Part_A_Jupyter

September 28, 2024

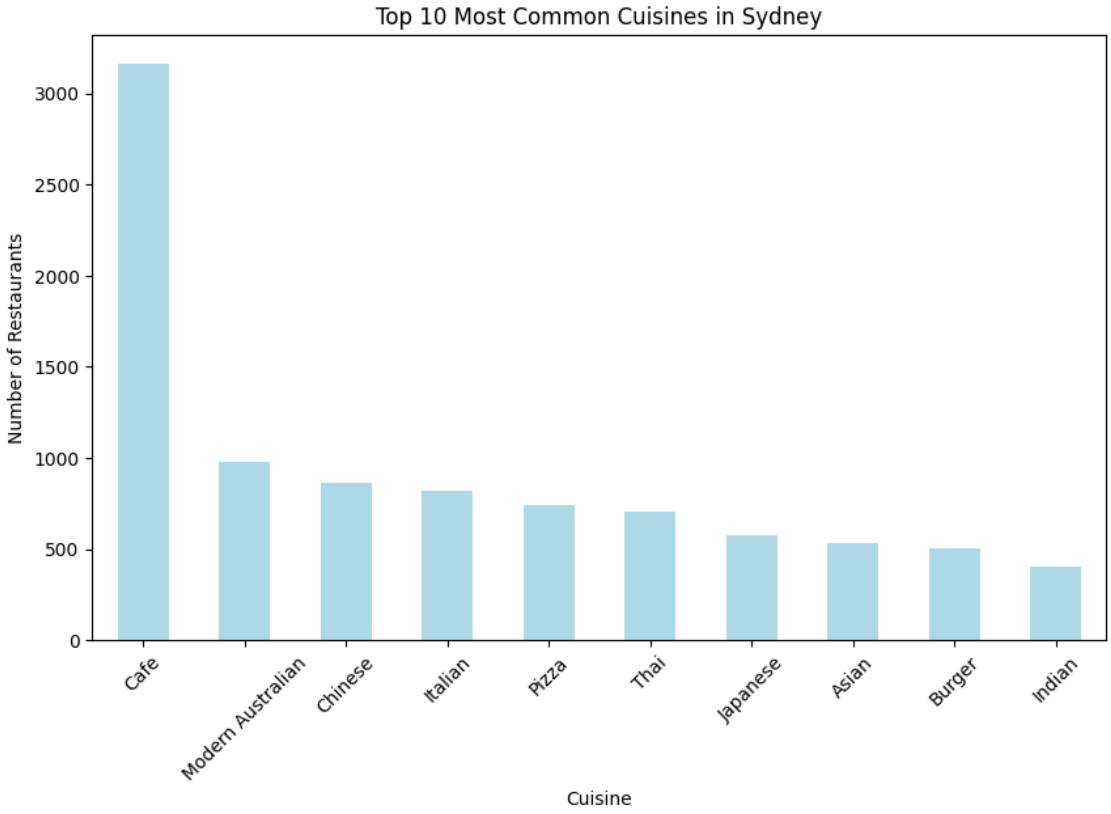
0.0.1 Part A - Data Exploration and Visualization

In this section, we will explore the Zomato dataset and conduct data analysis using various visualizations and feature engineering techniques. The main tasks include providing answers to exploratory questions, performing data cleaning, and creating interactive visualizations.

