Zomato is a leading Indian restaurant aggregator and online food delivery platform that connects users to a vast network of dining options. Founded in 2008, Zomato operates in 24 countries, offering services like restaurant discovery, online ordering, and table reservations. With its user-friendly interface and extensive database, Zomato has become a trusted resource for food enthusiasts. Data yang digunakan yaitu dari udemy ini mencatat, kolom yang terdiri dari :

| **Atribut** | **Penjelasan** |
| --- | --- |
| Index | Unique identifier for each entry. |
| url | Restaurant's webpage link. |
| addres | Full address of the restaurant. |
| name | Restaurant's name. |
| online\_order | Offers online ordering (Yes/No). |
| book\_order | Offers table booking (Yes/No). |
| rate | User rating of the restaurant. |
| votes | Number of user votes. |
| phone | Restaurant's contact number. |
| locations | Area or locality of the restaurant. |
| rest\_type | Type/category of the restaurant. |
| dish\_like | Popular dishes liked by customers. |
| cuisines | Types of cuisine offered. |
| approx\_cost(for two people) | Estimated cost for two. |
| reviews\_list | Collection of user reviews. |
| menu\_item | List of menu items. |
| listed\_in(type) | Dining category (e.g., lunch, dinner). |
| listed\_in(city) | City where the restaurant is located |

This analysis focuses on exploring various aspects of Bengaluru's restaurant scene using data from Zomato. The study includes creating a geographical map of restaurant outlets across the city and identifying the most popular dishes at specific locations through a word cloud visualization. Additionally, it aims to uncover the top cuisines favored by Bengaluru residents, examine the correlation between restaurant ratings and pricing, and showcase the city's top-rated restaurants through a bar chart. These insights provide a comprehensive understanding of the dining preferences and trends in Bengaluru.

**Preprocessing**

import data

| import sqlite3  import pandas as pd  import numpy as np  db = sqlite3.connect('zomato\_rawdata.sqlite')  df= pd.read\_sql\_query('SELECT \*FROM Users', db)  df.head() |
| --- |

duplicate data

| df.duplicated().any() |
| --- |

handling null values

| df.isnull().sum()  df\_clean = df.copy()  #rate  df['rate'].replace(('NEW' , '-') , np.nan , inplace=True)  df\_clean['rate'] = df\_clean['rate'].apply(lambda x: float(x.split('/')[0]) if type(x)==str else x)  df\_clean1 = df\_clean.dropna(subset=['rate'])  #approx\_cost(for two people)  df\_clean['approx\_cost(for two people)'] = df\_clean['approx\_cost(for two people)'].replace({'\$': '', ',': ''}, regex=True)  df\_clean['approx\_cost(for two people)'] = pd.to\_numeric(df\_clean['approx\_cost(for two people)'], errors='coerce')  df\_clean['approx\_cost(for two people)'].fillna(df\_clean['approx\_cost(for two people)'].mean(), inplace=True)  #cuisines  df\_clean['cuisines'].fillna('Tidak Terdeteksi', inplace=True)  #rest\_type  df\_clean['rest\_type'].fillna('Tidak Terdeteksi', inplace=True)  #dish\_like  df\_clean.drop('dish\_liked', axis=1, inplace=True) #di drop karen nilai null nya sudah sangat besar  #phone  df\_clean['phone'].replace('none', np.nan, inplace=True)  df\_clean['phone'] = df\_clean['phone'].str.replace(r'\s+|\r\n|,', ' ', regex=True)  df\_clean['phone'].fillna('Tidak Terdeteksi', inplace=True) |
| --- |

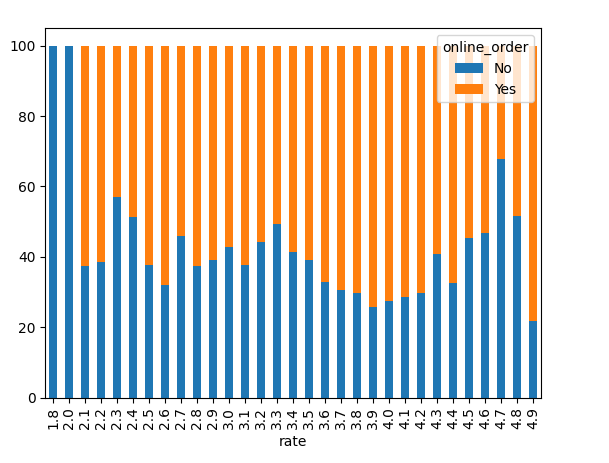
**Question 1.**

How does ordering food online differ from ordering directly based on the rating of a restaurant?

