THE FOOD PROJECT

Project 1: FOODHUB
Course name: PHYTHON BASICS

Date 8/04/2022

SANIA

Contents / Agenda

Executive Summary

Executive Summary

"Let food be thy medicine and medicine be thy food." "We all eat, and it would be a sad waste of opportunity to eat badly."
-FOODIE

Life is getting busier and busier with times and having a convenient food application solely focused on improving customer experience is a go-to!

New York being a busy city, with students and professionals having a hectic lifestyle, a hand-curated list of cuisines and delicacies just a touch away and delivered at the doorstep is delightful!

The objective is clear "**Customer experience enhancement** " by analyzing tones of food data that the food aggregator company has stored from the different orders made by the registered customers in their online portal.

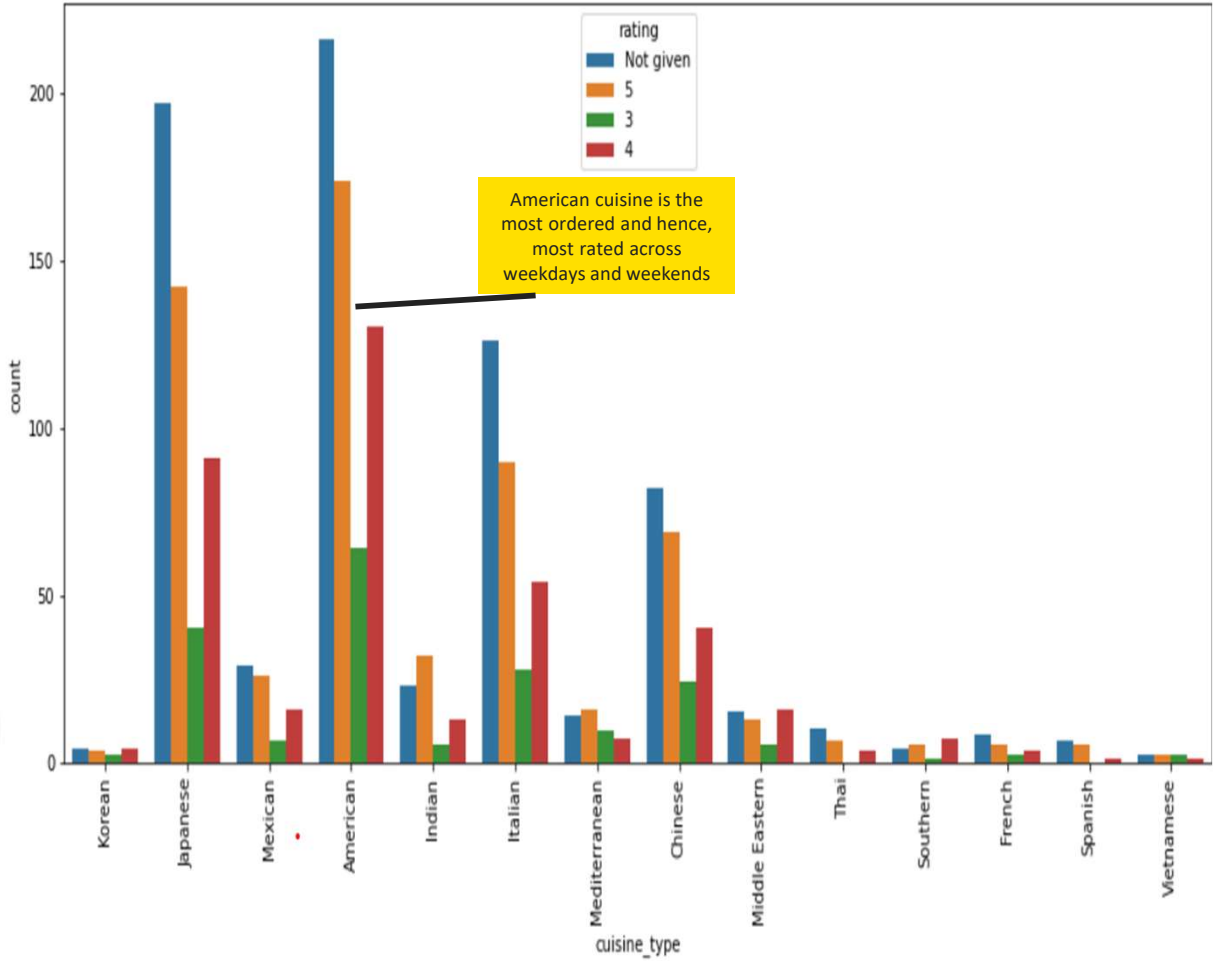
"Life is uncertain. Eat dessert first."
– Ernestine Ulmer

