

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

```
In [ ]: df = pd.read_csv('zomato[1].csv')
df.head()
```

```
In [ ]: df.shape
```

```
In [ ]: df.columns
```

```
In [ ]: df = df.drop(['url', 'address', 'phone', 'menu_item'], axis = 1)
df.head()
```

```
In [ ]: df.info()
```

```
In [ ]: df.drop_duplicates(inplace = True)
df.shape
```

```
In [ ]: df['rate'].unique()
```

```
In [ ]: def handlerate(value):
    if(value=='NEW' or value=='-'):
        return np.nan
    else:
        value = str(value).split('/')
        value = value[0]
        return float(value)

df['rate'] = df['rate'].apply(handlerate)
df['rate'].head()
```

```
In [ ]: df['rate'].fillna(df['rate'].mean(), inplace = True)
df['rate'].isnull().sum()
```

```
In [ ]: df.info()
```

```
In [7]: df.dropna(inplace = True)
df.head()
```

Out[7]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines	approx_cost
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian	
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani	



```
In [8]: df.rename(columns = {'approx_cost(for two people)': 'Cost2plates', 'listed_in(type)': '1'})
df.head()
```

Out[8]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines	Cc
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian	
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani	



In [9]:

df['location'].unique()

```
Out[9]: array(['Banashankari', 'Basavanagudi', 'Jayanagar', 'Kumaraswamy Layout',
        'Rajarajeshwari Nagar', 'Mysore Road', 'Uttarahalli',
        'South Bangalore', 'Vijay Nagar', 'Bannerghatta Road', 'JP Nagar',
        'BTM', 'Wilson Garden', 'Koramangala 5th Block', 'Shanti Nagar',
        'Richmond Road', 'City Market', 'Bellandur', 'Sarjapur Road',
        'Marathahalli', 'HSR', 'Old Airport Road', 'Indiranagar',
        'Koramangala 1st Block', 'East Bangalore', 'MG Road',
        'Brigade Road', 'Lavelle Road', 'Church Street', 'Ulsoor',
        'Residency Road', 'Shivajinagar', 'Infantry Road',
        'St. Marks Road', 'Cunningham Road', 'Race Course Road', 'Domlur',
        'Koramangala 8th Block', 'Frazer Town', 'Ejipura', 'Vasanth Nagar',
        'Jeevan Bhima Nagar', 'Old Madras Road', 'Commercial Street',
        'Koramangala 6th Block', 'Majestic', 'Langford Town',
        'Koramangala 7th Block', 'Brookefield', 'Whitefield',
        'ITPL Main Road, Whitefield', 'Varthur Main Road, Whitefield',
        'Koramangala 2nd Block', 'Koramangala 3rd Block',
        'Koramangala 4th Block', 'Koramangala', 'Bommanahalli',
        'Hosur Road', 'Seshadripuram', 'Electronic City', 'Banaswadi',
        'North Bangalore', 'RT Nagar', 'Kammanahalli', 'Hennur',
        'HBR Layout', 'Kalyan Nagar', 'Thippasandra', 'CV Raman Nagar',
        'Kaggadasapura', 'Kanakapura Road', 'Nagawara', 'Rammurthy Nagar',
        'Sankey Road', 'Central Bangalore', 'Malleshwaram',
        'Sadashiv Nagar', 'Basaveshwara Nagar', 'Rajajinagar',
        'New BEL Road', 'West Bangalore', 'Yeshwantpur', 'Sanjay Nagar',
        'Sahakara Nagar', 'Jalahalli', 'Yelahanka', 'Magadi Road',
        'KR Puram'], dtype=object)
```

```
In [10]: df['listed_in(city)'].unique()
```

```
Out[10]: array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
        'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
        'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
        'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
        'Koramangala 4th Block', 'Koramangala 5th Block',
        'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
        'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
        'Old Airport Road', 'Rajajinagar', 'Residency Road',
        'Sarjapur Road', 'Whitefield'], dtype=object)
```

```
In [11]: df = df.drop(['listed_in(city)'], axis = 1)
```

```
In [12]: df['Cost2plates'].unique()
```

```
Out[12]: array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
        '750', '200', '850', '1,200', '150', '350', '250', '1,500',
        '1,300', '1,000', '100', '900', '1,100', '1,600', '950', '230',
        '1,700', '1,400', '1,350', '2,200', '2,000', '1,800', '1,900',
        '180', '330', '2,500', '2,100', '3,000', '2,800', '3,400', '40',
        '1,250', '3,500', '4,000', '2,400', '1,450', '3,200', '6,000',
        '1,050', '4,100', '2,300', '120', '2,600', '5,000', '3,700',
        '1,650', '2,700', '4,500'], dtype=object)
```

```
In [13]: def handlecomma(value):
        value = str(value)
        if ',' in value:
            value = value.replace(',', '')
            return float(value)
        else:
            return float(value)
```

```
df['Cost2plates'] = df['Cost2plates'].apply(handlecomma)
df['Cost2plates'].unique()
```

```
Out[13]: array([ 800.,  300.,  600.,  700.,  550.,  500.,  450.,  650.,  400.,
        750.,  200.,  850., 1200.,  150.,  350.,  250., 1500., 1300.,
       1000.,  100.,  900., 1100., 1600.,  950.,  230., 1700., 1400.,
       1350., 2200., 2000., 1800., 1900.,  180.,  330., 2500., 2100.,
       3000., 2800., 3400.,   40., 1250., 3500., 4000., 2400., 1450.,
       3200., 6000., 1050., 4100., 2300.,  120., 2600., 5000., 3700.,
       1650., 2700., 4500.] )
```

```
In [14]: df.head()
```

```
Out[14]:
```