| pvt = pd.crosstab(df\_clean['rate'] , df['online\_order'])  pvt.sum()  pvt.plot(kind='bar', stacked= True) |
| --- |

Normalization

| pvt.sum(axis=1).astype(float)  normlz\_pvt = pvt.div(pvt.sum(axis=1).astype(float), axis = 0)  (normlz\_pvt\*100).plot(kind='bar', stacked= True) |
| --- |

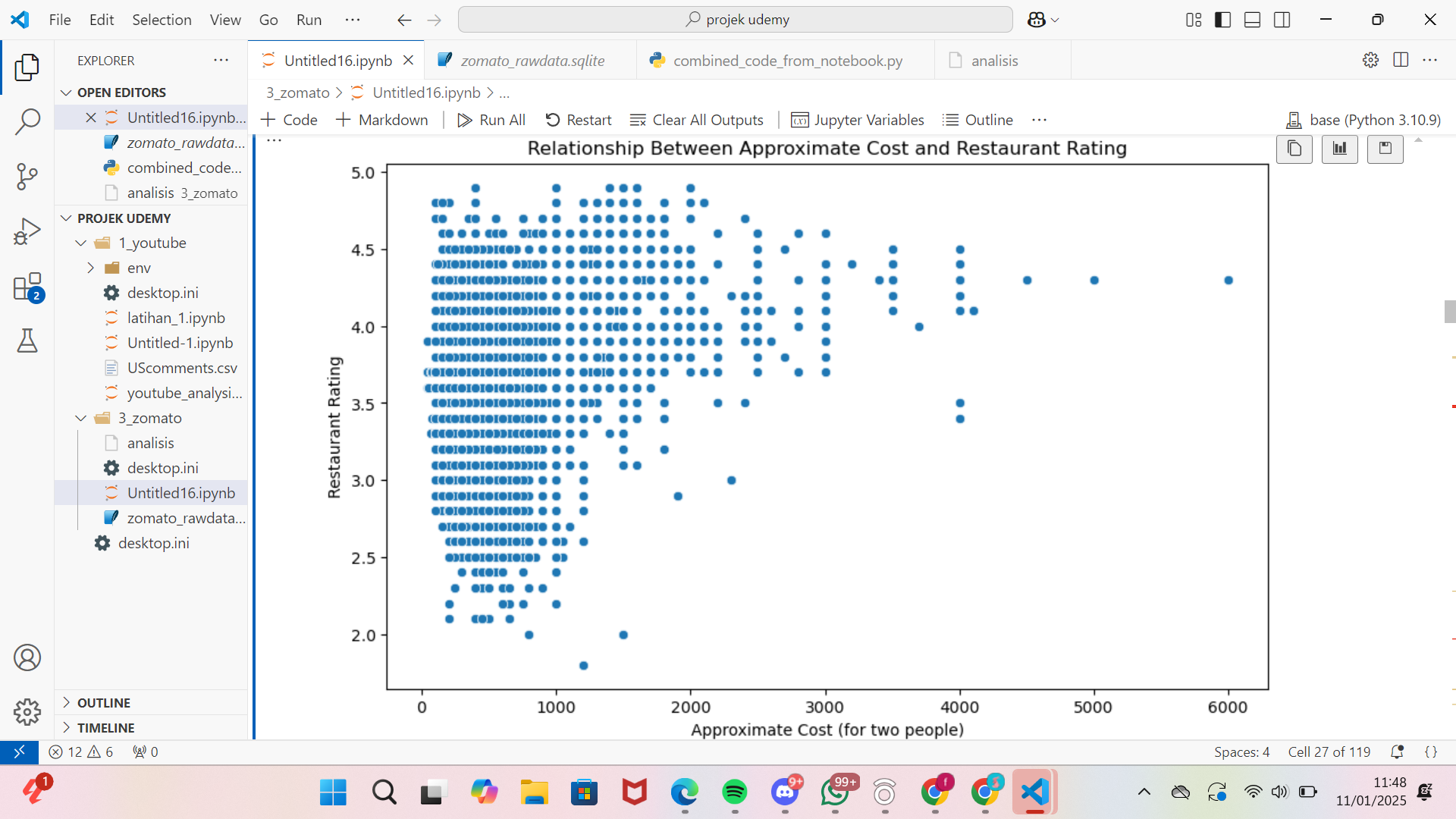


After normalization, it was found that restaurants with high ratings (4.0 and above) attract more online customers, as seen from the dominance of the orange color on the graph. This indicates a greater interest on food ordering platforms, possibly due to more reliable service and better user experience, reflecting customer trust in food quality and service. Restaurants with ratings below 3.0 tend to attract fewer customers who do not order online, as seen from the dominance of the blue color on the graph. This suggests difficulties in attracting online customers due to poor service quality or user experience, leading them to prefer visiting the restaurant in person. For medium ratings (3.0–3.8), there is a shift from offline to online ordering, with an increase in the number of customers ordering online. This indicates that restaurants with medium ratings are optimizing their online services.

**Question 2.**

How is the relationship between restaurant prices and the ratings given by customers?

| import matplotlib.pyplot as plt  import seaborn as sns  df\_clean['rate'] = pd.to\_numeric(df\_clean['rate'], errors='coerce')  df\_clean['approx\_cost(for two people)'] = pd.to\_numeric(df\_clean['approx\_cost(for two people)'], errors='coerce')  # scatter plot  plt.figure(figsize=(10, 6))  sns.scatterplot(x=df\_clean['approx\_cost(for two people)'], y=df\_clean['rate'])  plt.title('Relationship Between Approximate Cost and Restaurant Rating')  plt.xlabel('Approximate Cost (for two people)')  plt.ylabel('Restaurant Rating')  plt.show() |
| --- |



