

Cuisine Analysis

handling of missing value from cuisines

forward fill missing value in the 'cuisines' column

```
missing_cuisines_count = df['cuisines'].isna().sum()
print(missing_cuisines_count)
```

```
470
```

```
df['cuisines'] = df['cuisines'].ffill()
```

```
missing_cuisines_count = df['cuisines'].isna().sum()
print(missing_cuisines_count)
```

```
0
```

Most Popular Cuisines among restaurants

```
cuisine_counts = df['cuisines'].value_counts()
cuisine_counts.head(10)
```

```
cuisines
North Indian                    4690
```

```
Fast Food                    2177
North Indian, Chinese        1815
Bakery                       1626
South Indian                 1626
Street Food                  1224
Cafe                         1180
Mithai                       1043
Desserts                      954
Bakery, Desserts              874
Name: count, dtype: int64
```

Correlation between the variety of cuisines and ratings

```
df['new_cuisines'] = df['cuisines'].apply(lambda x: len(x.split(',')))

# Create the scatter plot with Plotly Express
fig = px.scatter(df, x='new_cuisines', y='aggregate_rating',
                 title='Cuisine Variety Vs Rating',
                 labels={'new_cuisines': 'Number of Cuisines',
'aggregate_rating': 'Rating'})

# Show the plot
fig.show()
```

Observations:

There doesn't seem to be a strong correlation between the number of cuisines offered by a restaurant and its rating.

Restaurants with a wide range of cuisines (up to 8) have similar ratings to those with fewer cuisines.

Recommendations:

Focus on Quality Over Quantity: Rather than focusing on offering a wide variety of cuisines, restaurants should prioritize offering high-quality dishes within a few core cuisines.

Customer Feedback Analysis: Analyze customer feedback to understand the most popular cuisines and dishes, and focus on improving these offerings.

Unique Selling Proposition: Restaurants should aim to differentiate themselves by offering unique dishes or dining experiences, rather than simply focusing on the number of cuisines.

Efficient Operations: Offering a wide variety of cuisines can increase operational complexity and costs. Restaurants should focus on streamlining operations and optimizing their menu to maintain quality and profitability.

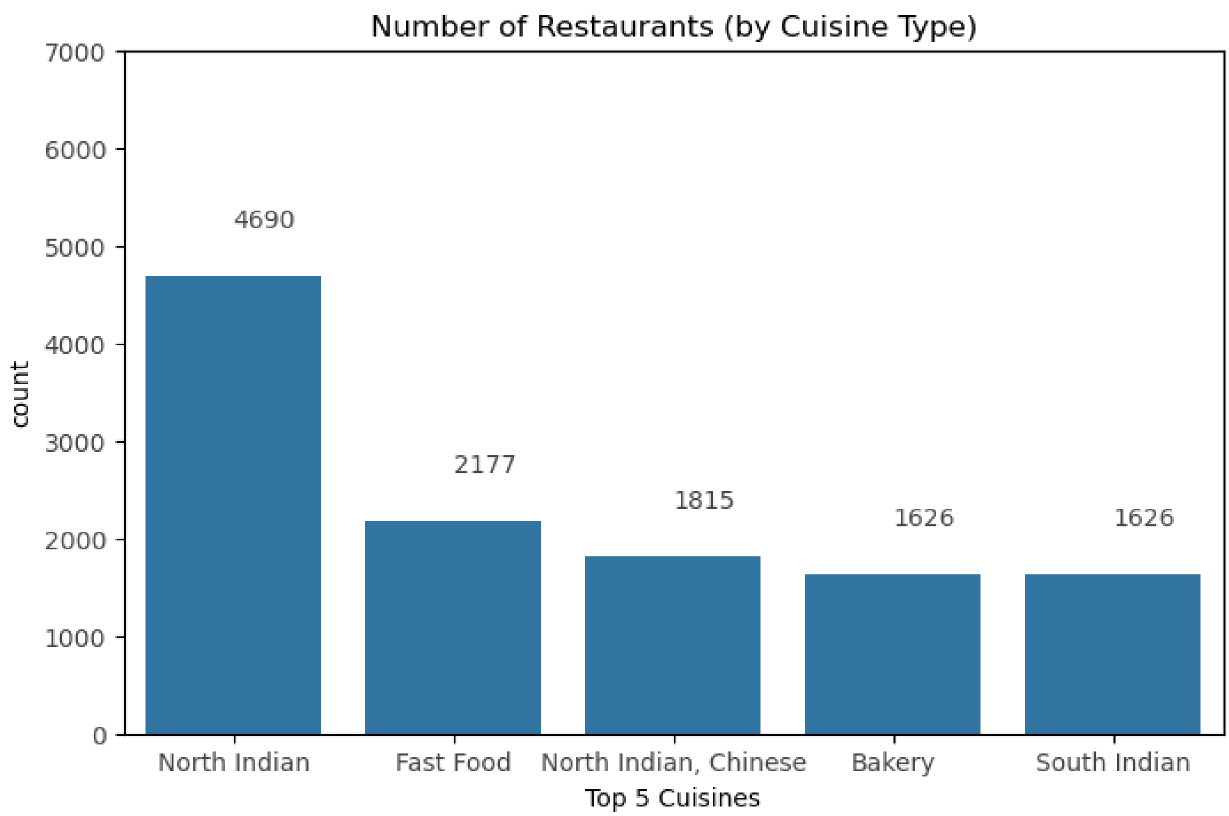
Number of Restaurants (By Cuisine)

```
cuisiness = df['cuisines']
# Calculate the top 5 cuisines
c_count = cuisiness.value_counts()[:5].reset_index()
c_count.columns = ['cuisine', 'count']
c_count
```

	cuisine	count
0	North Indian	4690
1	Fast Food	2177
2	North Indian, Chinese	1815
3	Bakery	1626
4	South Indian	1626

```
# Plotting with Seaborn
plt.figure(figsize=(8, 5))
```

```
sns.barplot(x='cuisine', y='count', data=c_count)
plt.xticks(color="#424242")
plt.yticks(range(0, 8000, 1000), color="#424242")
plt.xlabel("Top 5 Cuisines")
plt.title("Number of Restaurants (by Cuisine Type)")

# Adding labels on bars
for index, value in enumerate(c_count['count']):
    plt.text(index, value + 500, str(value), color='#424242')

plt.show()
```



Observations:

North Indian cuisine has the highest number of restaurants, followed by Fast Food and North Indian, Chinese.

Bakery and South Indian cuisines have a significantly lower number of restaurants.

Recommendations:

Focus on Popular Cuisines: Zomato should continue to focus on expanding the availability of popular cuisines like North Indian and Fast Food, as they have a high demand.

Promote Less Popular Cuisines: Zomato can promote less popular cuisines like Bakery and South Indian through targeted marketing campaigns and special offers to increase their visibility and attract customers.

Data-Driven Expansion: Utilize data analytics to identify areas with high demand for specific cuisines and encourage restaurants to open in those areas.