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines	Cc
--	------	--------------	------------	------	-------	----------	-----------	------------	----------	----

0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese	
---	-------	-----	-----	-------	-----	--------------	---------------	---	--------------------------------	--

1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai	
---	----------------	-----	----	-------	-----	--------------	---------------	---	-----------------------------	--

2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian	
---	-----------------	-----	----	-------	-----	--------------	---------------------	---	------------------------	--

3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian	
---	-----------------------	----	----	-------	----	--------------	-------------	-------------	----------------------------	--

4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani	
---	---------------	----	----	-------	-----	--------------	---------------	---------------------	--------------------------	--

```
In [15]: rest_types = df['rest_type'].value_counts(ascending = False)
rest_types
```

```
Out[15]: Casual Dining      7382
         Quick Bites      5269
         Cafe             2379
         Dessert Parlor    1085
         Casual Dining, Bar 989
         ...
         Fine Dining, Microbrewery 5
         Club, Casual Dining 4
         Dessert Parlor, Kiosk 2
         Dhaba             1
         Food Court, Casual Dining 1
         Name: rest_type, Length: 73, dtype: int64
```

```
In [16]: rest_types_lessthan1000 = rest_types[rest_types<1000]
         rest_types_lessthan1000
```

```
Out[16]: Casual Dining, Bar      989
         Delivery              707
         Bar                  358
         Bar, Casual Dining    351
         Takeaway, Delivery    344
         ...
         Fine Dining, Microbrewery 5
         Club, Casual Dining 4
         Dessert Parlor, Kiosk 2
         Dhaba             1
         Food Court, Casual Dining 1
         Name: rest_type, Length: 69, dtype: int64
```

```
In [17]: def handle_rest_type(value):
         if(value in rest_types_lessthan1000):
             return 'others'
         else:
             return value

         df['rest_type'] = df['rest_type'].apply(handle_rest_type)
         df['rest_type'].value_counts()
```

```
Out[17]: Casual Dining      7382
         others            7291
         Quick Bites      5269
         Cafe             2379
         Dessert Parlor    1085
         Name: rest_type, dtype: int64
```

```
In [18]: location = df['location'].value_counts(ascending = False)

         location_lessthan300 = location[location<300]

         def handle_location(value):
             if(value in location_lessthan300):
                 return 'others'
             else:
                 return value

         df['location'] = df['location'].apply(handle_location)
         df['location'].value_counts()
```

```
Out[18]:
```

others	3735
Koramangala 5th Block	1799
BTM	1484
Indiranagar	1371
HSR	1177
Jayanagar	1067
JP Nagar	1020
Whitefield	835
Koramangala 7th Block	744
Koramangala 6th Block	728
Marathahalli	683
Koramangala 4th Block	664
MG Road	602
Brigade Road	570
Church Street	512
Bannerghatta Road	498
Ulsoor	477
Bellandur	474
Kalyan Nagar	468
Koramangala 1st Block	466
Sarjapur Road	457
Lavelle Road	446
Residency Road	442
Banashankari	387
Malleshwaram	372
Richmond Road	351
Cunningham Road	333
Electronic City	327
Brookefield	307
New BEL Road	306
St. Marks Road	304

Name: location, dtype: int64

```
In [19]: cuisines = df['cuisines'].value_counts(ascending = False)

cuisines_lessthan100 = cuisines[cuisines<100]

def handle_cuisines(value):
    if(value in cuisines_lessthan100):
        return 'others'
    else:
        return value

df['cuisines'] = df['cuisines'].apply(handle_cuisines)
df['cuisines'].value_counts()
```