The scatter plot shows the correlation between average expenditure and restaurant ratings. Many low-cost establishments (0–1000) have ratings between 2.5 and 4.5, indicating that budget-friendly options are common but quality varies. In contrast, higher-cost restaurants (above 2000) are less frequent but generally receive better ratings (around 4.0 and above), suggesting superior service and experiences. Lower-rated establishments (below 3.0) are mostly found in the low-cost segment, hinting at potential quality issues. While a strong correlation isn't clear, there is a trend of higher-priced restaurants achieving better ratings, indicating that food and service quality often align with costs.

**Quesion 3**

What restaurant type categories are often liked or frequently visited by people?

| data= df\_clean1.dropna(subset=['rest\_type'])  df\_clean1['rest\_type'].unique()  # mengelompokkan jenis restoran ke dalam kategori  def categorize\_rest\_type(rest\_type):  if 'Quick Bites' in rest\_type or 'Takeaway' in rest\_type or 'Food Truck' in rest\_type or 'Kiosk' in rest\_type:  return 'Quick Service / Fast Food'  elif 'Casual Dining' in rest\_type:  return 'Casual Dining'  elif 'Fine Dining' in rest\_type:  return 'Fine Dining'  elif 'Cafe' in rest\_type:  return 'Cafe'  elif 'Dessert Parlor' in rest\_type or 'Sweet Shop' in rest\_type or 'Bakery' in rest\_type:  return 'Dessert and Sweet Shops'  elif 'Pub' in rest\_type or 'Bar' in rest\_type or 'Lounge' in rest\_type or 'Microbrewery' in rest\_type:  return 'Bars, Pubs, and Lounges'  elif 'Food Court' in rest\_type or 'Club' in rest\_type or 'Dhaba' in rest\_type or 'Confectionery' in rest\_type:  return 'Specialized Services'  else:  return 'Others'  # Menambahkan kolom kategori ke dataframe  df\_clean1['rest\_category'] = df\_clean1['rest\_type'].apply(lambda x: categorize\_rest\_type(str(x)))  print(df\_clean1['rest\_category'].value\_counts()) |
| --- |

