

Level_1_tasks

May 31, 2024

COGNIFYZ TECHNOLOGIES

Task Level - 1

```
[1]: # Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

0.0.1 Loading Dataset

```
[2]: df = pd.read_csv('cognifyz_dataset.csv')
```

0.1 Basic Data Exploration

```
[3]: df.shape # 9551 rows and 21 columns
      ↪ columns
```

```
[3]: (9551, 21)
```

```
[4]: df.head()
```

```
[4]: Restaurant ID      Restaurant Name  Country Code      City \
0      6317637      Le Petit Souffle      162      Makati City
1      6304287      Izakaya Kikufuji      162      Makati City
2      6300002      Heat - Edsa Shangri-La      162      Mandaluyong City
3      6318506      Ooma      162      Mandaluyong City
4      6314302      Sambo Kojin      162      Mandaluyong City
```

```
Address \
0 Third Floor, Century City Mall, Kalayaan Avenu...
1 Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
2 Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...
3 Third Floor, Mega Fashion Hall, SM Megamall, O...
4 Third Floor, Mega Atrium, SM Megamall, Ortigas...
```

```
Locality \
0 Century City Mall, Poblacion, Makati City
```

```

1 Little Tokyo, Legaspi Village, Makati City
2 Edsa Shangri-La, Ortigas, Mandaluyong City
3 SM Megamall, Ortigas, Mandaluyong City
4 SM Megamall, Ortigas, Mandaluyong City

```

```

                Locality Verbose Longitude Latitude \
0 Century City Mall, Poblacion, Makati City, Mak... 121.027535 14.565443
1 Little Tokyo, Legaspi Village, Makati City, Ma... 121.014101 14.553708
2 Edsa Shangri-La, Ortigas, Mandaluyong City, Ma... 121.056831 14.581404
3 SM Megamall, Ortigas, Mandaluyong City, Mandal... 121.056475 14.585318
4 SM Megamall, Ortigas, Mandaluyong City, Mandal... 121.057508 14.584450

```

```

                Cuisines ... Currency Has Table booking \
0 French, Japanese, Desserts ... Botswana Pula(P) Yes
1 Japanese ... Botswana Pula(P) Yes
2 Seafood, Asian, Filipino, Indian ... Botswana Pula(P) Yes
3 Japanese, Sushi ... Botswana Pula(P) No
4 Japanese, Korean ... Botswana Pula(P) Yes

```

```

Has Online delivery Is delivering now Switch to order menu Price range \
0 No No No 3
1 No No No 3
2 No No No 4
3 No No No 4
4 No No No 4

```

```

Aggregate rating Rating color Rating text Votes
0 4.8 Dark Green Excellent 314
1 4.5 Dark Green Excellent 591
2 4.4 Green Very Good 270
3 4.9 Dark Green Excellent 365
4 4.8 Dark Green Excellent 229

```

[5 rows x 21 columns]

```
[5]: df.tail()
```

```

[5]: Restaurant ID      Restaurant Name Country Code      City \
9546      5915730      Naml Gurme      208      stanbul
9547      5908749      Ceviz A ac      208      stanbul
9548      5915807      Huqqa      208      stanbul
9549      5916112      A k Kahve      208      stanbul
9550      5927402      Walter's Coffee Roastery      208      stanbul

```

```

                Address      Locality \
9546      Kemanke      Karamustafa Pa a Mahallesi, R ht m ...      Karak_y
9547      Ko uyolu Mahallesi, Muhittin st_nda Cadd...      Ko uyolu

```

```

9548 Kuru_e me Mahallesi, Muallim Naci Caddesi, N... Kuru_e me
9549 Kuru_e me Mahallesi, Muallim Naci Caddesi, N... Kuru_e me
9550 Cafea a Mahallesi, Bademalt Sokak, No 21/B, ... Moda

```

```

      Locality Verbose Longitude Latitude \
9546 Karak_y, stanbul 28.977392 41.022793
9547 Ko uyolu, stanbul 29.041297 41.009847
9548 Kuru_e me, stanbul 29.034640 41.055817
9549 Kuru_e me, stanbul 29.036019 41.057979
9550 Moda, stanbul 29.026016 40.984776

```