Conclusions

CONCLUSIONS…

While going through…

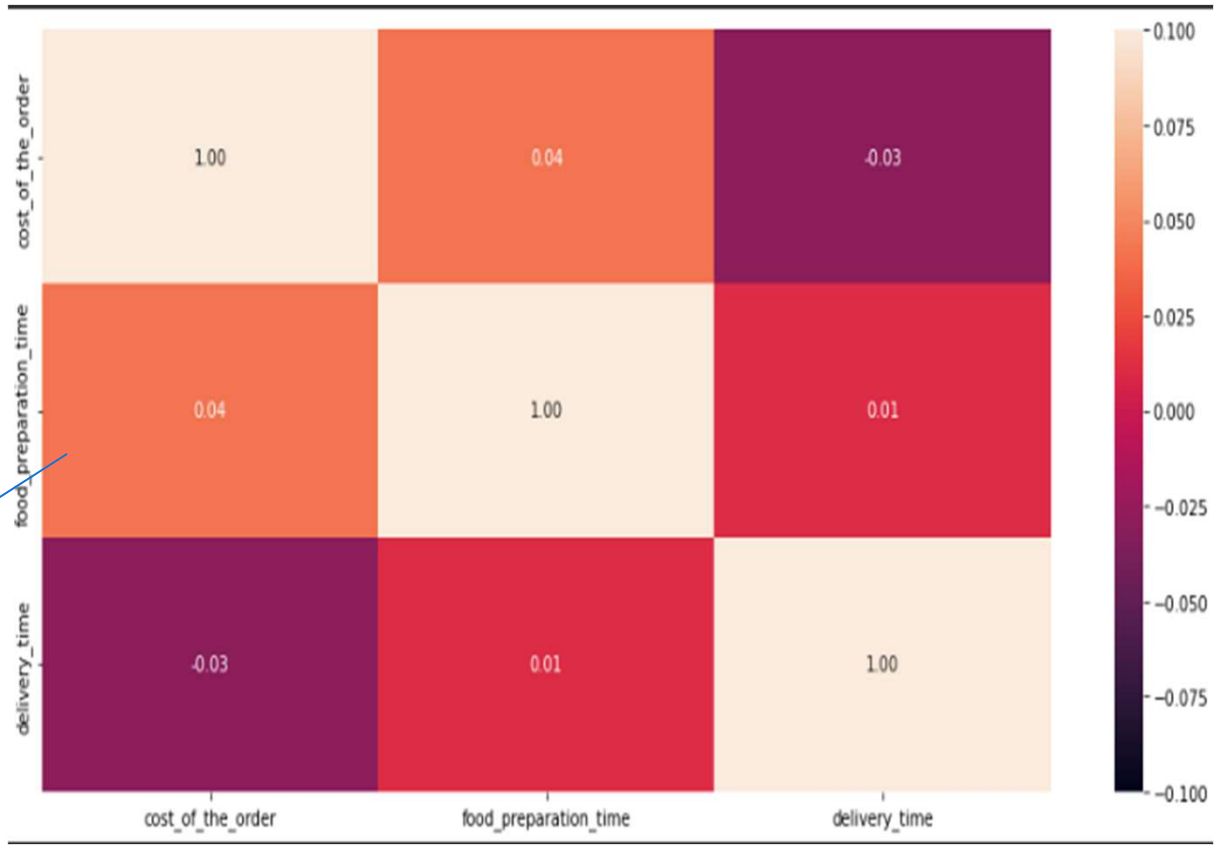
- **American cuisine** is ordered the most in the weekends, **most ordered** and hence, **most rated** across **weekdays and weekends**
- **10.54%** take more than 60 minutes of time to deliver
- Most orders are placed on the weekends than on the weekdays.
- **'Not given'** highest count of rating followed by 5 and then 4. **Almost 35%** of customer ratings are **not received**
- The range of the cost of the orders is between **~4–35 and around 75 %** of the cost of orders is below ~$23. Cost of the orders is slightly **right-skewed**