output

Quick Service / Fast Food 16804

Casual Dining 12277

Cafe 4031

Dessert and Sweet Shops 3281

Others 2657

Bars, Pubs, and Lounges 1643

Specialized Services 570

Fine Dining 402

From the output above, it is found that there are 16,804 restaurants in the "Quick Service / Fast Food" category, making it the most dominant category. We can analyze the "Quick Bites" subcategory to understand characteristics such as average ratings, prices, and types of services. This approach helps identify customer preferences for fast-food restaurants with significant market potential.

**Question 4**

From customer reviews, what words are commonly used when reviewing that contribute to the performance rating of a restaurant?

| from nltk import FreqDist  fd= FreqDist()  for word in total\_reviews\_1D:  fd[word] = fd[word] + 1  fd.most\_common(20) |
| --- |

[('place', 184236),

('I', 176489),

('good', 173562),

('food', 158990),

('The', 128836),

('chicken', 60607),

('service', 53897),

('ordered', 51008),

('taste', 49251),

('great', 45759),

('really', 45120),

('ambience', 42271),

('time', 42100),

('try', 41974),

('one', 41641),

('It', 40195),

('also', 39728),

('like', 38823),

('visit', 35637),

('We', 34289)]

Based on the output above, the most frequently reviewed elements are location/restaurant, food, and service. The term 'good' indicates customer satisfaction with the quality of food and service. The frequency of the word 'taste' highlights the importance of the flavor of the food. Words such as 'chicken' and 'ambience' reflect customers' focus on specific menu items and the restaurant's atmosphere. Terms like 'ordered' and 'try' indicate discussions about the food ordered, while the use of 'great' and 'really' signifies high satisfaction. These findings suggest that food quality, service, and ambiance are highly valued, emphasizing the need for restaurants to maintain these standards.

| from nltk import FreqDist, bigrams,trigrams  bi\_grams = bigrams(total\_reviews\_1D)  bi\_grams  fd\_grams = FreqDist()  for bigram in bi\_grams:  fd\_grams[bigram]= fd\_grams[bigram] + 1  fd\_grams.most\_common(20) |
| --- |

outputnya

[(('The', 'food'), 13629),

(('really', 'good'), 12651),

(('I', 'ordered'), 12164),

(('This', 'place'), 10726),

(('must', 'try'), 10313),

(('We', 'ordered'), 9727),

(('I', 'would'), 9404),

(('visit', 'place'), 8828),

(('food', 'good'), 8720),

(('The', 'place'), 8409),

(('good', 'food'), 8287),

(('good', 'place'), 7323),

(('main', 'course'), 7103),

(('ice', 'cream'), 6922),

(('non', 'veg'), 6893),

(('The', 'ambience'), 6676),

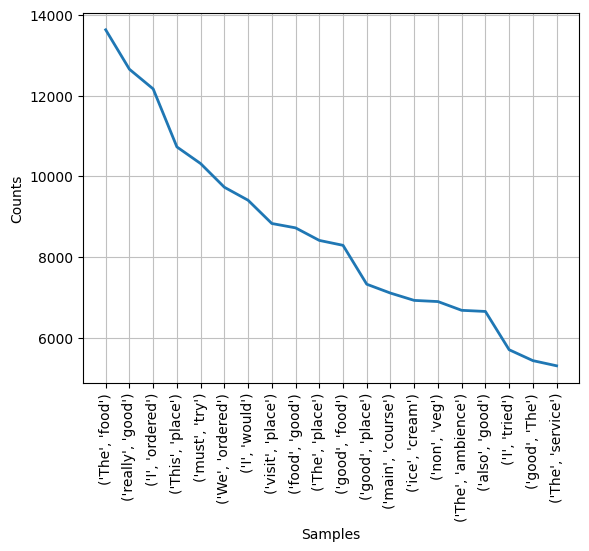
(('also', 'good'), 6648),

(('I', 'tried'), 5699),

(('good', 'The'), 5429),

(('The', 'service'), 5301)]