Price Range And Rating

```python
# Calculate the value counts for price ranges
pr_count = df.groupby("price_range").count()["name"].reset_index()
pr_count.columns = ['price_range', 'count']

# Plotting with Seaborn
plt.figure(figsize=(8, 5))
sns.barplot(x='price_range', y='count', data=pr_count)
plt.xticks(range(0, 5), color="#424242")
plt.yticks(range(0, 40000, 5000), color="#424242")
plt.xlabel("Price Ranges")
plt.title("Number of Restaurants (by Price Ranges)")

# Adding labels on bars
for index, value in enumerate(pr_count['count']):
    plt.text(index, value + 700, str(value), color='#424242')

plt.show()
```


Number of Restaurants (by Price Ranges)
count
Price Ranges
30680
18378
8301
3058
0
5000
10000
15000
20000
25000
30000
35000
1
2
3
4

Observations:

The distribution of restaurants across price ranges is uneven, with most restaurants falling into the lowest price range (1).

There is a significant drop in the number of restaurants as the price range increases.

Recommendations:

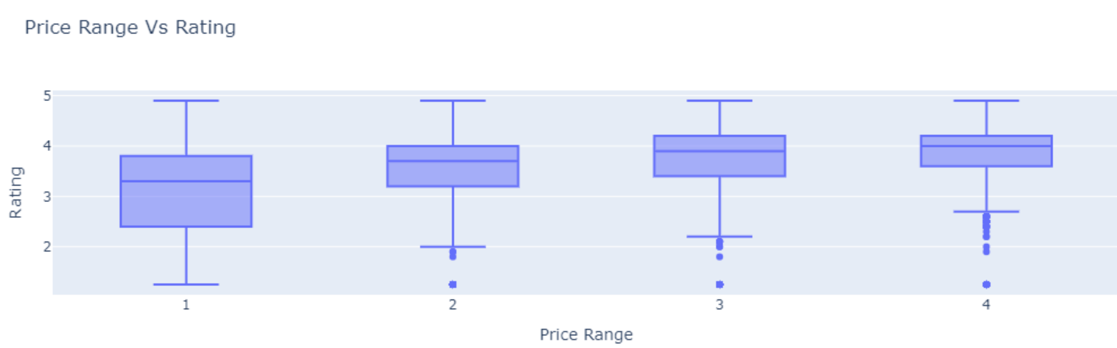
Focus on Affordable Options: Zomato should continue to focus on expanding the availability of affordable restaurants to cater to the majority of customers.

Promote High-End Dining: Zomato can promote high-end restaurants (price ranges 3 and 4) through targeted marketing campaigns and exclusive offers to attract a premium customer segment.

Price Range And Rating

Relationship Between Price Range and Ratings

```
fig = px.box(df, x='price_range', y='aggregate_rating',
             title='Price Range Vs Rating',
             labels={'price_range': 'Price Range', 'aggregate_rating':
'Rating'})
fig.show()
```



Observations:

Price Range and Rating: There appears to be a slight positive correlation between price range and rating. Restaurants with higher price ranges tend to have slightly higher median ratings.

Outliers: Restaurants in higher price ranges have a few outliers with lower ratings. These could be due to specific instances of poor service or food quality.

Recommendations:

Maintain Quality and Consistency: Restaurants in higher price ranges should maintain high standards of food quality, service, and ambiance to justify the higher prices and avoid negative reviews.

Value Proposition: Restaurants in lower price ranges should focus on offering good value for money and ensuring a positive customer experience to maintain higher ratings.

Customer Feedback Analysis: Regularly analyze customer feedback and reviews to identify areas for improvement and address any issues that may be impacting ratings.

Targeted Marketing: Implement targeted marketing campaigns to promote high-rated restaurants and highlight their unique selling points.

Calculate the average cost for two people in different price categories

```
price_rating = df.groupby('price_range')
['average_cost_for_two'].mean()
price_rating
```

```
price_range
1     219.208605
2     524.777941
3    1104.843874
4    2283.108568
Name: average_cost_for_two, dtype: float64
```

Online Orders and Table Booking

Investigate the Impact of Online Order Availability on Restaurant Ratings

Categorize Restaurants by Online Order Availability

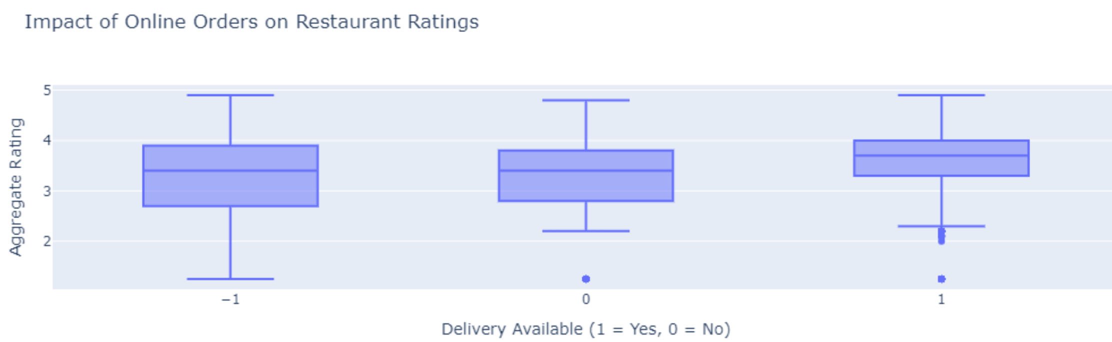
```
delivery_group = df.groupby('delivery')['aggregate_rating'].median()
delivery_group

delivery
-1    3.4
 0    3.4
 1    3.7
Name: aggregate_rating, dtype: float64
```

Perform a Statistical Test: If we want to check if the difference in ratings between the two categories (delivery vs. no delivery) is statistically significant, you can perform a t-test

```
from scipy.stats import ttest_ind
# Split the dataset into two groups: one with delivery, one without
delivery_yes = df[df['delivery'] == 1]['aggregate_rating'].dropna()
delivery_no = df[df['delivery'] == 0]['aggregate_rating'].dropna()

# Perform a t-test
t_stat, p_val = ttest_ind(delivery_yes, delivery_no)

print(f"T-statistic: {t_stat}, P-value: {p_val}")

T-statistic: 10.667179238117443, P-value: 1.720529864348582e-26
```

A p-value below 0.05 would indicate a statistically significant difference in ratings between the two groups.

```
fig = px.box(df, x='delivery', y='aggregate_rating',
             title='Impact of Online Orders on Restaurant Ratings',
             labels={'delivery': 'Delivery Available (1 = Yes, 0 = 
No)', 'aggregate_rating': 'Aggregate Rating'})

# Show the plot
fig.show()
```



Observations:

Delivery and Rating: Restaurants offering delivery generally have slightly higher ratings compared to those that don't.

Recommendations:

Prioritize Delivery: Zomato should encourage more restaurants to offer delivery services, as it seems to positively impact customer ratings.

Delivery Quality: Focus on improving delivery speed, packaging, and food quality to maintain high ratings for delivery orders.

Partner with Reliable Delivery Services: Partner with reliable delivery services to ensure timely and efficient delivery.