1.1 How many unique cuisines are served by Sydney restaurants?

[1]:
```
import pandas as pd
import geopandas as gpd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

# Load the Zomato dataset
zomato_df = pd.read_csv('zomato_df_final_data.csv')

# Load the Sydney GeoJSON data
sydney_geo = gpd.read_file('sydney.geojson')

# Convert 'cuisine' column from a string representation to an actual list
zomato_df.loc[:, 'cuisine'] = zomato_df['cuisine'].apply(lambda x: eval(x))

# Explode the list of cuisines into individual entries
exploded_cuisines = zomato_df.explode('cuisine')

# Count unique cuisines
unique_cuisines_count = exploded_cuisines['cuisine'].nunique()
print(f"Number of unique cuisines: {unique_cuisines_count}")

# Plot the top 10 most common cuisines
top_cuisines = exploded_cuisines['cuisine'].value_counts().head(10)
plt.figure(figsize=(10, 6))
top_cuisines.plot(kind='bar', color='lightblue')
plt.title("Top 10 Most Common Cuisines in Sydney")
plt.xlabel("Cuisine")
plt.ylabel("Number of Restaurants")
plt.xticks(rotation=45)
plt.show()
```

```
Number of unique cuisines: 134
```



Answer 1.1: There are 134 unique cuisines served by Sydney restaurants. The top 10 most common cuisines are led by 'Cafe', 'Modern Australian', 'Chinese', 'Italian', 'Pizza', 'Thai', 'Japanese', 'Asian', 'Burger', and 'Indian'.

1.2 Which suburbs (top 3) have the highest number of restaurants?

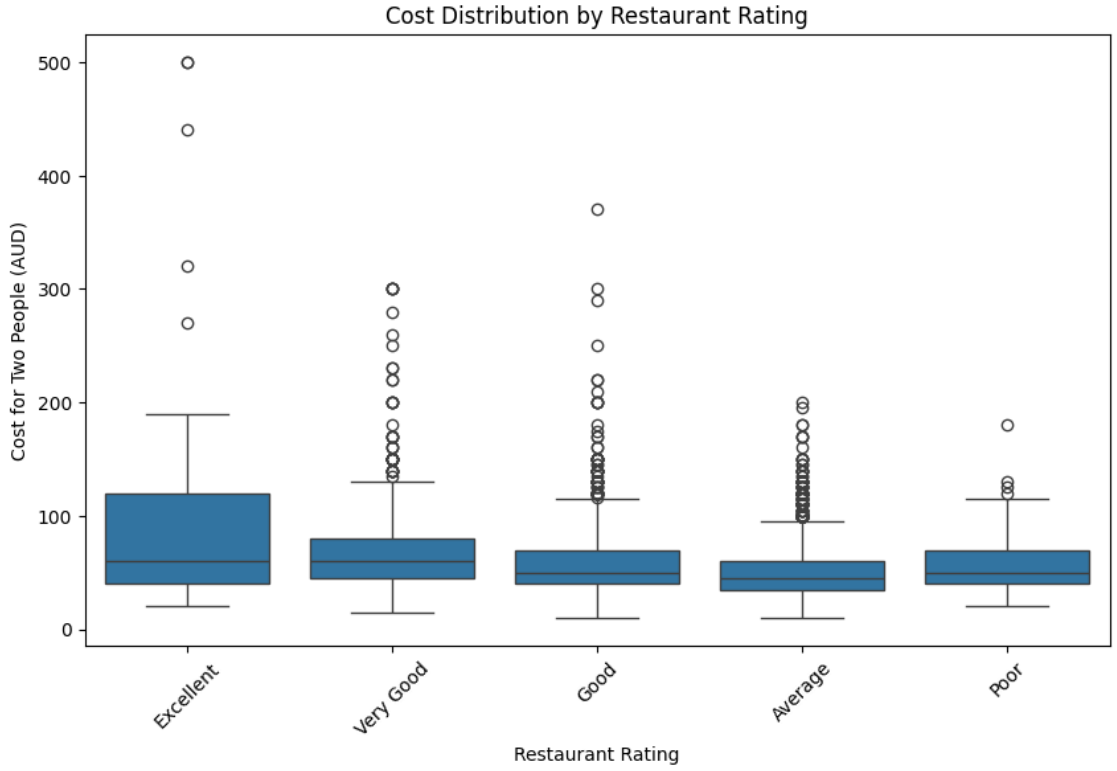
[2]:
```python
# Count the number of restaurants in each suburb (subzone)
top_suburbs = zomato_df['subzone'].value_counts().head(3)
print("Top 3 suburbs with the most restaurants:")
print(top_suburbs)
```

```
Top 3 suburbs with the most restaurants:
subzone
CBD            476
Surry Hills    260
Parramatta     225
Name: count, dtype: int64
```