```

      Cuisines ... Currency \
9546 Turkish ... Turkish Lira(TL)
9547 World Cuisine, Patisserie, Cafe ... Turkish Lira(TL)
9548 Italian, World Cuisine ... Turkish Lira(TL)
9549 Restaurant Cafe ... Turkish Lira(TL)
9550 Cafe ... Turkish Lira(TL)

```

```

Has Table booking Has Online delivery Is delivering now \
9546 No No No
9547 No No No
9548 No No No
9549 No No No
9550 No No No

```

```

Switch to order menu Price range Aggregate rating Rating color \
9546 No 3 4.1 Green
9547 No 3 4.2 Green
9548 No 4 3.7 Yellow
9549 No 4 4.0 Green
9550 No 2 4.0 Green

```

```

Rating text Votes
9546 Very Good 788
9547 Very Good 1034
9548 Good 661
9549 Very Good 901
9550 Very Good 591

```

[5 rows x 21 columns]

```

[6]: # Columns
df.columns

```

```

[6]: Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
          'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
          'Average Cost for two', 'Currency', 'Has Table booking',

```

```
'Has Online delivery', 'Is delivering now', 'Switch to order menu',
'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
'Votes'],
dtype='object')
```

```
[7]: # Data types
df.dtypes
```

```
[7]: Restaurant ID          int64
Restaurant Name          object
Country Code            int64
City                    object
Address                 object
Locality                object
Locality Verbose        object
Longitude               float64
Latitude               float64
Cuisines                 object
Average Cost for two     int64
Currency                object
Has Table booking       object
Has Online delivery     object
Is delivering now       object
Switch to order menu    object
Price range             int64
Aggregate rating        float64
Rating color            object
Rating text            object
Votes                  int64
dtype: object
```

0.2 Data Cleaning

```
[8]: # Finding Null values
df.isnull().sum()                                ## No null values
```

```
[8]: Restaurant ID          0
Restaurant Name          0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude               0
Cuisines                 9
Average Cost for two     0
```

```

Currency          0
Has Table booking  0
Has Online delivery 0
Is delivering now  0
Switch to order menu 0
Price range       0
Aggregate rating   0
Rating color       0
Rating text        0
Votes             0
dtype: int64

```

```

[9]: # Finding Duplicates
df.duplicated().sum()                ## No duplicates
↪present

```

```

[9]: 0

```

```

[10]: # Cleaning Cuisines column for analysis
df['Cuisines'] = df['Cuisines'].str.split(',').str[0]

```

0.3 Data Analysis

0.3.1 Task-1: Top Cuisines

- Determine the top three most common cuisines in the dataset.
- Calculate the percentage of restaurants that serve each of the top cuisines.

```

[11]: # Determine the top three most common cuisines in the dataset.
cuisines_df = df.Cuisines.value_counts().head(3)
print('Top three most common Cuisines in the dataset are: \n', cuisines_df)

```

```

Top three most common Cuisines in the dataset are:
Cuisines
North Indian    2992
Chinese         855
Fast Food       672
Name: count, dtype: int64