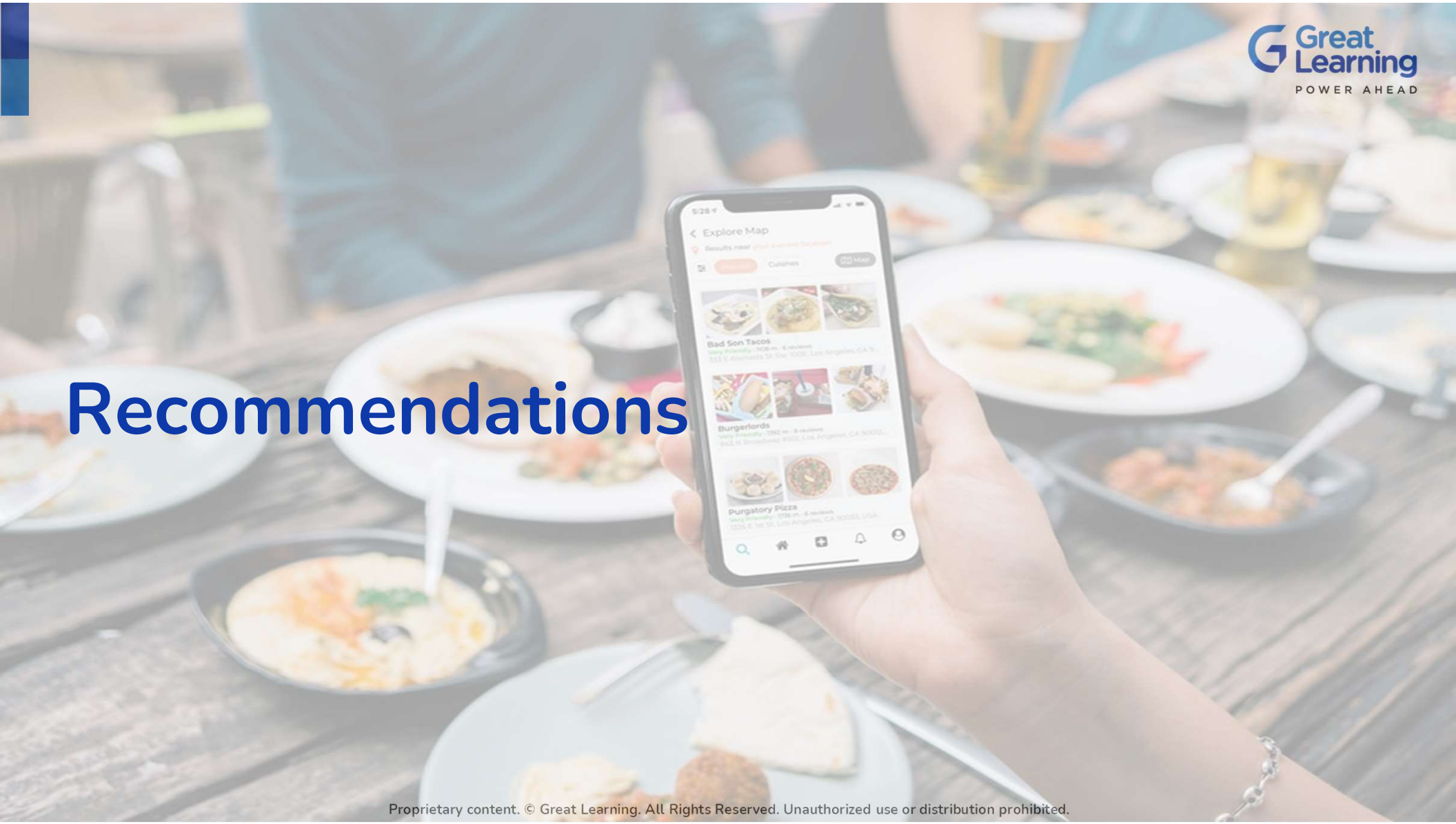
CONCLUSIONS…

Also came across…these

- **Korean, Mediterranean** and **Vietnamese** cuisine types, have cost outliers
- Median cost of order of French cuisine is the highest compared to other cuisines
- Weekdays have a higher median of the **delivery time** than weekends
- Food preparation time vs the ratings has a **very slight variation** across the **means**
- **Delivery time and cost of order** has a **negative correlation**
- **Food preparation time** has a **slightly more positive correlation** with **cost of order** than **delivery time**.
- **Cost of order** has a very weak association with the food preparation **time** but greater than the association between delivery time and food preparation time



Recommendations



Recommendations…

Aim for 5:

My first recommendation to the food app will be to aim for a rating of 5. We can see that maximum of the ratings count are either 'Not given' or that a rating 3 is mostly seen for higher delivery times

Must have

For '**Not givens**':

- Adding a **must rate option** before making the next order can be added as a setting in the application
- This can be frustrating for many customers and so adding a **"delivery person" testimonial** can add an **emotional/human** touch on the importance of their rating. The testimonials can read as: "*Your ratings help me achieve better…xyz,delivery person*"
- We can also i**ncentivize** the customers by adding **a rate and spin the wheel** for offers option after every 3-4 ratings.

For **improving the ratings from 3 to 5:**

we can see that during the weekdays the median of delivery time is higher and delivery time also affects the ratings. This can either be due to traffic during the weekday or lesser number of delivery persons. **Recommending adding a few more delivery persons or outsourcing delivery to professional agencies to mostly ordered restaurants on weekdays.**

Restaurants can be guided on **optimizing a workflow process of packaging**. Around **10.54%** of orders take more than 1 hour of total time to be delivered, and this might be one of the reasons for a bad customer experience.

Can have

Constantly in touch:

- We saw that **around 10.54%** of orders take a total time of **1 hour** or more to be delivered, we can add a feature of chats for the customer to constantly contact the delivery agent.
- **Human behavior is greatly impacted** by communication, if the agent is in contact with the customer, it adds value.

More data:

- Adding a **review** column to understand **why are certain cuisines rated low or how can we improve our service**.