Answer 1.2: The top 3 suburbs with the highest number of restaurants are: 1. CBD: 476 restaurants 2. Surry Hills: 260 restaurants 3. Parramatta: 225 restaurants

1.3 Analyze the relationship between cost and rating

[3]:
```
# Define the custom order for the ratings
rating_order = ["Excellent", "Very Good", "Good", "Average", "Poor"]

# Convert 'rating_text' to a categorical type with the specific order
zomato_df.loc[:, 'rating_text'] = pd.Categorical(zomato_df['rating_text'],␣
 ↪categories=rating_order, ordered=True)

# Plot a boxplot to analyze the relationship between cost and rating
plt.figure(figsize=(10, 6))
sns.boxplot(data=zomato_df, x='rating_text', y='cost', order=rating_order)
plt.title("Cost Distribution by Restaurant Rating")
plt.xlabel("Restaurant Rating")
plt.ylabel("Cost for Two People (AUD)")
plt.xticks(rotation=45)
plt.show()
```



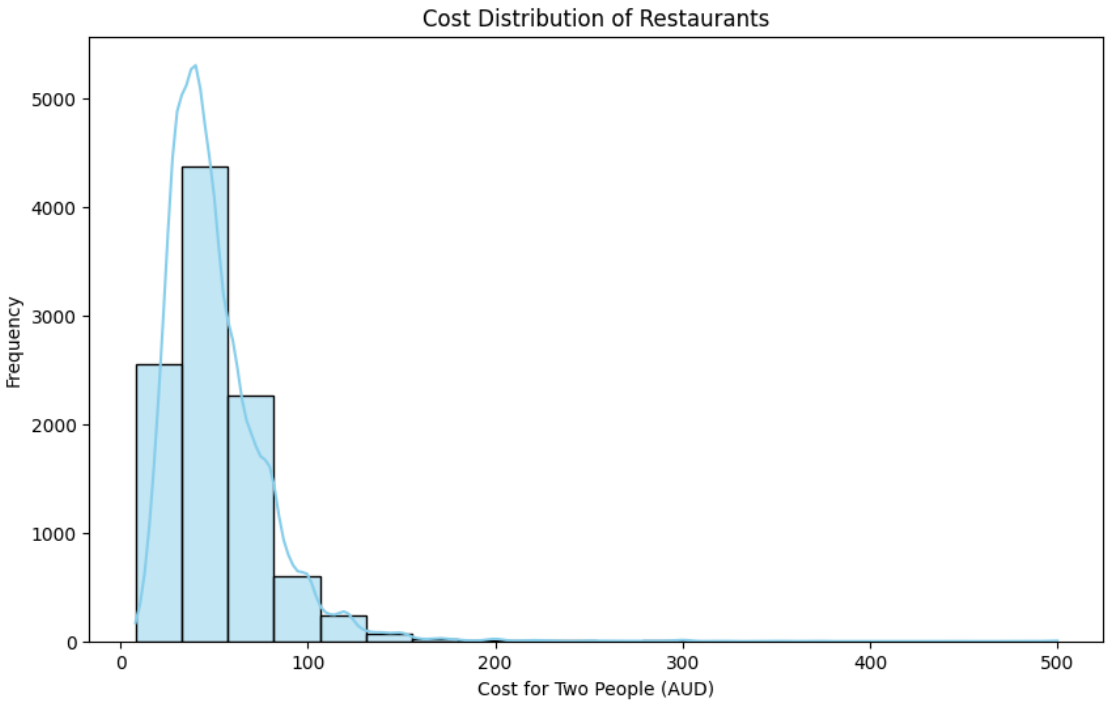
Answer 1.3: Restaurants with Excellent ratings are more costly, while those with Poor ratings are less expensive. The boxplot shows a clear trend where higher-rated restaurants have a wider and higher cost range compared to lower-rated restaurants.