Calculate the average cost for two people in different price categories

```
price_rating = df.groupby('price_range')
['average_cost_for_two'].mean()
price_rating

price_range
1     219.208605
2     524.777941
3    1104.843874
4    2283.108568
Name: average_cost_for_two, dtype: float64
```

Online Orders and Table Booking

Investigate the Impact of Online Order Availability on Restaurant Ratings

Categorize Restaurants by Online Order Availability

```
delivery_group = df.groupby('delivery')['aggregate_rating'].median()
delivery_group

delivery
-1    3.4
 0    3.4
 1    3.7
Name: aggregate_rating, dtype: float64
```

Perform a Statistical Test: If we want to check if the difference in ratings between the two categories (delivery vs. no delivery) is statistically significant, you can perform a t-test

```
# Split the dataset into two groups: one with delivery, one without
delivery_yes = df[df['delivery'] == 1]['aggregate_rating'].dropna()
delivery_no = df[df['delivery'] == 0]['aggregate_rating'].dropna()

# Perform a t-test
t_stat, p_val = ttest_ind(delivery_yes, delivery_no)

print(f"T-statistic: {t_stat}, P-value: {p_val}")

T-statistic: 10.667179238117443, P-value: 1.720529864348582e-26
```

A p-value below 0.05 would indicate a statistically significant difference in ratings between the two groups

Visualize the Impact on Ratings

```
sns.boxplot(x='delivery', y='aggregate_rating', data=df)
plt.title('Impact of Online Orders on Restaurant Ratings')
plt.xlabel('Delivery Available (1 = Yes, 0 = No)')
plt.ylabel('Aggregate Rating')
plt.show()
```


Impact of Online Orders on Restaurant Ratings
Aggregate Rating
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
-1
0
1
Delivery Available (1 = Yes, 0 = No)

Observations:

Restaurants offering online delivery have a slightly higher median rating compared to those that don't.

However, there is a wider range of ratings for restaurants with online delivery, indicating more variability in customer experiences.

Recommendations:

Prioritize Delivery: Zomato should encourage more restaurants to offer online delivery to improve customer satisfaction and ratings.

Quality Control: Restaurants offering delivery should focus on maintaining food quality and packaging to ensure a positive customer experience.

Partner with Reliable Delivery Services: Partner with reliable delivery services to ensure timely and efficient delivery.

Analyze the Distribution of Restaurants Offering Table Booking

check for missing values in the opentable_support column

```
df['opentable_support'].isna().sum()
```

```
19
```

```
df['opentable_support'].fillna(df['opentable_support'].mean())
```

```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
         ...
211882    0.0
211925    0.0
211926    0.0
211940    0.0
211942    0.0
Name: opentable_support, Length: 60417, dtype: float64
```

Check the Count of Restaurants Offering Table Booking

```
df['opentable_support'].value_counts()
```

```
opentable_support
0.0    60398
Name: count, dtype: int64
```

Top Restaurant Chains

Identify top Restaurant Chains Based On the Number Of Outlets

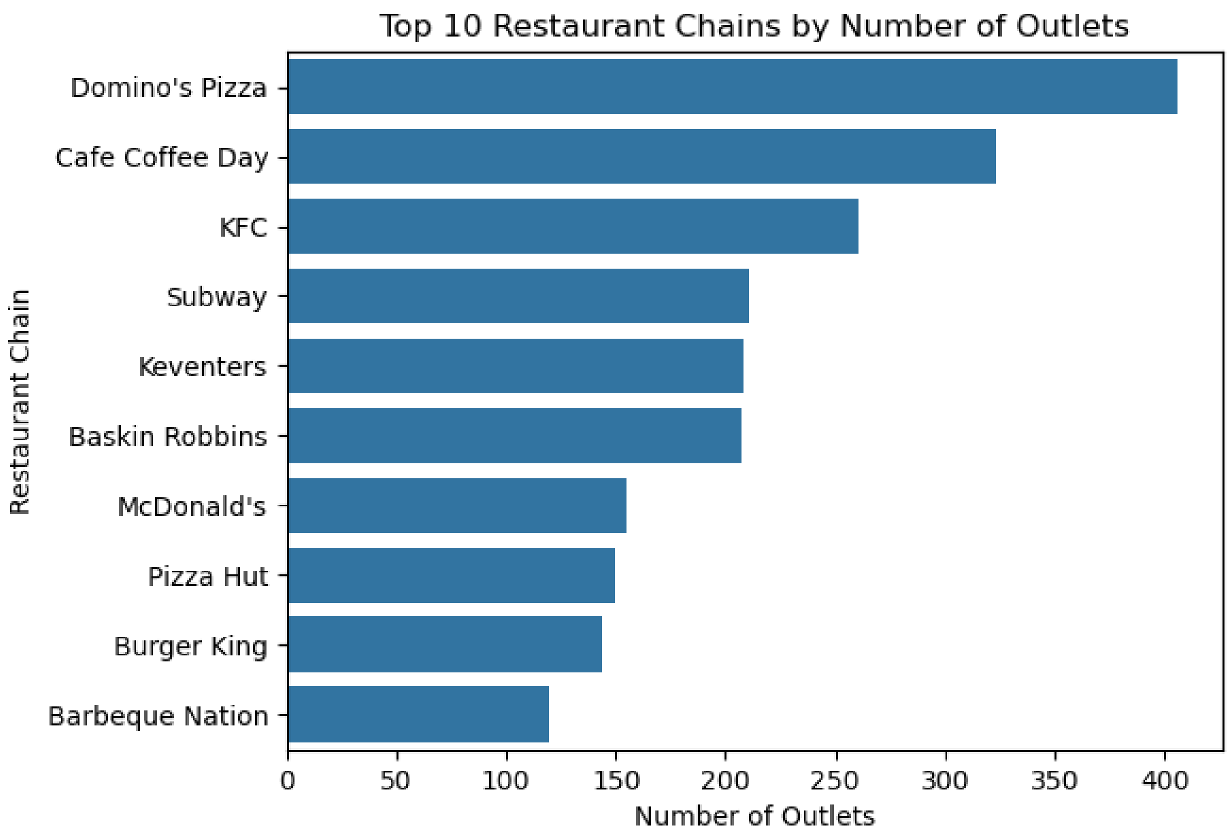
Count the number of outlets for each restaurant using the name column, and find the top chains

```
restaurant_counts = df['name'].value_counts()
top_chains = restaurant_counts.head(10)
top_chains
```

```
name
Domino's Pizza      406
Cafe Coffee Day     323
KFC                 261
Subway              211
Keventers           208
Baskin Robbins      207
McDonald's          155
Pizza Hut           150
Burger King         144
Barbeque Nation     120
Name: count, dtype: int64
```

Visualize the Top Restaurant Chains Based on Number of Outlets

```
sns.barplot(x=top_chains.values,y=top_chains.index)
plt.title('Top 10 Restaurant Chains by Number of Outlets')
plt.xlabel('Number of Outlets')
plt.ylabel('Restaurant Chain')
plt.show()
```



Top 10 Restaurant Chains by Number of Outlets
Domino's Pizza
Cafe Coffee Day
KFC
Subway
Keventers
Baskin Robbins
McDonald's
Pizza Hut
Burger King
Barbeque Nation
Restaurant Chain
0
50
100
150
200
250
300
350
400
Number of Outlets

Observations:

Domino's Pizza is the clear leader in terms of the number of outlets, followed by Cafe Coffee Day.

KFC, Subway, and Keventers also have a significant number of outlets.

Recommendations:

Strategic Partnerships: Zomato can partner with these top restaurant chains to offer exclusive deals, discounts, and loyalty programs to customers.

Data-Driven Insights: Utilize data analytics to identify high-performing outlets and optimize marketing efforts accordingly.

Geographic Expansion: Encourage these chains to expand their presence in areas with high demand and limited competition.