- Adding a voice review option, and the speech data can be converted to text and analyzed using ML for words that describe the service. This can be particularly useful for customers who do not enjoy reviewing by typing.

Recommendations...

Promos on the Go:

My second recommendation is providing offers and promotional codes

Must have

Week offers:

Adding offers like a 5-10 % off on selected cuisines like **American cuisine** followed **by Japanese, Italian and Chinese**. Since these are the most ordered cuisines mostly. And adding **corporate offers or student offers on weekdays on** corporate registered emails or student registered emails **to increase sales. Adding free deliveries and buy one get one on selected cuisines** is also a good idea for weekdays to **increase weekday revenue as well as customer satisfaction.**

Rated:

Providing incentives in the **form of reward points** that can be used later when a certain amount is reached to customers for ordering the months highly rated cuisine or from the **most ordered restaurants.**

Can have

Outliers:

Korean, Mediterranean and Vietnamese cuisines have **cost outliers**. These cuisines are ordered but not in a larger count like the American cuisine. These cuisines are less ordered than the orders and hence, less rated. It is fair to assume that **either due to the cost of order** (students) or freshness of the food or maybe the locations of these restaurants are farther from **prime consumer location that these restaurants are not ordered from** much.

Adding a **one on one (buy one and the other dish free) promo to** these can improve the restaurant's business as well as the food app can promote cultural variation.

Business Problem Overview and Solution Approach

Business Problem Overview and Solution Approach

Business Problem

The food aggregator company aims to improve their business in the food delivery industry. Its objective is to enhance customer satisfaction by addressing unmet needs and understanding barriers that might lead to poor rating.

Solution Approach

The solution involves in-depth statistical and exploratory data analysis (EDA) of the food app data and visualizations using python as a tool that support the hypotheses. “cuisine type” as well as “ratings” were few of the data that has been utilized further to achieve results

Data Overview

Q1. How many rows and columns are present in the data?

Number of rows= 1898; Number of columns=9

Q2. What are the datatypes of the different columns in the dataset?