0.0.2 2 Exploratory Data Analysis

We will now perform exploratory analysis for the following variables: Cost, Rating, and Type.

2.1 Exploratory analysis for 'Cost'

[4]:
```
plt.figure(figsize=(10, 6))
sns.histplot(zomato_df['cost'], bins=20, kde=True, color='skyblue')
plt.title("Cost Distribution of Restaurants")
plt.xlabel("Cost for Two People (AUD)")
plt.ylabel("Frequency")
plt.show()
```



Answer 2.1 (Cost): The cost distribution is skewed to the right, with most restaurants having a cost between 30-70 AUD for two people. Higher-cost restaurants are less common.

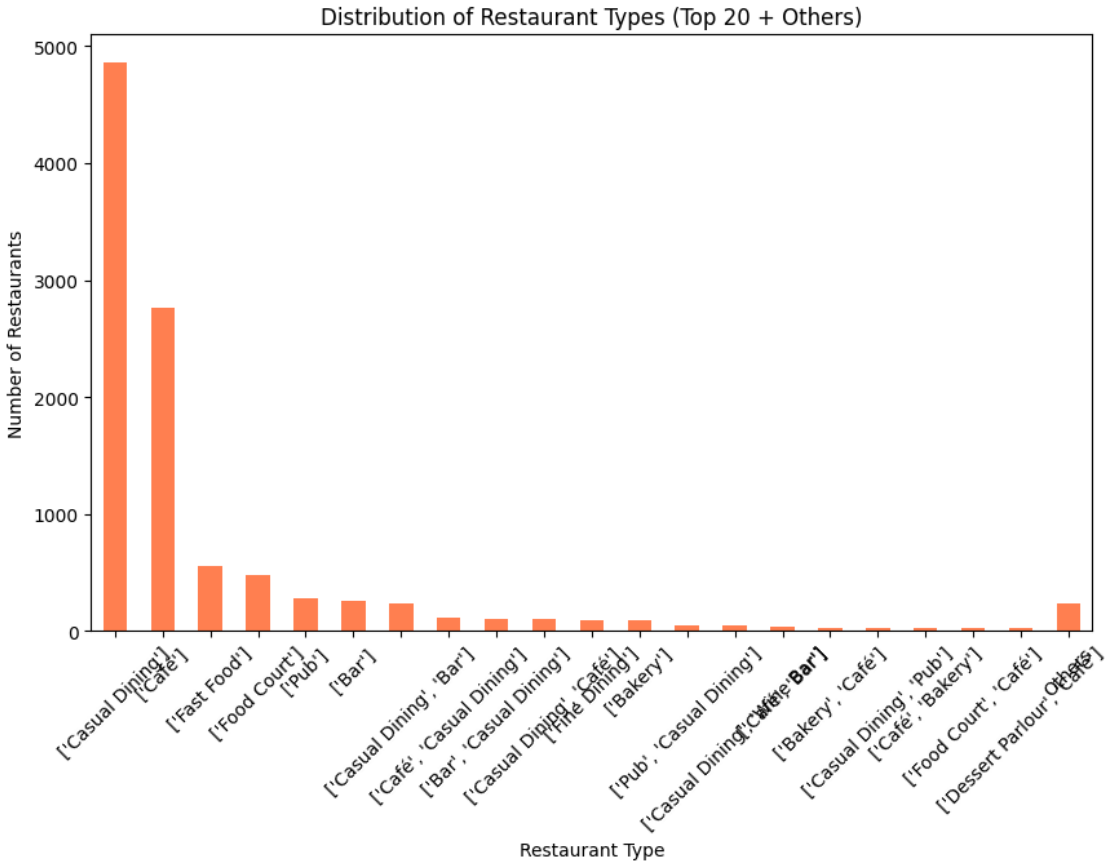
2.2 Exploratory analysis for 'Rating'

[5]:
```
plt.figure(figsize=(10, 6))
sns.histplot(zomato_df['rating_number'], bins=10, kde=True, color='lightgreen')
plt.title("Rating Distribution of Restaurants")
plt.xlabel("Rating (Out of 5)")
plt.ylabel("Frequency")
plt.show()
```

Answer 2.2 (Rating): Most restaurants have ratings around 3 to 4, with very few restaurants rated below 2 or above 4.5. This indicates that most Sydney restaurants are mid-to-highly rated.