| fd\_grams.plot(20) |
| --- |



Many customers comment on the food at restaurants using the word 'food,' often followed by 'The,' likely as part of sentences like "The food is great." Customer satisfaction with food becomes a central focus of reviews, with bigrams such as 'food', 'good', and 'taste.' Restaurant recommendations are evident through bigrams like 'must try' and 'I would,' indicating popularity. The restaurant's atmosphere, reflected in the bigram 'The ambience,' influences customer experience. Service quality and orders are also crucial, highlighting attention to service and food. Key menu items and dish variations, such as 'main course,' 'ice cream,' and 'non-veg,' are frequently discussed, reflecting specific customer evaluations. This bigram analysis provides insight into customer experiences, emphasizing food quality, ambiance, and service, while offering valuable guidance for restaurant improvement and marketing strategies.

| tri\_grams = trigrams(total\_reviews\_1D)  print(tri\_grams)  tri\_graams = FreqDist()  for tri\_grms in tri\_grams:  tri\_graams[tri\_grms] = tri\_graams[tri\_grms] + 1  tri\_graams.most\_common(20) |
| --- |

[(('veg', 'non', 'veg'), 1525),

(('must', 'visit', 'place'), 1487),

(('The', 'food', 'good'), 1339),

(('place', 'hangout', 'friends'), 1165),

(('I', 'must', 'say'), 1131),

(('I', 'would', 'recommend'), 1059),

(('I', 'visited', 'place'), 1054),

(('I', 'would', 'say'), 979),

(('place', 'hang', 'friends'), 967),

(('food', 'really', 'good'), 886),

(('nFood', 'nAmbience', 'nService'), 845),

(('A', 'must', 'visit'), 768),

(('The', 'ambience', 'good'), 751),

(('A', 'good', 'place'), 737),

(('North', 'Indian', 'food'), 675),

(('I', 'would', 'like'), 619),

(('nFood', 'nService', 'nAmbience'), 607),

(('A', 'must', 'try'), 603),

(('I', 'really', 'liked'), 595),

(('I', 'ordered', 'chicken'), 582)]