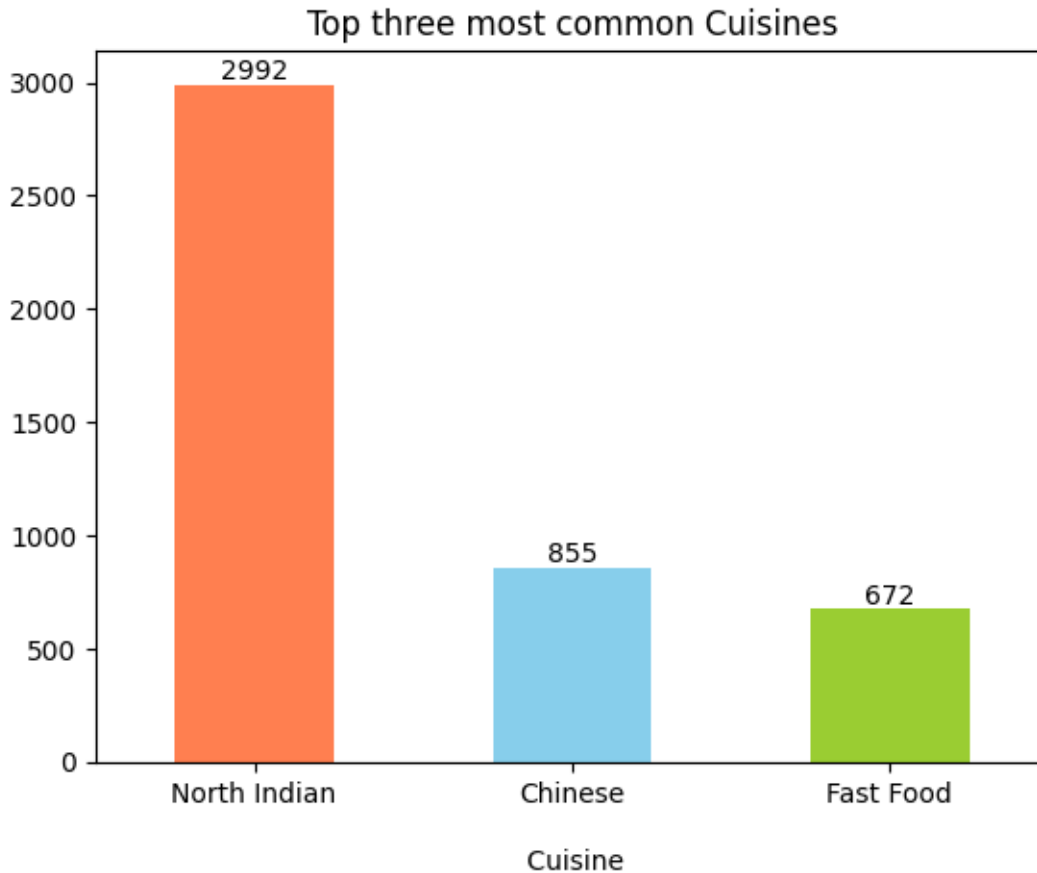
```

```

[12]: ax = cuisines_df.plot(kind='bar', color=['coral', 'skyblue', 'yellowgreen'])
plt.title('Top three most common Cuisines')
plt.xticks(rotation=0)
plt.xlabel('\n Cuisine');

# Data labels
for i in ax.containers:
    ax.bar_label(i, label_type='edge')

```



North Indian, Chinese, and Fast Food are the most common Cuisines in this dataset

```
[13]: # Calculate the percentage of restaurants that serve each of the top cuisines.
total_restaurants = len(df)
cuisines_by_restaurants = round((cuisines_df/total_restaurants)*100,2).head(3)
print('Percentage of restaurants that serve each of the top cuisines:\n' ,
      ↪cuisines_by_restaurants)
```

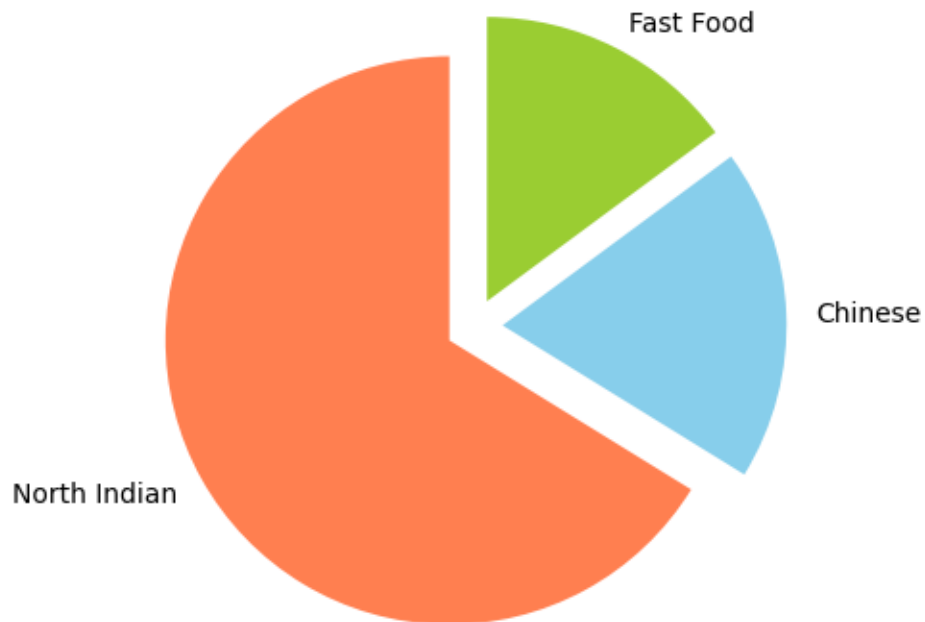
Percentage of restaurants that serve each of the top cuisines:

```
Cuisines
North Indian    31.33
Chinese         8.95
Fast Food       7.04
Name: count, dtype: float64
```

```
[14]: ax = cuisines_by_restaurants.plot(kind='pie', colors=['coral', 'skyblue',
      ↪'yellowgreen'],
                                         explode=(0.1, 0.1, 0.1), startangle=90,
                                         radius=1)
plt.title('Percentage of restaurants that serve each of the top cuisines')
```

```
plt.ylabel('');
```

Percentage of restaurants that serve each of the top cuisines



The percentage of restaurants serving North Indian is approximately 31% followed by Chinese and Fast food at 9% and 7% respectively.

0.3.2 Task-2: City Analysis

- Identify the city with the highest number of restaurants in the dataset.
- Calculate the average rating for restaurants in each city.
- Determine the city with the highest average rating.

```
[15]: # Identify the city with the highest number of restaurants in the dataset.  
city_with_restaurants = df.City.value_counts()  
city_with_restaurants
```

```
[15]: City  
New Delhi      5473  
Gurgaon        1118  
Noida          1080  
Faridabad       251  
Ghaziabad       25  
...  
Panchkula       1
```

```

Mc Millan          1
Mayfield           1
Macedon            1
Vineland Station   1
Name: count, Length: 141, dtype: int64

```

```

[16]: max_restaurants_city = city_with_restaurants.idxmax()
max_restaurants_counts = city_with_restaurants.max()
print(f'The city which has maximum number of restaurants is_
↳{max_restaurants_city} with {max_restaurants_counts} restaurants')

```

The city which has maximum number of restaurants is New Delhi with 5473 restaurants

```

[53]: # Calculate the average rating for restaurants in each city.
avg_rating = round(df.groupby('City')['Aggregate rating'].mean().
↳sort_values(ascending=False), 1)
avg_rating

```

```

[53]: City
Inner City          4.9
Quezon City         4.8
Makati City         4.6
Pasig City          4.6
Mandaluyong City    4.6
...
New Delhi           2.4
Montville           2.4
Mc Millan           2.4
Noida               2.0
Faridabad           1.9
Name: Aggregate rating, Length: 141, dtype: float64

```

```

[42]: top_city_rating = avg_rating.head(10).sort_values(ascending=True)