1. The order_id, customer_id, food_preparation_time and delivery_time are integers.
2. restaurant_name, cuisine_type, day_of_the_week and rating are object.
3. cost of the order is float

Q3. Are there any missing values in the data? If yes, treat them using an appropriate method.

No, There are 0 missing values in the data

```
df.isnull().sum() #code to find

order_id                  0
customer_id               0
restaurant_name           0
cuisine_type              0
cost_of_the_order         0
day_of_the_week           0
rating                    0
food_preparation_time     0
delivery_time             0
dtype: int64

print(df.isnull().sum().sum())

0
```

Q4. Check the statistical summary of the data. What is the minimum, average, and maximum time it takes for food to be prepared once an order is placed?

For statistical summary,

df.describe(include='all').T

The minimum time for food preparation is:

20.00 minutes

The average time for food preparation is: 27.37 minutes

The maximum time for food preparation is: 35 minutes

Q5. How many orders are not rated?

736 orders are not rated. We can see this by using the code- df['rating'].value_counts()

```
# Write the code here
df['rating'].value_counts()
```

```
Not given    736
5            588
4            386
3            188
Name: rating, dtype: int64
```

EDA-Univariate Analysis

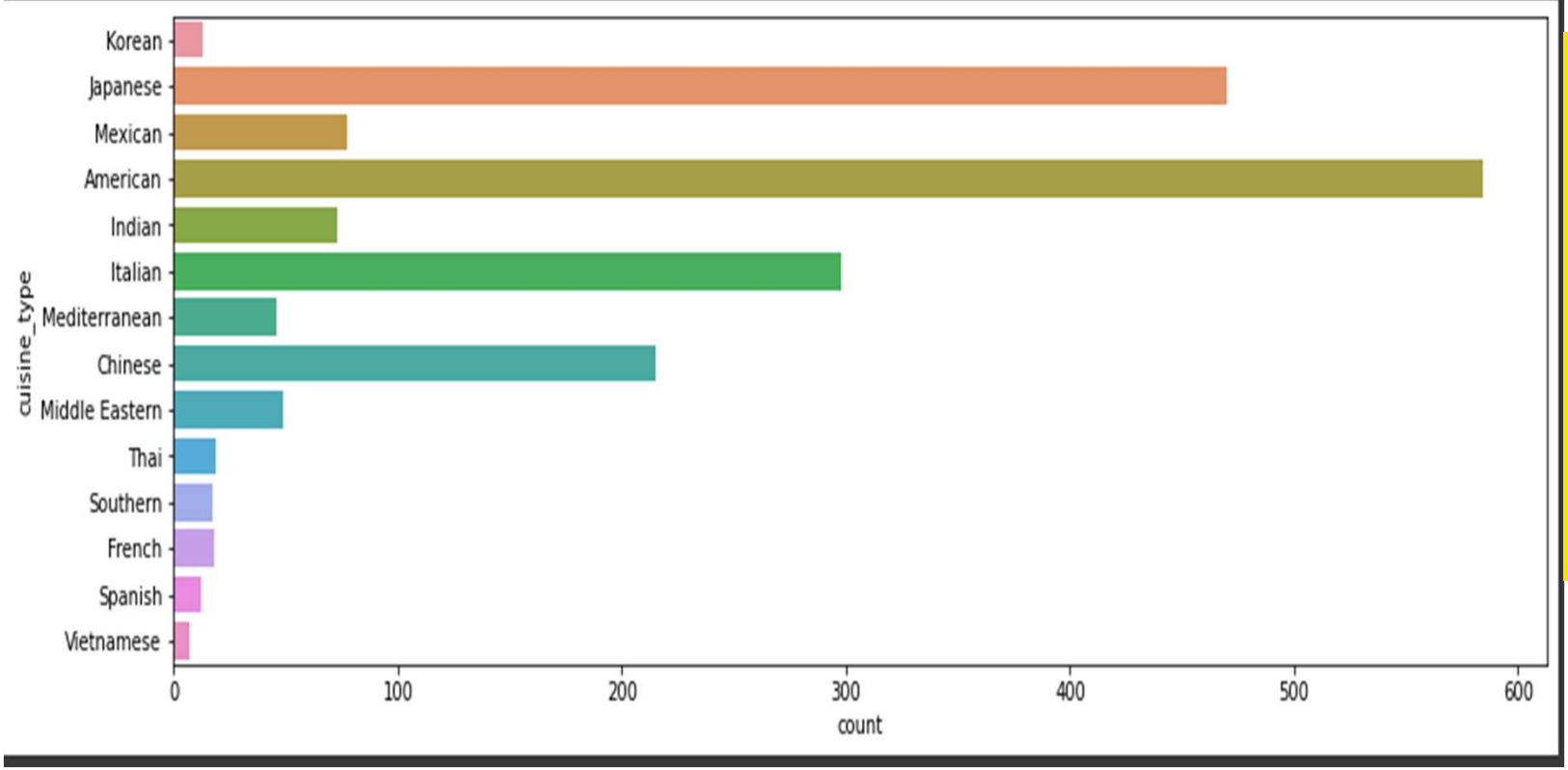
Univariate Analysis

Q6. Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)

Order ID:	df['**order_id**'].nunique(). There are 1898 number of unique Order IDs in the dataset
Customer ID:	df['**customer_id**'].nunique(). There are 1200 number of unique Customer IDs in the dataset
Restaurant name:	df['**restaurant_name**'].nunique(). There are 178 unique number of restaurant names
Cuisine type:	df['**cuisine_type**'].nunique(). There 14 unique cuisine type

Univariate Analysis- Countplot for cuisine type

```
plt.figure(figsize = (15,5))
sns.countplot(data = df, y = 'cuisine_type'); ## Create a countplot for cuisine type.
```