Explore the Ratings of the Top Chains

Calculate Average Rating for the Top Chains

```
avg_ratings = df.groupby('name')['aggregate_rating'].mean()
avg_ratings
```

```
name
# Wednesday                                       3.5
#1, Culinary Avenue - The Red Maple               3.9
#788 Avenue                                       3.9
#BC                                               4.2
#BEiR                                             4.1
                                                 ...
 Food Street - Veg                                2.9
ट 4 Tasty                                         3.7
द Vege टेबल                          4.2
स्पेस Bar                                        4.3
ह-tea The Tea Hut                                 4.2
Name: aggregate_rating, Length: 41100, dtype: float64
```

filter for the top chains

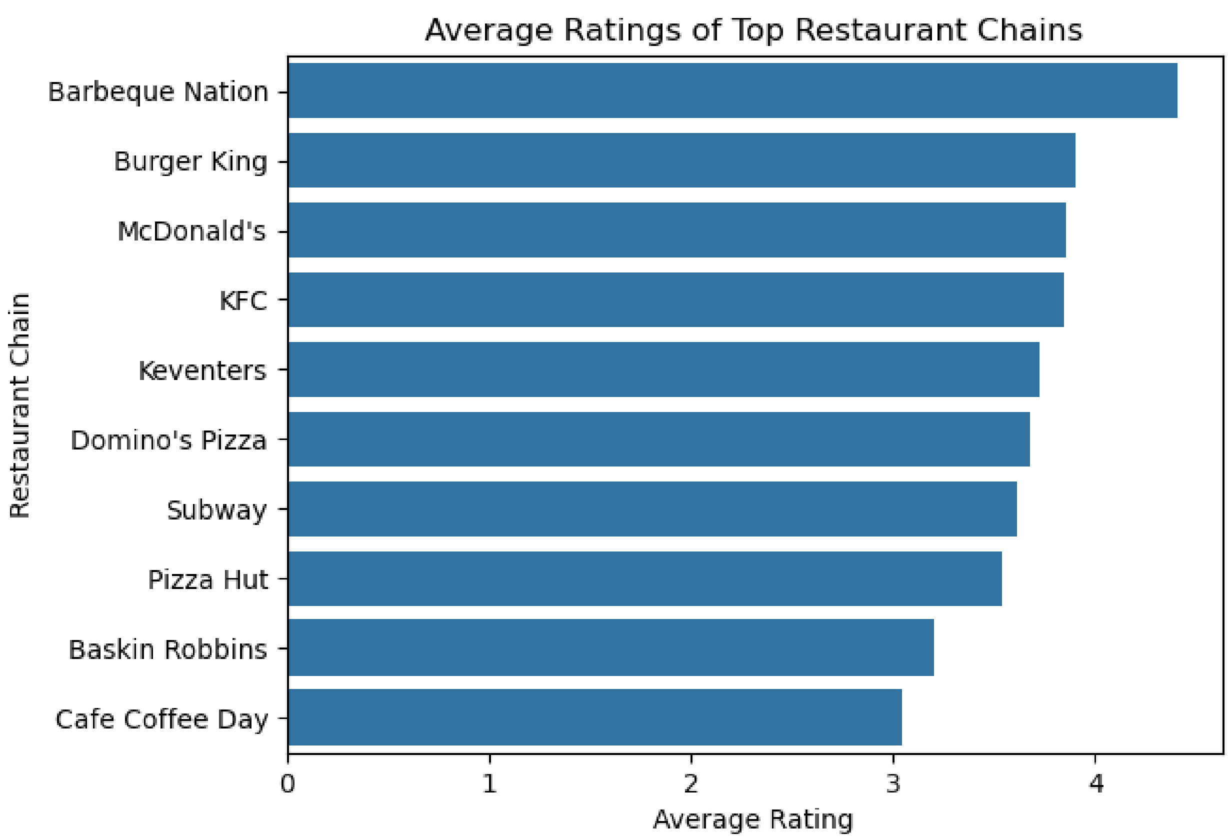
```
top_chains_rating = avg_ratings[top_chains.index]
top_chains_rating
# Sort the average ratings in ascending order
```

```
top_chains_ratings = top_chains_rating.sort_values(ascending=False)
top_chains_ratings
```

```
name
Barbeque Nation     4.407917
Burger King         3.902083
McDonald's          3.857097
KFC                 3.845785
Keventers           3.728846
Domino's Pizza      3.681527
Subway              3.612322
Pizza Hut           3.535333
Baskin Robbins      3.199034
Cafe Coffee Day     3.043653
Name: aggregate_rating, dtype: float64
```

Visualize the Ratings of the Top Chains

```
sns.barplot(x=top_chains_ratings.values,y=top_chains_ratings.index)
plt.title("Average Ratings of Top Restaurant Chains")
plt.xlabel("Average Rating")
plt.ylabel("Restaurant Chain")
plt.show()
```



Observations:

Barbeque Nation has the highest average rating among the top 10 restaurant chains.

Cafe Coffee Day has the lowest average rating.

Recommendations:

Highlight High-Rated Chains: Zomato can promote high-rated chains like Barbeque Nation to attract customers and boost their sales.

Identify Areas for Improvement: Analyze customer feedback and ratings for lower-rated chains like Cafe Coffee Day to identify areas for improvement and suggest corrective actions.

Partner with Top Chains: Zomato can partner with top-rated chains to offer exclusive deals and promotions to customers.

Explore the Rating Distribution for the Top Chains

Filter the dataset to include only the top chains

```
top_chains_data =df['name'].isin(top_chains.index)
top_chains_data =df['aggregate_rating']
top_chains_data
```

```
0         4.4
1         4.4
2         4.2
3         4.3
4         4.9
         ...
211882    2.9
211925    4.0
211926    3.9
211940    4.1
211942    3.7
Name: aggregate_rating, Length: 60417, dtype: float64
```

Restaurant Features:

clean the highlights column to ensure it's in a usable format for analysis

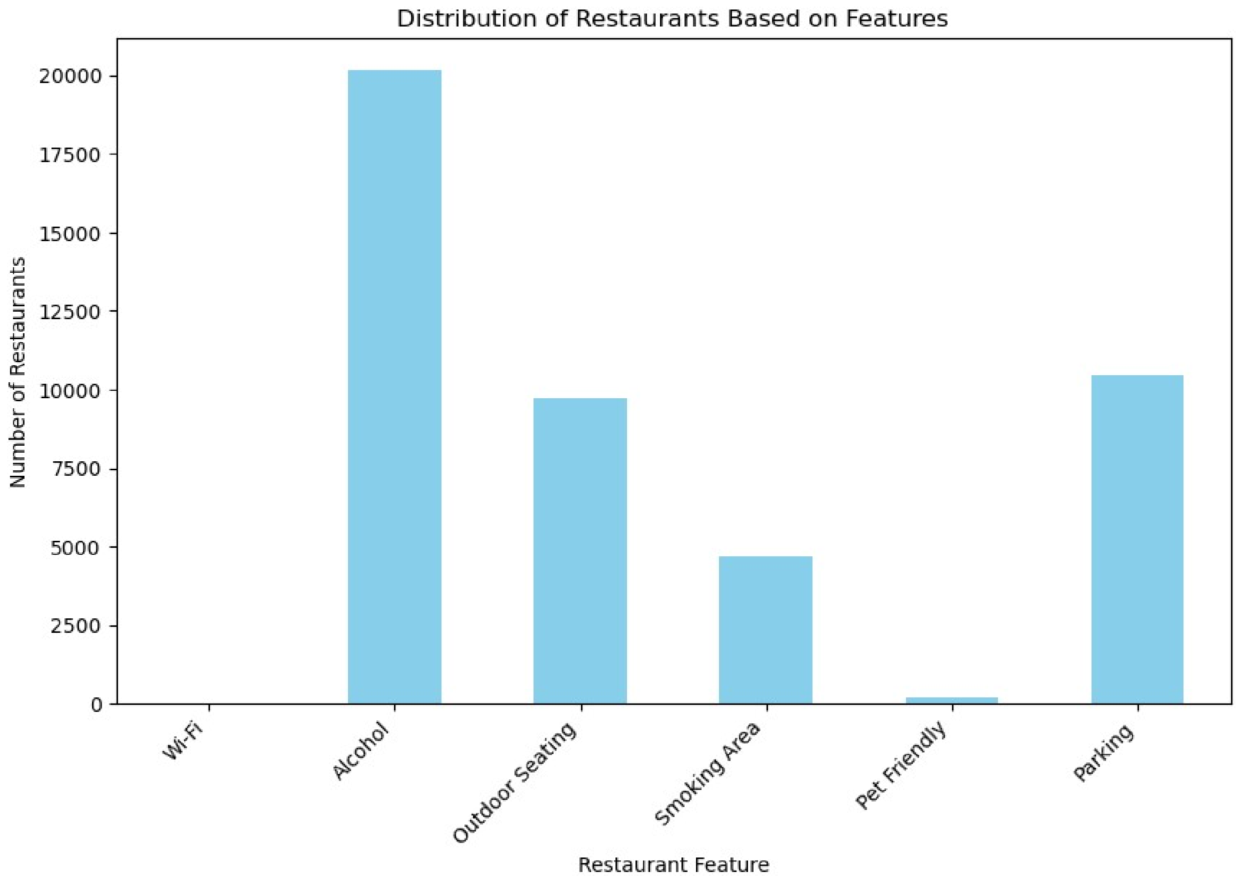
```
df['highlights'].isna().sum()

0
```

Identify and Extract Specific Features

```
# Define a list of features to check for in the 'highlights' column
features = ['Wi-Fi', 'Alcohol', 'Outdoor Seating', 'Smoking Area',
'Pet Friendly', 'Parking']

# Create new columns for each feature indicating whether the feature
is available (1) or not (0)
for feature in features:
    df[feature] = df['highlights'].apply(lambda x: 1 if feature in x
else 0)

# Check if the new columns were created successfully
(df[features].head(10))

   Wi-Fi  Alcohol  Outdoor Seating  Smoking Area  Pet Friendly
Parking
0      0        0                0             0             0
0
1      0        1                0             0             0
0
2      0        1                1             0             0
0
3      0        0                0             1             0
0
4      0        1                1             1             0
0
5      0        1                0             0             0
1
6      0        0                0             0             0
0
7      0        1                0             0             0
0
8      0        0                0             0             0
0
9      0        1                1             0             0
0
```

Analyze the Distribution of Restaurants with Features

```
# Plot the distribution of restaurants with each feature
feature_counts = df[features].sum()

plt.figure(figsize=(10, 6))
feature_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Restaurants Based on Features')
plt.xlabel('Restaurant Feature')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Observations:

Wi-Fi and Alcohol are the most common features among restaurants.

Pet-Friendly and Smoking Area are the least common features.

Recommendations:

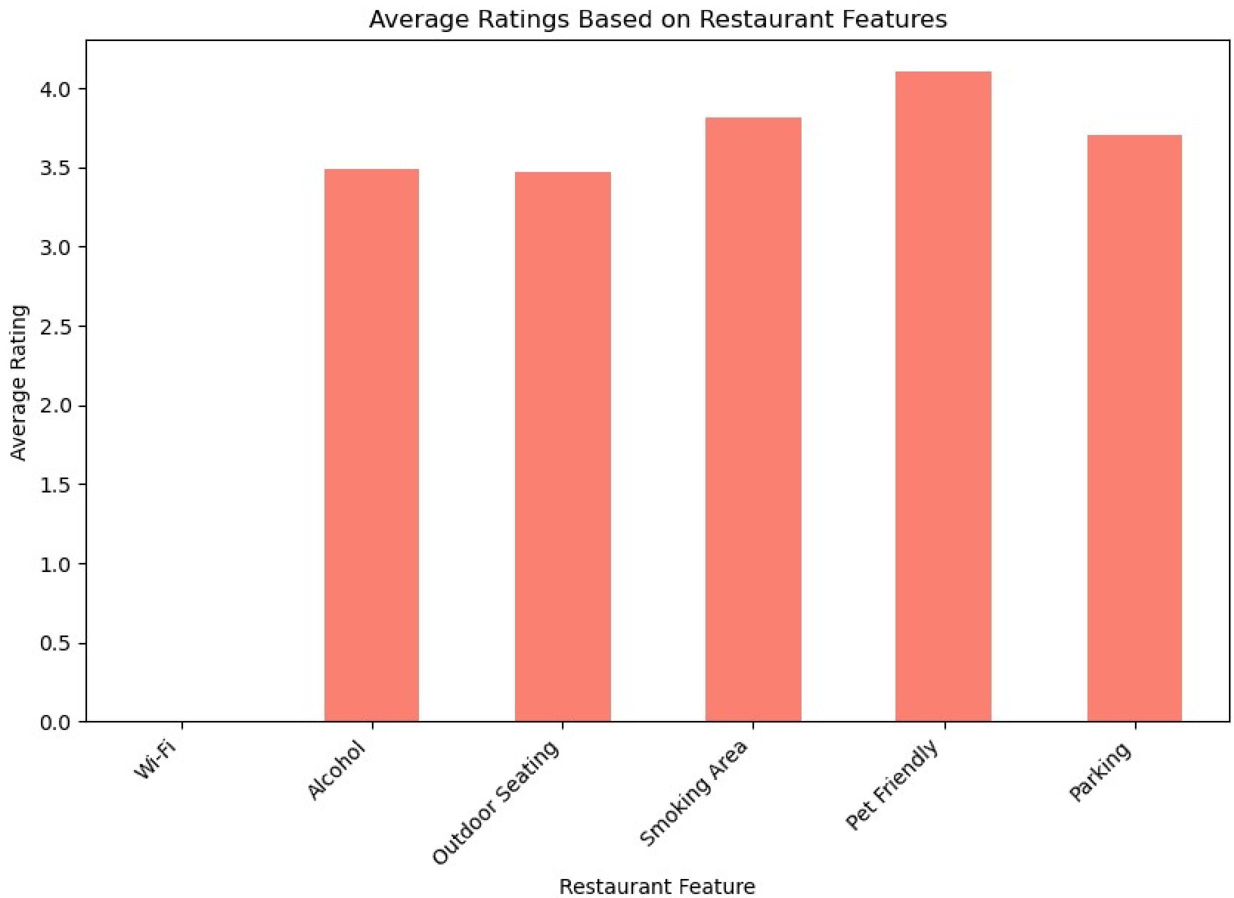
Highlight Popular Features: Promote restaurants with Wi-Fi and alcohol availability to attract customers.

Target Specific Segments: Target specific customer segments by highlighting restaurants with features like pet-friendly or outdoor seating.

Partner with Venues: Partner with venues that offer unique features like smoking areas or parking to attract a wider customer base.

Investigate Correlation Between Features and Ratings