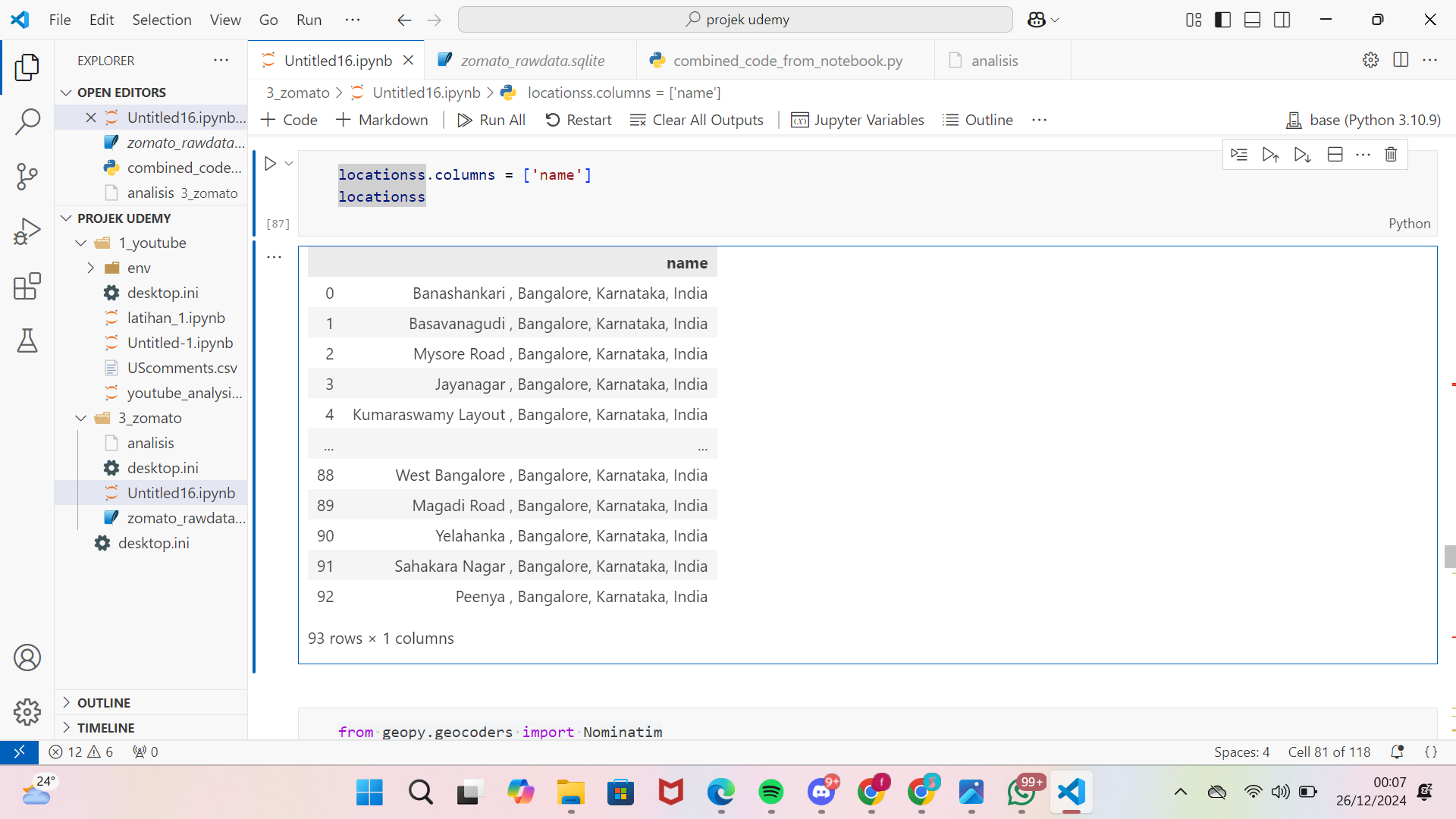
Based on the trigram analysis of the reviews, it can be concluded that the majority of reviews focus on food quality, ambiance, and recommendations from reviewers. Trigrams like 'must visit place' and 'The food good' indicate that the establishment has a good reputation, with food and ambiance being the most appreciated aspects. Social activities, such as gathering with friends, are also a key reason for visitors, as reflected in the trigram 'place hangout friends.' Additionally, certain types of food, such as 'North Indian food' and chicken dishes, receive special attention from reviewers. To enhance appeal, business owners are advised to consistently maintain food quality, ambiance, and provide facilities that support social activities. This analysis can be further expanded using approaches like sentiment analysis or categorizing reviews into different groups.

**Question 5**

Where is the location or area of the restaurant that is most frequently visited by customers?