2.3 Exploratory analysis for 'Type'

[6]:
```
# Get the count of each restaurant type
restaurant_type_counts = zomato_df['type'].value_counts()

# Keep the top 20 restaurant types, and sum the rest into 'Others'
top_20_types = restaurant_type_counts.head(20)
others_count = restaurant_type_counts[20:].sum()

# Add the 'Others' category
top_20_types['Others'] = others_count

# Plot the modified bar chart for the top 20 types + 'Others'
plt.figure(figsize=(10, 6))
top_20_types.plot(kind='bar', color='coral')
plt.title("Distribution of Restaurant Types (Top 20 + Others)")
plt.xlabel("Restaurant Type")
plt.ylabel("Number of Restaurants")
plt.xticks(rotation=45)
plt.show()
```



Answer 2.3 (Type): 'Casual Dining' and 'Café' are the most common types of restaurants in Sydney, followed by 'Fast Food' and 'Food Court'. Fine dining is less common compared to casual dining establishments.

0.0.3 Part A - Interactive Visualizations

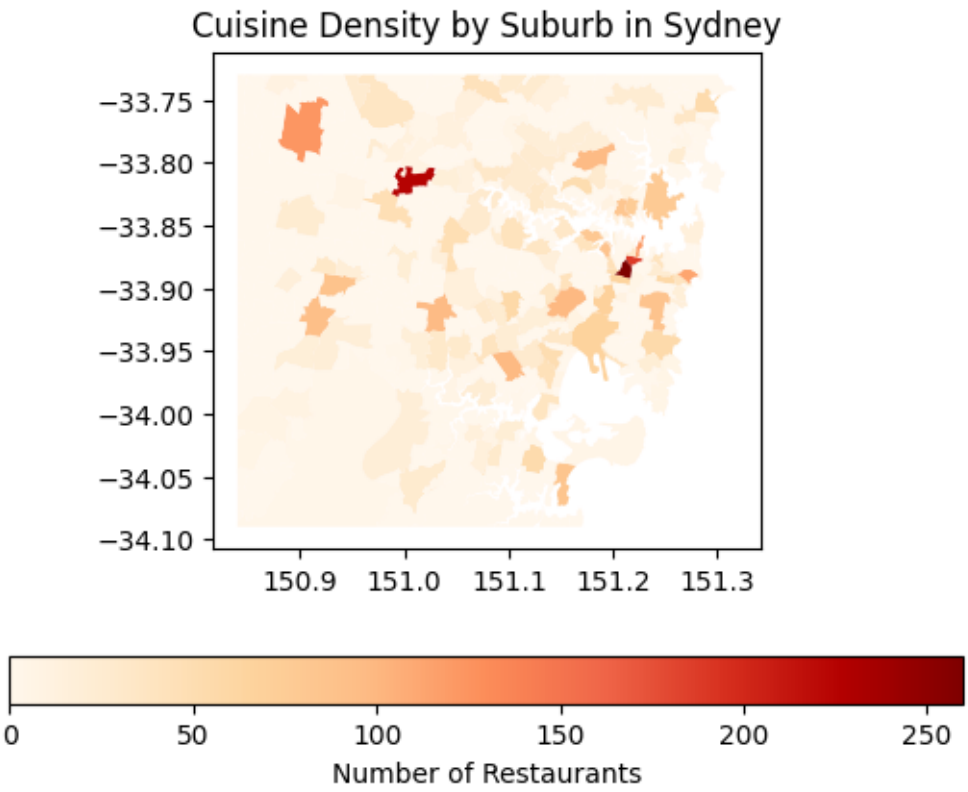
We will now create interactive visualizations using Plotly for better data exploration.