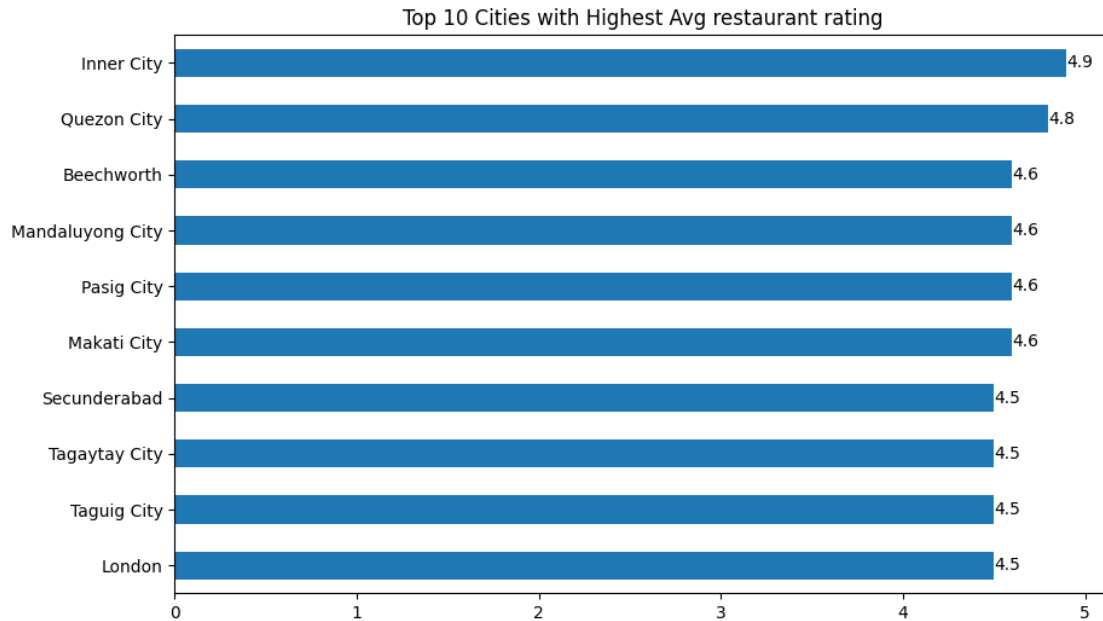
```

```

[45]: plt.figure(figsize=(10,6))
ax= top_city_rating.plot(kind='barh')
plt.title('Top 10 Cities with Highest Avg restaurant rating')
plt.xlabel('')
plt.ylabel('');

for i in ax.containers:
    ax.bar_label(i, label_type='edge')

```

```
[48]: # Determine the city with the highest average rating.
rating_top_city = top_city_rating.idxmax()
highest_rating = top_city_rating.max()
print(f"The city with the highest avg rating is {rating_top_city} with
      ↳{highest_rating} rating")
```

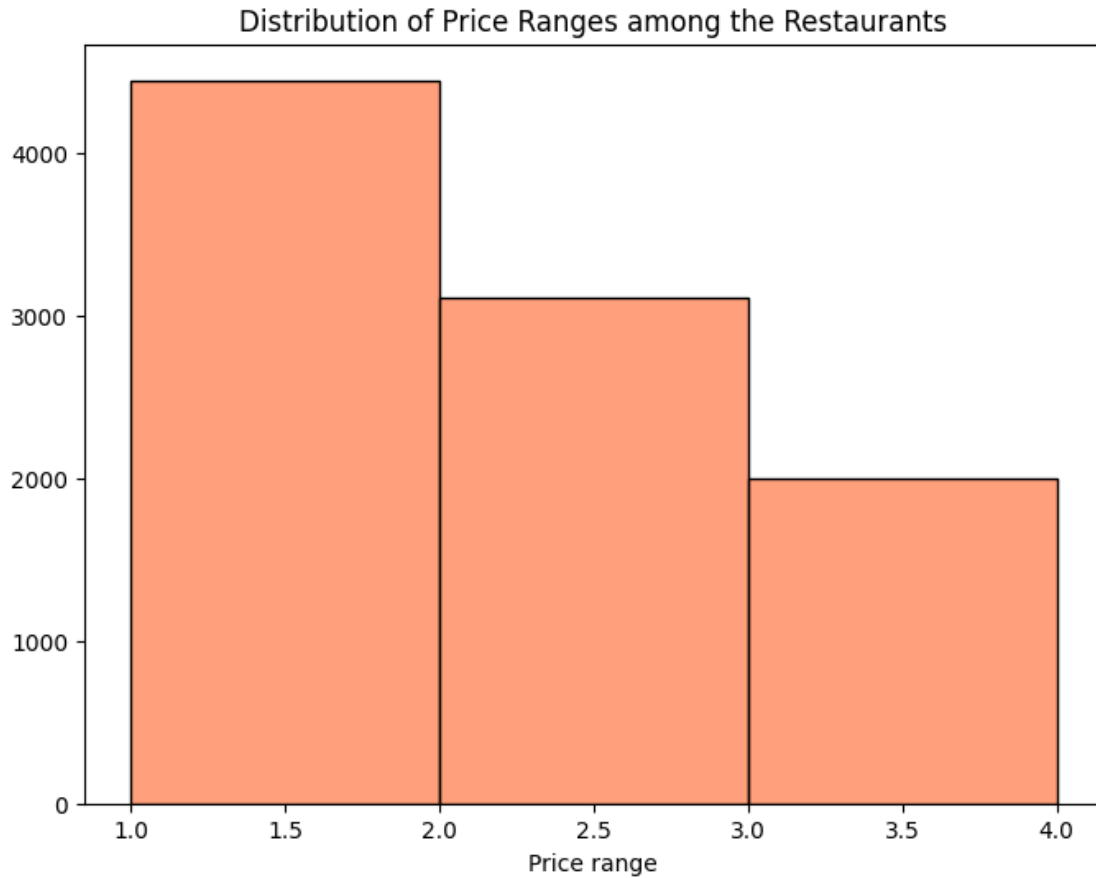
The city with the highest avg rating is Inner City with 4.9 rating

0.3.3 Task-3: Price Range Distribution

- Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants.
- Calculate the percentage of restaurants in each price range category.

The restaurants offering online delivery options have an average rating of 3.2, while those without online delivery have an average rating of 2.5.