```
# Calculate average rating for each feature (only for rows where the feature is present)
feature_ratings = {}
for feature in features:
    avg_rating = df[df[feature] == 1]['aggregate_rating'].mean()
    feature_ratings[feature] = avg_rating

# Convert the dictionary into a pandas series for easier visualization
feature_ratings_series = pd.Series(feature_ratings)

# Plot average ratings based on features
plt.figure(figsize=(10, 6))
feature_ratings_series.plot(kind='bar', color='salmon')
plt.title('Average Ratings Based on Restaurant Features')
plt.xlabel('Restaurant Feature')
plt.ylabel('Average Rating')
plt.xticks(rotation=45, ha='right')
plt.show()
```


Average Ratings Based on Restaurant Features
Average Rating
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Wi-Fi
Alcohol
Outdoor Seating
Smoking Area
Pet Friendly
Parking
Restaurant Feature

Observations:

Pet-Friendly restaurants have the highest average rating, followed by Smoking Area and Wi-Fi.

Outdoor Seating and Alcohol have slightly lower average ratings.

Recommendations:

Promote Pet-Friendly Restaurants: Highlight pet-friendly restaurants to attract customers with pets.

Improve Outdoor Seating: Enhance the outdoor seating experience by providing comfortable seating, shade, and ambiance.

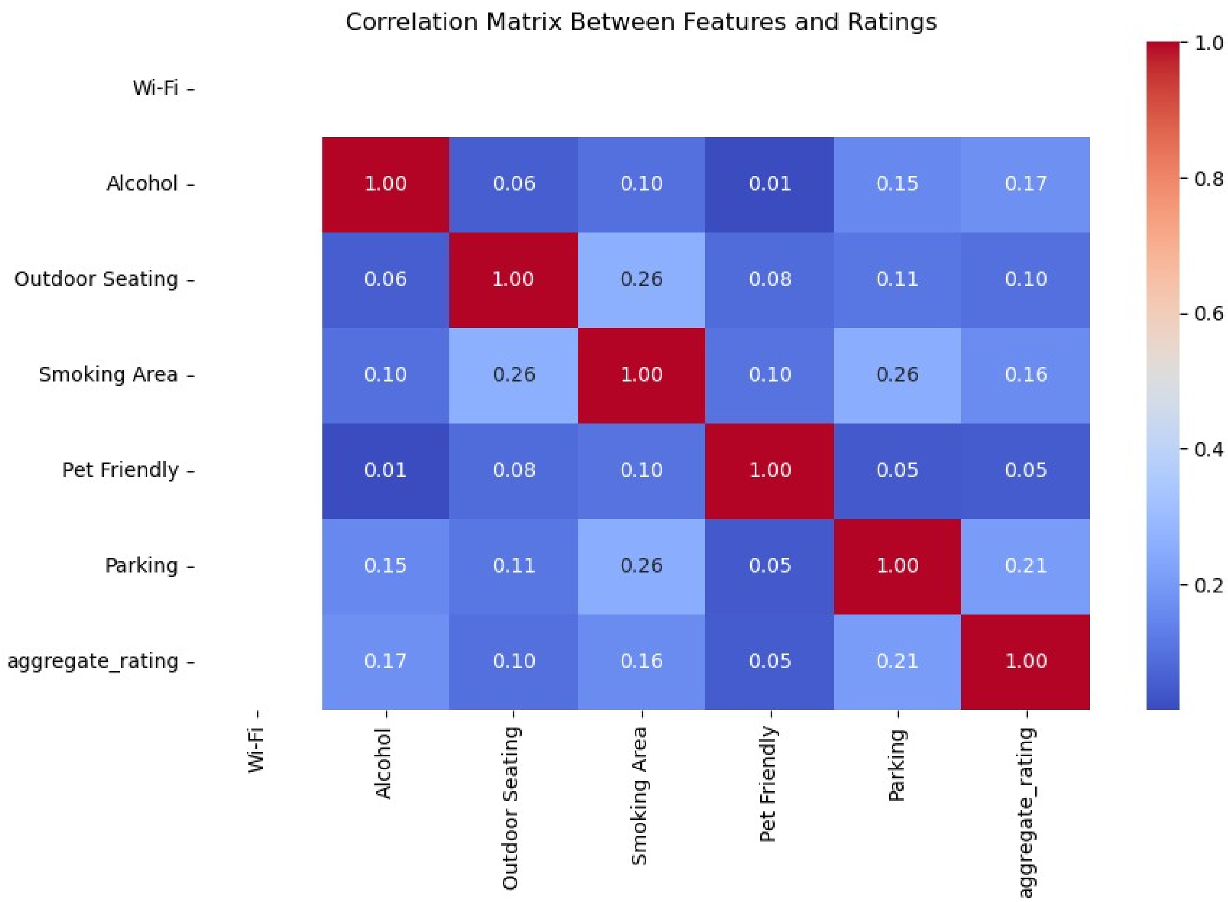
Customer Feedback Analysis: Analyze customer feedback to identify areas for improvement in restaurants with lower ratings, especially for outdoor seating and alcohol-serving establishments.

Statistical Analysis

```
# Correlation analysis between features and aggregate ratings
correlation_data = df[features + ['aggregate_rating']]
correlation_matrix = correlation_data.corr()

# Display correlation matrix
print(correlation_matrix)

# Plot the heatmap of the correlation matrix
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f')
plt.title('Correlation Matrix Between Features and Ratings')
plt.show()
```

```
                  Wi-Fi   Alcohol  Outdoor Seating  Smoking Area  \
Wi-Fi               NaN       NaN              NaN           NaN
Alcohol             NaN  1.000000         0.056342      0.097201
Outdoor Seating     NaN  0.056342         1.000000      0.259716
Smoking Area        NaN  0.097201         0.259716      1.000000
Pet Friendly        NaN  0.014851         0.084435      0.104023
Parking             NaN  0.154388         0.114670      0.257135
aggregate_rating    NaN  0.174033         0.098383      0.164597