Out[19]:

others	17671
North Indian	1160
North Indian, Chinese	779
South Indian	366
Cafe	285
South Indian, North Indian, Chinese	233
Desserts, Beverages	216
Bakery, Desserts	216
Ice Cream, Desserts	212
Chinese	210
Desserts	209
Biryani	169
Chinese, Momos	162
Finger Food	151
Fast Food	150
North Indian, Chinese, Biryani	140
Mithai, Street Food	138
North Indian, Mughlai	134
Burger, Fast Food	132
Cafe, Continental	125
Desserts, Ice Cream	122
Chinese, Thai	117
Pizza, Fast Food	106
South Indian, North Indian, Chinese, Street Food	103
Cafe, Desserts	100
Name: cuisines, dtype: int64	

In [20]:

df.head()

Out[20]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines	Cost
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	others	
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	others	
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	others	Churros, Cannelloni, Minestrone Soup, Hot Choc...	others	
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	Masala Dosa	others	
4	Grand Village	No	No	3.8/5	166	others	Casual Dining	Panipuri, Gol Gappe	others	



In []:

```
def bar_chart(feature):
    online_order = df[df['online_order']==1][feature].value_counts()
    df.plot(kind='bar',stacked=True, figsize=(15,7))

bar_chart('online_order')
```

In [22]:

```
df.columns
```

Out[22]:

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'Cost2plates', 'reviews_list',
       'Type'],
      dtype='object')
```

In [23]:

```
df.isnull().sum()
```

```
Out[23]: name          0
online_order  0
book_table    0
rate          0
votes         0
location      0
rest_type     0
dish_liked    0
cuisines      0
Cost2plates   0
reviews_list  0
Type          0
dtype: int64
```

```
In [ ]: bar_chart('online_order')
```

```
In [ ]: bar_chart('location')
```

```
In [ ]: group = df.groupby(['location', 'rate'])
location_rate = group.size().unstack()

sns.heatmap(location_rate, annot = True, fmt = "d")
```

```
In [ ]: sns.violinplot(x = "rest_type", y = "rate", hue = "Cost2plates",
data = df, split = True)
```

```
In [ ]: df1 = df.groupby(['location', 'online_order'])['name'].count()
df1.to_csv('zomato[1].csv')
df1 = pd.read_csv('zomato[1].csv')
df1 = pd.pivot_table(df1, values=None, index=['location'], columns=['online_order'], fill_valu
df1
```

```
In [ ]: df1.plot(kind = 'bar', figsize = (15,8))
```

```
In [ ]: df2 = df.groupby(['location', 'book_table'])['name'].count()
df2.to_csv('zomato[1].csv')
df2 = pd.read_csv('zomato[1].csv')
df2 = pd.pivot_table(df2, values=None, index=['location'], columns=['book_table'], fill_valu
df2
```

```
In [ ]: df2.plot(kind = 'bar', figsize = (15,8))
```

```
In [ ]: plt.figure(figsize = (14, 8))
sns.boxplot(x = 'Type', y = 'rate', data = df, palette = 'inferno')
```

```
In [ ]: df3 = df.groupby(['location', 'Type'])['name'].count()
df3.to_csv('zomato[1].csv')
df3 = pd.read_csv('zomato[1].csv')
df3 = pd.pivot_table(df3, values=None, index=['location'], columns=['Type'], fill_valu
df3
```

```
In [ ]: df3.plot(kind = 'bar', figsize = (36,8))
```

```
In [ ]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.metrics import mean_squared_error
import xgboost as xgb
```

```
In [ ]: data = pd.read_csv('zomato[1].csv')
```

```
In [ ]: data = pd.get_dummies(data)
```

```
In [ ]: X = data.drop('rate', axis=1)
y = data['rate']
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

```
In [ ]: df.head()
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)
```

```
In [ ]: xgb_model = xgb.XGBRegressor(random_state=42)
xgb_model.fit(X_train, y_train)
```

```
In [ ]: rf_predictions = rf_model.predict(X_test)
rf_rmse = mean_squared_error(y_test, rf_predictions, squared=False)
print("Random Forest RMSE:", rf_rmse)
```

```
In [ ]: xgb_predictions = xgb_model.predict(X_test)
xgb_rmse = mean_squared_error(y_test, xgb_predictions, squared=False)
print("XGBoost RMSE:", xgb_rmse)
```

```
In [ ]:
```