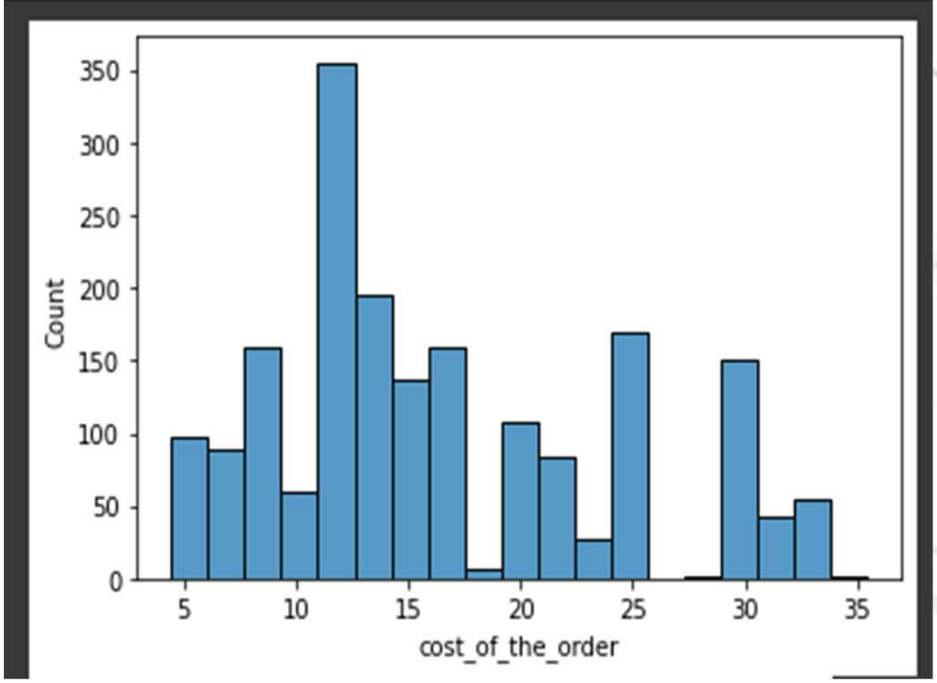


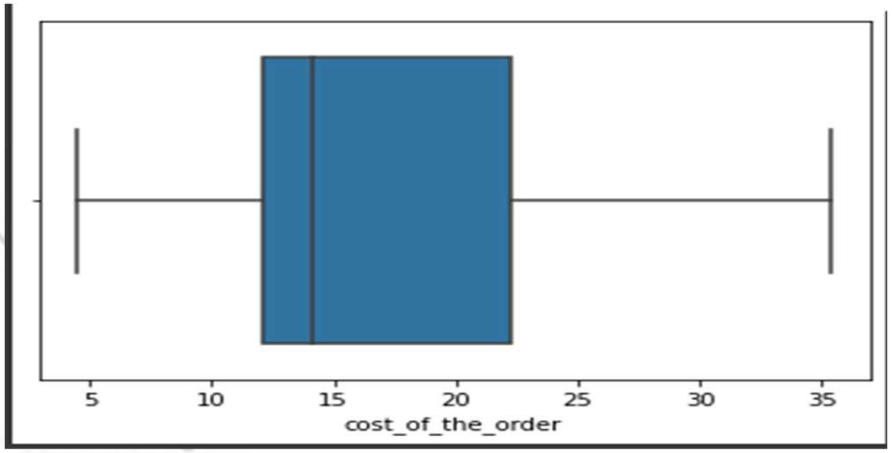
Observations:

It can be observed that American cuisine_type was ordered the maximum number of times followed by Japanese cuisine. Vietnamese cuisine was ordered the least number of times

Univariate Analysis- Cost of the order

```
sns.histplot(data=df,x='cost_of_the_order') ## Hist
plt.show()
sns.boxplot(data=df,x='cost_of_the_order') ## Boxpl
plt.show()
```





Observations:

From the **histogram** above, we can see that **maximum number of orders had a cost almost around $11 or $12.**