                  Pet Friendly   Parking  aggregate_rating
```

```
Wi-Fi                   NaN       NaN              NaN
Alcohol            0.014851  0.154388         0.174033
Outdoor Seating    0.084435  0.114670         0.098383
Smoking Area       0.104023  0.257135         0.164597
Pet Friendly       1.000000  0.048823         0.050128
Parking            0.048823  1.000000         0.206227
aggregate_rating   0.050128  0.206227         1.000000
```



Observations:

Smoking Area and Outdoor Seating have the strongest positive correlation with the aggregate rating.

Pet-Friendly and Wi-Fi have a weaker correlation with the aggregate rating.

Recommendations:

Prioritize Smoking Area and Outdoor Seating: Zomato can promote restaurants with smoking areas and outdoor seating to attract customers and improve ratings.

Focus on Core Offerings: Restaurants should focus on providing high-quality food, excellent service, and a pleasant ambiance, rather than solely relying on features like Wi-Fi and pet-friendliness.

Data-Driven Marketing: Utilize data on restaurant features and ratings to optimize marketing campaigns and target customers with relevant offers.

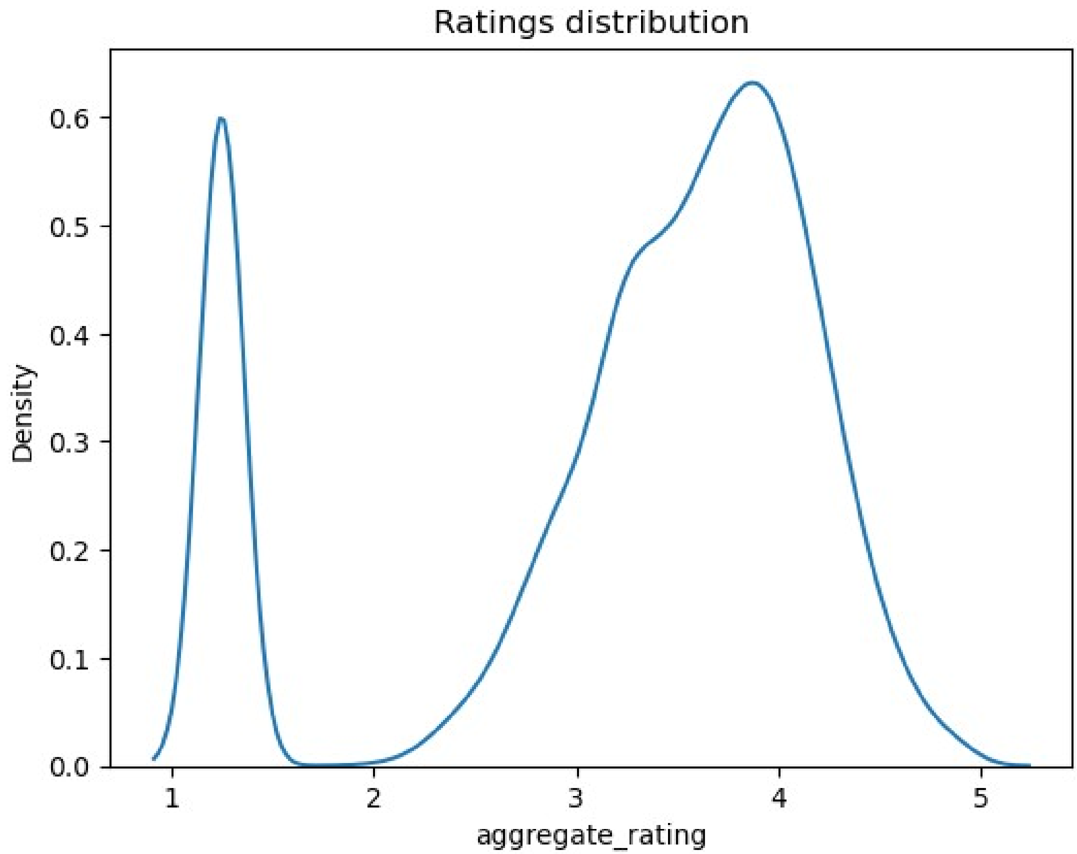
Word Cloud for Reviews

Rating and Cost

Ratings Distribution

Let's see how the ratings are distributes

```
sns.kdeplot(df['aggregate_rating'])
plt.title("Ratings distribution")
plt.show()
```


Ratings distribution
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
2
3
4
5
aggregate_rating

Observations:

The distribution of restaurant ratings is bimodal, with peaks around 1.5 and 4. This indicates that a significant proportion of restaurants either have very low ratings or very high ratings.

Recommendations:

Focus on High-Rated Restaurants: Prioritize marketing and promotions for restaurants with high ratings (4 and above) to attract more customers.

Address Low-Rated Restaurants: Identify the reasons for low ratings and take corrective actions, such as improving service quality, food quality, or ambiance.

Customer Feedback Analysis: Regularly analyze customer feedback and reviews to identify areas for improvement and implement necessary changes.

```
df['rating_text'].value_counts()
```

```
rating_text
Good             17569
Average          16782
Very Good        12714
Not rated        10159
Excellent         2065
Poor               590
Çok iyi             56
Sangat Baik         44
Muito Bom           43
Excelente           34
Muy Bueno           33
Bardzo dobrze       30
Bom                 26
Baik                24
Skvělé              24
Velmi dobré         22
İyi                 19
Harika              18
Ottimo              17
Veľmi dobré         16
Buono               14
Terbaik             14
```

```
Skvělá volba        13
Dobré               12
Bueno               11
Dobrze               9
Wybitnie             8
Eccellente           8
Vynikajúce           7
Průměr               6
Média                5
Promedio             5
Muito bom            5
Ortalama             3
Średnio              3
Priemer              3
Media                3
Biasa                2
Scarso               1
Name: count, dtype: int64
```

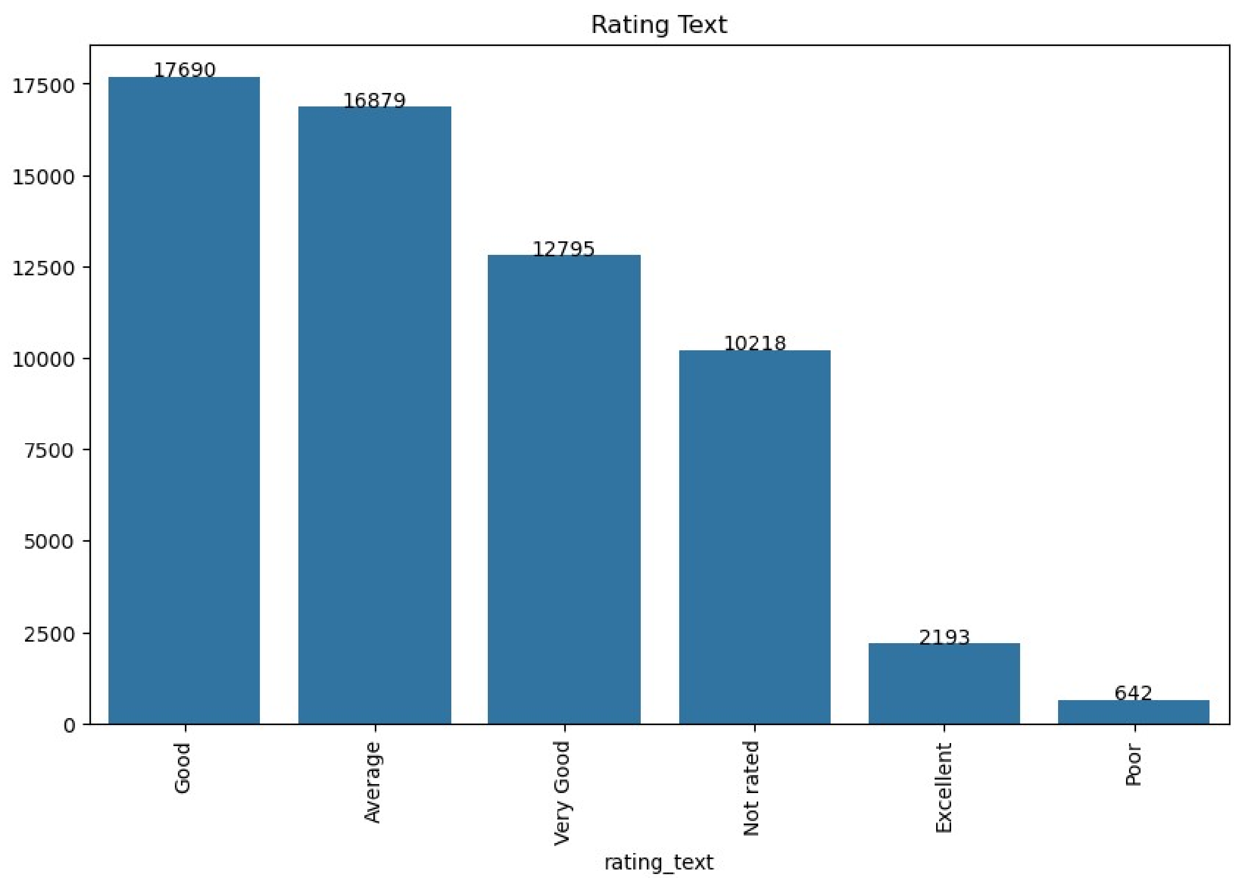
Replacing specific rating texts