```
[112]: # Create a histogram or bar chart to visualize the distribution of price ranges
      ↳among the restaurants.
plt.figure(figsize=(8,6))
sns.histplot(x='Price range',
             data=df,
             bins=3,
             color='coral')
plt.ylabel('')
plt.title('Distribution of Price Ranges among the Restaurants');
```

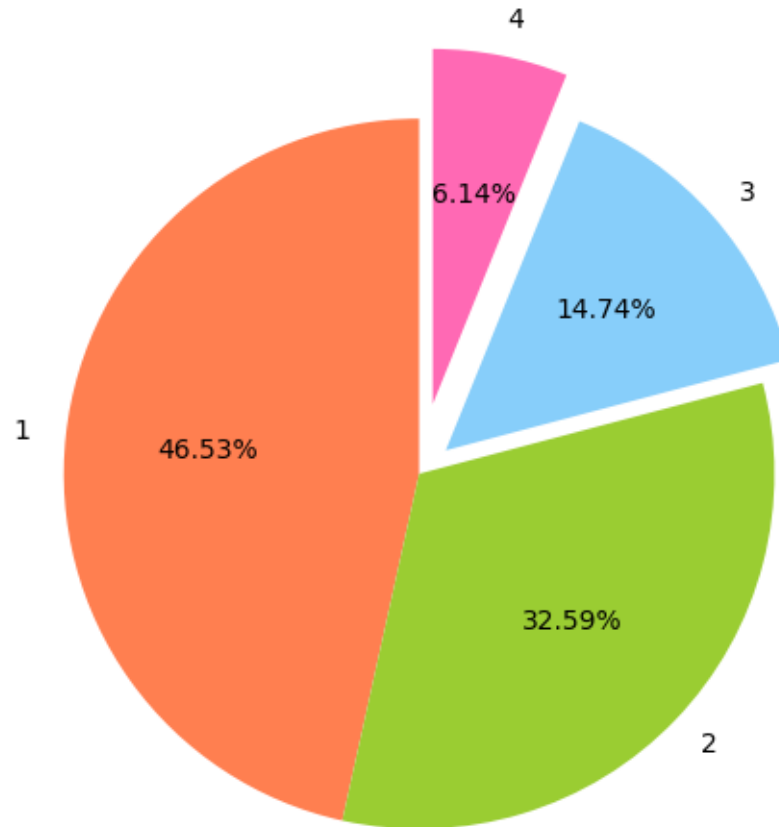


```
[115]: # Calculate the percentage of restaurants in each price range category.
restaurants_price_range = round(df['Price range'].
    ↪value_counts(normalize=True)*100, 2)
restaurants_price_range
```

```
[115]: Price range
1      46.53
2      32.59
3      14.74
4       6.14
Name: proportion, dtype: float64
```