3. Cuisine Density Map

[7]:
```
# Clean and prepare data for visualization
cuisine_density = zomato_df.groupby('subzone')['cuisine'].count().reset_index()
cuisine_density.columns = ['subzone', 'restaurant_count']

# Merge with geojson data
geo_data_merged = sydney_geo.merge(cuisine_density, left_on='SSC_NAME',␣
 ↪right_on='subzone', how='left')
geo_data_merged['restaurant_count'] = geo_data_merged['restaurant_count'].
 ↪fillna(0)

# Plot cuisine density map using geopandas
```

```
plt.figure(figsize=(12, 8))
geo_data_merged.plot(column='restaurant_count', cmap='OrRd', legend=True,
                     legend_kwds={'label': "Number of Restaurants",␣
 ↪'orientation': "horizontal"})
plt.title("Cuisine Density by Suburb in Sydney")
plt.show()
```

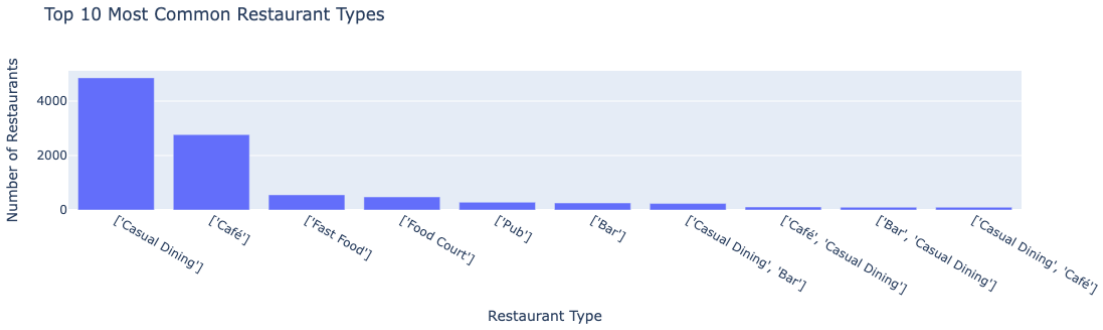
```
<Figure size 1200x800 with 0 Axes>
```

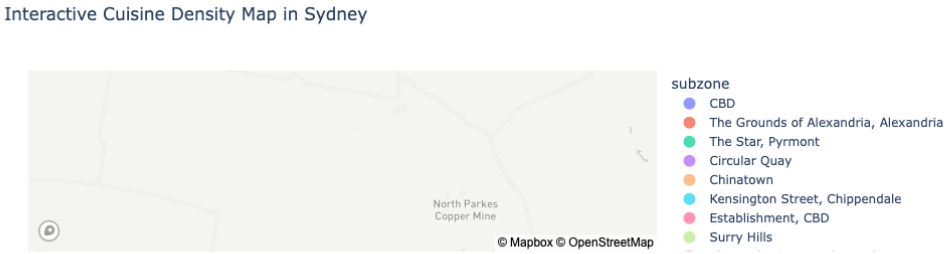


4. Interactive Plotly Visualizations

[8]:
```
# Interactive bar chart of restaurant types
top_types = zomato_df['type'].value_counts().head(10).reset_index()
top_types.columns = ['Restaurant Type', 'Count']

fig = px.bar(top_types, x='Restaurant Type', y='Count',
             labels={'Restaurant Type': 'Restaurant Type', 'Count': 'Number of␣
 ↪Restaurants'},
             title="Top 10 Most Common Restaurant Types")
fig.show()

# Interactive scatter map using Plotly and Mapbox
```

```
px.set_mapbox_access_token('pk.
 ↪eyJ1IjoieWVleWV1bmciLCJhIjoiY2OxaXpkd3piMDAwNjJqb2xzMjllYXhxMyJ9.
 ↪OEx8wPOgLWMreHCvoADdnQ')

# Drop rows where 'lat', 'lng', or 'cost' are NaN for plotting
zomato_df_clean = zomato_df.dropna(subset=['lat', 'lng', 'cost'])

fig = px.scatter_mapbox(zomato_df_clean, lat="lat", lon="lng", color="subzone",␣
 ↪size="cost",
                        hover_name="title", hover_data=["cuisine",␣
 ↪"rating_text"],
                        title="Interactive Cuisine Density Map in Sydney",␣
 ↪zoom=10)
fig.show()
```