```
df['rating_text']=df['rating_text'].replace({'Çok iyi' : 'Good',
'Sangat Baik' : 'Average', 'Muito Bom' : 'Very Good',
                                             'Excelente' :
'Excellent', 'Muy Bueno' : 'Excellent' ,'Excelente' : 'Excellent',
'Muy Bueno' : 'Poor',
                                             'Bardzo dobrze' : 'Good',
'Bom' : 'Average' , 'Baik': 'Excellent', 'Skvělé' : 'Not rated','Velmi
dobré' : 'Not rated',
                                             'Buono' : 'Excellent',
'Dobrze' : 'Poor', 'Wybitnie' : 'Not rated', 'Eccellente' : 'Very
Good' , 'Vynikajúce' : 'Average',
                                             'Průměr' : 'Poor',
'Média' : 'Good','Promedio':'Not rated','Muito bom' :
'Excellent','Ortalama': 'Poor', 'Średnio' : 'Good',
                                             'Priemer' :
'Good','Media' : 'Average','Biasa' : 'Excellent','Scarso':
'Poor','İyi' : 'Excellent', 'Harika' : 'Very Good',
                                             'Ottimo' :
'Average','Veľmi dobré': 'Excellent','Terbaik' : 'Excellent', 'Skvělá
volba' : 'Good', 'Dobré' : 'Very Good',
                                             'Bueno' : 'Good'})
df['rating_text'].value_counts()
```

```
rating_text
Good          17690
Average       16879
Very Good     12795
Not rated     10218
```

```
Excellent     2193
Poor           642
Name: count, dtype: int64
```

```python
# Calculate the value counts
high = df['rating_text'].value_counts()

# Plotting the barplot
plt.figure(figsize=(10, 6))
g = sns.barplot(x=high.index, y=high.values)
plt.xticks(rotation=90)
plt.title("Rating Text")

# Adding labels on bars
for index, value in enumerate(high.values):
    plt.text(index, value + 0.01, str(value), ha='center')

plt.show()
```



Observations:

The majority of customers have rated the restaurants as "Good" or "Average".

A significant proportion of customers have not rated the restaurants.

Recommendations:

Encourage Customer Feedback: Implement strategies to encourage more customers to leave ratings and reviews, such as offering incentives or making the rating process easier.

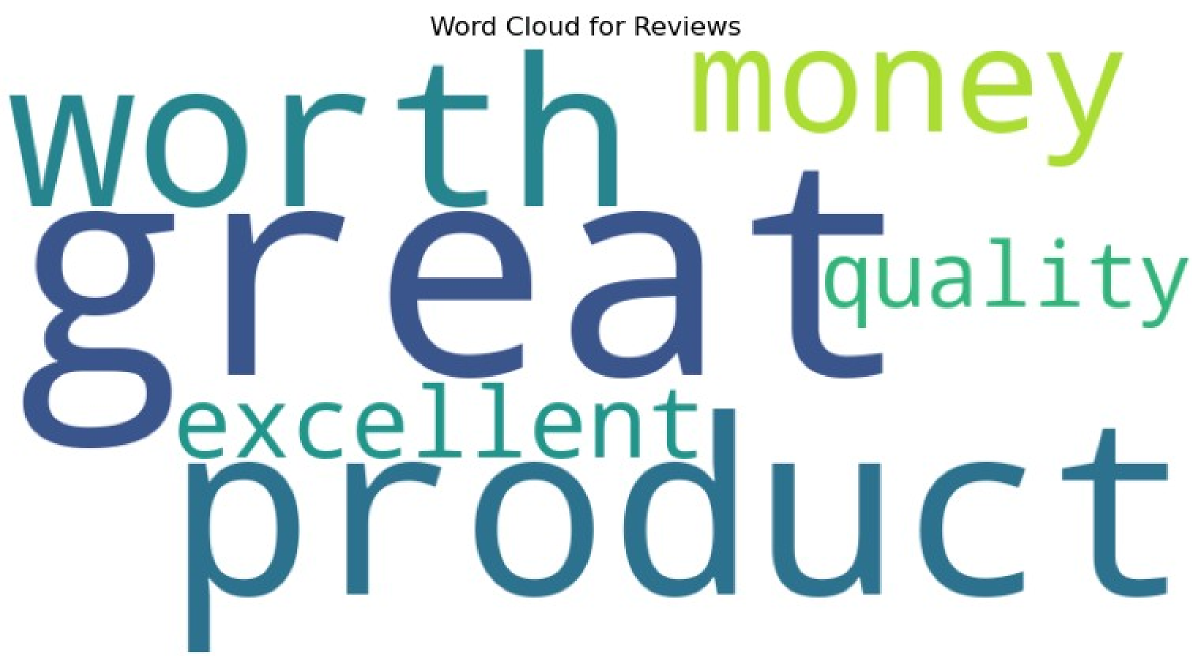
Focus on Improving "Good" Ratings: Identify areas where "Good" rated restaurants can improve to reach "Very Good" or "Excellent" ratings. This could include enhancing food quality, service, or ambiance.

Address "Poor" Ratings: Analyze the reasons for poor ratings and take corrective actions to improve customer satisfaction and prevent future negative reviews.

```
import string
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.corpus import stopwords
import nltk

# Download stopwords if not already downloaded
nltk.download('stopwords')

# Example reviews data
reviews = pd.Series(["This is a great product!", "Not worth the
money.", "Excellent quality."])

# Function to clean text
def clean_text(text):
    # Remove punctuation
    text = text.translate(str.maketrans("", "", string.punctuation))
    # Convert text to lowercase
    text = text.lower()
    # Remove stop words
    stop_words = set(stopwords.words('english'))
    text = ' '.join([word for word in text.split() if word not in
```

```
stop_words])
    return text

# Clean the reviews text
cleaned_reviews = reviews.apply(clean_text)

# Join all reviews into a single string
all_reviews = ' '.join(cleaned_reviews)

# Generate a Word Cloud
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(all_reviews)

# Plot the Word Cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud for Reviews')
plt.show()
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```



Conclusion

The analysis highlights key factors driving restaurant performance, such as high ratings, popular cuisines, and delivery services, while addressing areas for improvement, including low-rated establishments and underperforming features like outdoor seating. To enhance Zomato’s value proposition, strategies should focus on promoting high-rated and unique restaurants, improving customer feedback mechanisms, and leveraging data-driven insights for targeted marketing and operational optimization.