```
[133]: plt.figure(figsize=(10, 6))
plt.pie(restaurants_price_range, labels=restaurants_price_range.index,
    autopct='%1.2f%%', explode=(0, 0, 0.1, 0.2),
    colors=['coral', 'yellowgreen', 'lightskyblue', 'hotpink'],
    ↪startangle=90)
plt.ylabel('')
plt.title('Percentage of Restaurants in each Price Range category \n');
```

Percentage of Restaurants in each Price Range category



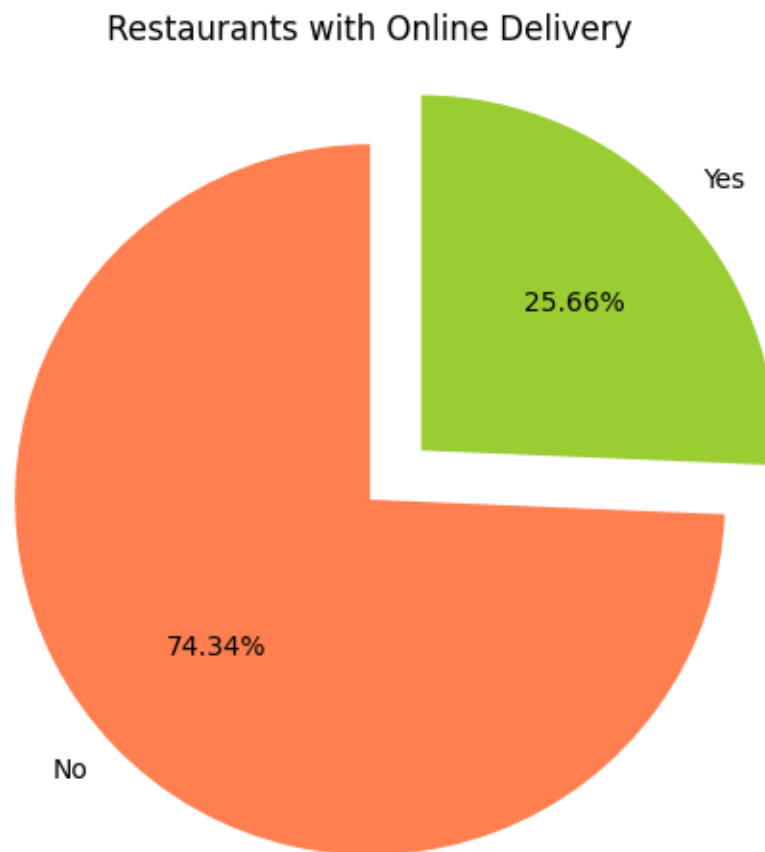
0.3.4 Task-4: Online Delivery

- Determine the percentage of restaurants that offer online delivery.
- Compare the average ratings of restaurants with and without online delivery.

```
[71]: # Determine the percentage of restaurants that offer online delivery.
restaurants_od = round(df['Has Online delivery'].
    ↪value_counts(normalize=True)*100, 2)
print(restaurants_od)
print('25.66% of Restaurants offers Online Delivery')
```

```
Has Online delivery
No      74.34
Yes     25.66
Name: proportion, dtype: float64
25.66% of Restaurants offers Online Delivery
```

```
[79]: # Visualization
plt.figure(figsize=(10, 6))
restaurants_od.plot(kind='pie', autopct='%1.2f%%', labels=restaurants_od.index,
                    colors=['coral', 'yellowgreen'], explode=(0, 0.2),
                    radius=1, startangle=90)
plt.title('Restaurants with Online Delivery')
plt.ylabel('');
```



```
[86]: # Compare the average ratings of restaurants with and without online delivery.
avg_ratings_by_delivery = round(df.groupby('Has Online delivery')['Aggregate_
    rating'].mean(), 1)
avg_ratings_by_delivery
```

```
[86]: Has Online delivery
No      2.5
Yes     3.2
Name: Aggregate rating, dtype: float64
```

```
[91]: ax = avg_ratings_by_delivery.plot(kind='bar', color=['coral', 'yellowgreen'])
plt.title('Average ratings of restaurants with and without Online delivery')
plt.xticks(rotation=0)

for i in ax.containers:
    ax.bar_label(i, label_type='edge')
```