Limitations of Non-Interactive Plotting

1. **Limited Depth of Information**
   - In non-interactive bar charts or scatter plots, users are limited to viewing only the fixed data points that are presented. For example, in a static bar chart of restaurant types,

users can only see the restaurant count for each type but cannot access further details like individual restaurant names, ratings, or other attributes.
    - For static scatter maps, users can only view the plotted data points but cannot explore additional details such as specific restaurant characteristics (e.g., cuisine type, ratings) by interacting with the map.
2. **Inability to Display Multi-Dimensional Data**
    - Traditional plotting libraries are limited in showing only two dimensions of data at a time, such as x and y axes in a bar chart. This can be restrictive when visualizing data that contains multiple important attributes. For example, in a static bar chart, you may only show the restaurant type and count, without any indication of price range, rating, or location.
    - On a scatter map, static plots cannot dynamically represent multiple attributes (such as restaurant cost, ratings, and cuisine) in one plot.
3. **Lack of Customization and Exploration**
    - Non-interactive charts are static and do not allow users to customize their exploration of the data. If a user is interested in filtering the dataset by a specific type of restaurant or region, they would need to generate a new static chart. Static charts do not allow dynamic filtering, zooming, or selecting specific data points.

How Interactive Libraries Solve These Issues

1. **Dynamic Information Display**:
    - **Hover Tooltips**: With interactive libraries such as Plotly, users can hover over data points to reveal detailed information such as restaurant names, ratings, cuisines, and cost. This significantly enhances the user's ability to understand and explore the dataset without cluttering the visual representation with excessive labels.
    - **Zooming and Panning**: For maps like the interactive scatter map using Plotly and Mapbox, users can zoom into specific areas, drag to pan across different regions, and drill down to explore detailed information for each restaurant. This provides a much richer exploratory experience compared to static maps.
2. **Handling Multi-Dimensional Data**:
    - Interactive libraries allow for more dimensions of data to be displayed. For example:
        - **Color Coding**: You can color-code data points based on different restaurant characteristics like cuisine type or rating, making it easier to distinguish between different categories in one plot.
        - **Size Variation**: In a scatter map, the size of each point can represent a restaurant's cost, allowing users to visually assess both the location and price range in one view.
    - These additional visual cues provide users with a more comprehensive understanding of the data by allowing them to analyze multiple variables simultaneously.
3. **Customization and Data Exploration**:
    - Interactive libraries enable users to dynamically filter and customize the data they are viewing. For example, in a restaurant map, users can apply filters to view restaurants in specific areas or with specific ratings. This ability to interactively explore and drill down into the data empowers users to gain more tailored insights.
    - **User Controls**: Features such as sliders or dropdowns in Plotly allow users to adjust parameters like price range, rating categories, or restaurant type without needing to generate new charts.

Conclusion Interactive plotting libraries like Plotly or Bokeh significantly improve upon the limitations of traditional, non-interactive charts by allowing for deeper information exploration, multi-dimensional data representation, and user customization. By incorporating interactive elements, users can dynamically explore the dataset, visualize more attributes in a single plot, and tailor the visualizations to their specific interests.

5. Export Cleaned Data for Tableau

[9]:
```
# Export cleaned data for Tableau
zomato_df.to_csv('zomato_cleaned_data_for_tableau.csv', index=False)
```

Link to Tableau Dashboard: Zomato Sydney Data Analysis