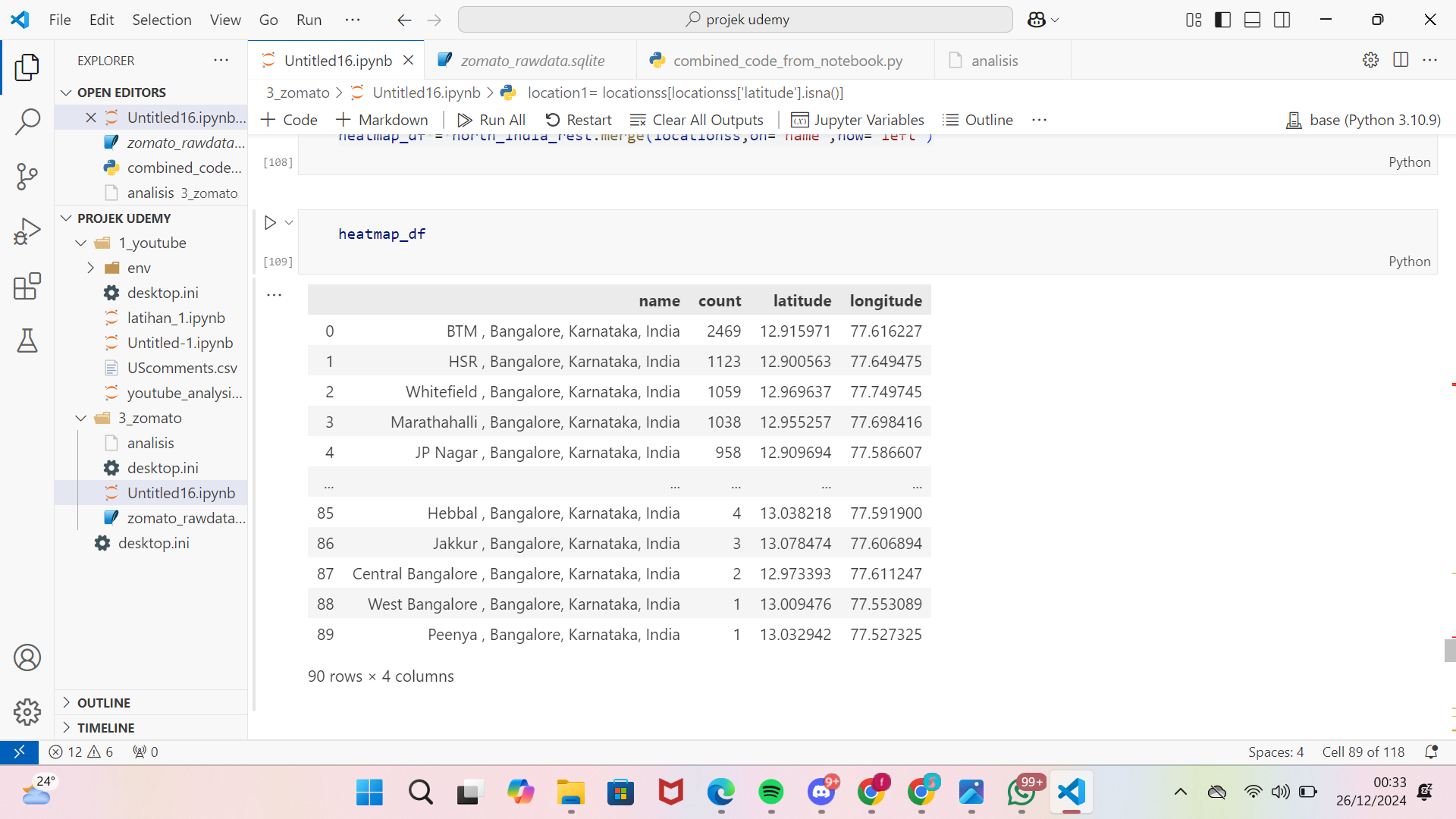
| df['location']  df['location'].unique()  len(df['location'].unique())  df\_copy = df.copy()  df\_copy['location'].isnull().sum()  df\_copy = df\_copy.dropna(subset=['location'])  df\_copy['location'].isnull().sum()  locationss.columns = ['name']  locationss |
| --- |

output



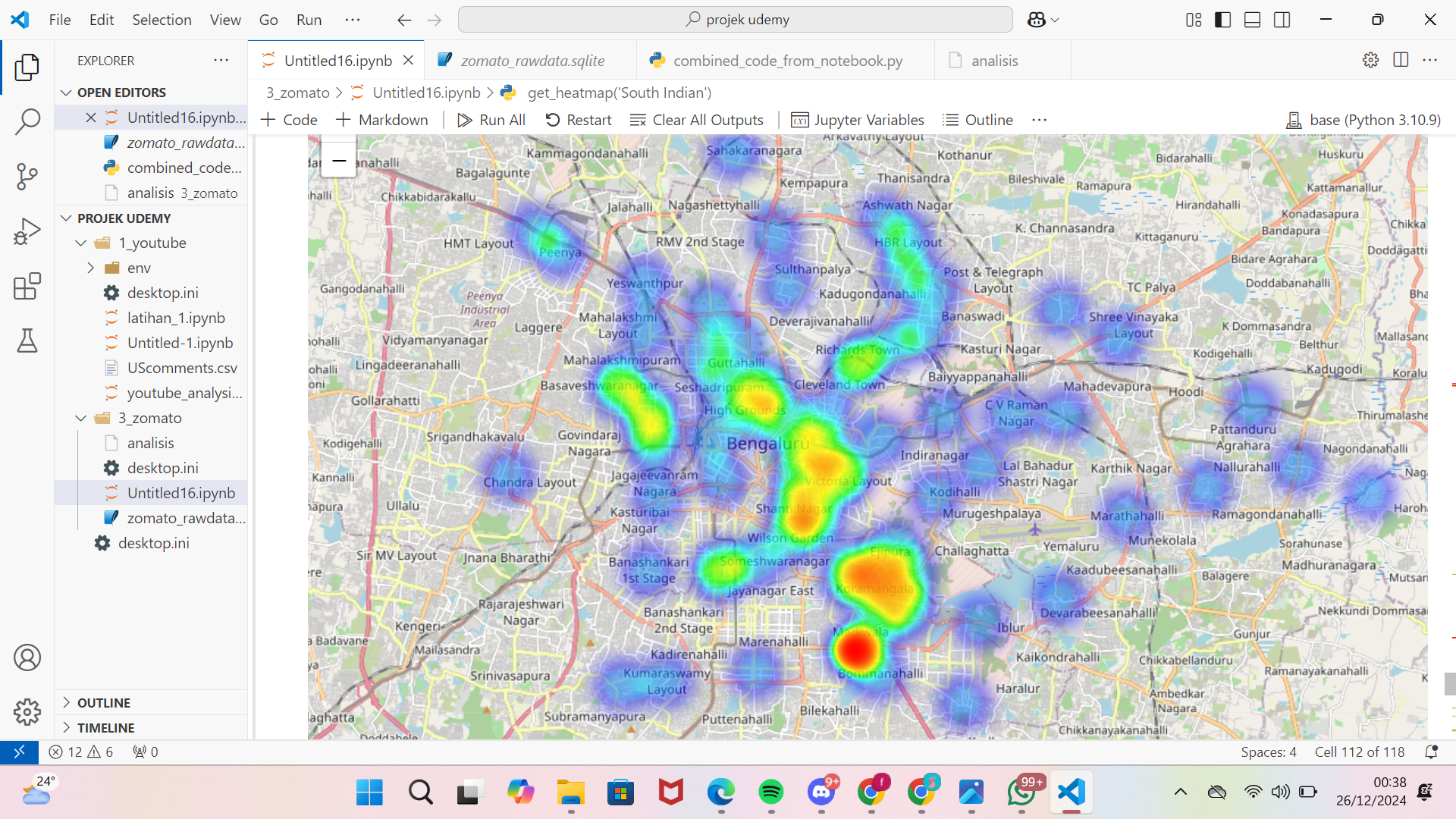
This code is used to display the results, which show the locations of restaurants in various areas of Bangalore, Karnataka, India. These locations include Banashankari, Basavanagudi, Mysore Road, Jayanagar, Kumaraswamy Layout, and others, with a total of 93 locations recorded in the dataset.

| from geopy.geocoders import Nominatim  geolocator= Nominatim(user\_agent="app", timeout=None)  lat=[]  lon=[]  for location in locationss['name']:  location = geolocator.geocode(location)  if location is None:  lat.append(np.nan)  lon.append(np.nan)  else:  lat.append(location.latitude)  lon.append(location.longitude)  locationss['latitude'] = lat  locationss['longitude'] = lon  locationss |
| --- |



The data presented consists of geographical coordinates in the form of longitude and latitude values for each restaurant location in Bangalore, India. This data is used to accurately map the positions of restaurants and serves as the basis for creating heatmap visualizations to analyze the spatial distribution of restaurants in the area.

| import folium  basemap = folium.Map()  from folium.plugins import HeatMap  heatmap\_df.columns  HeatMap(heatmap\_df[['latitude', 'longitude', "count"]]).add\_to(basemap)  basemap |
| --- |



The image above displays a heatmap of Zomato restaurant locations in Bengaluru, India. Red and yellow points indicate high restaurant density, while blue or green areas represent lower concentrations. Areas like Koramangala and Indiranagar appear to be popular culinary hubs, possibly due to supporting infrastructure, dense population, or tourist appeal. This map can help identify preferred locations for culinary business development or assist customers in finding dining spots. Further analysis is required to explore factors such as restaurant types or peak visitation times.