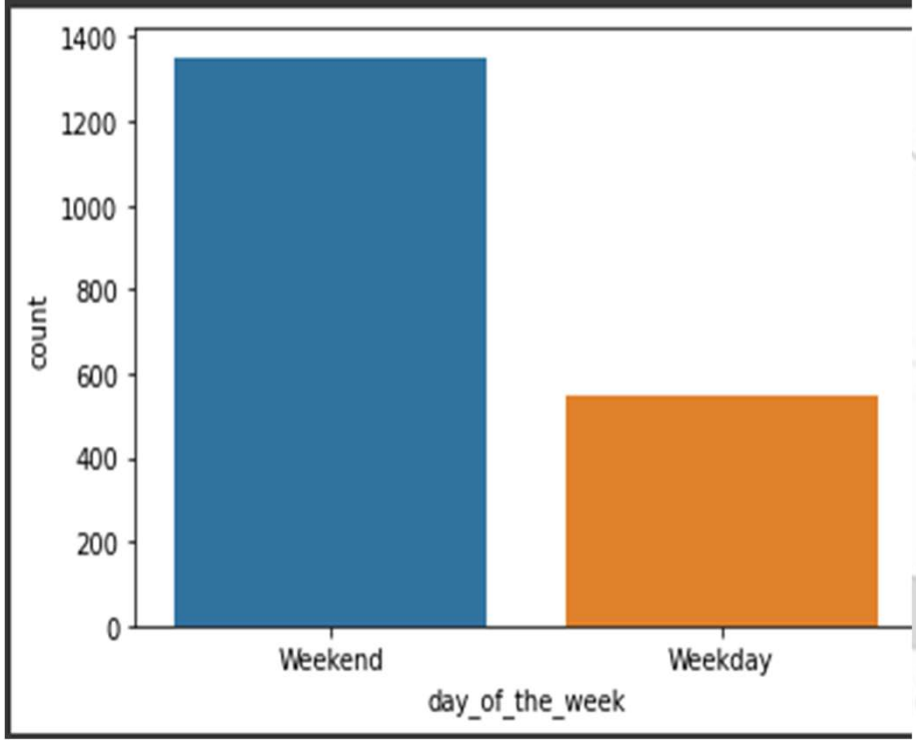
The **boxplot** shows that **50% of cost of orders fall between ~$12 to ~$22.**

The **Median is around /$14.**

The cost of order variable is **slightly right-skewed**

Univariate Analysis- Day of the week

```
sns.countplot(data = df, x = 'day_of_the_week'); #
```




```
# # Check the unique values
df['day_of_the_week'].unique()  ## Complete the code
```

```
array(['Weekend', 'Weekday'], dtype=object)
```

Observations:

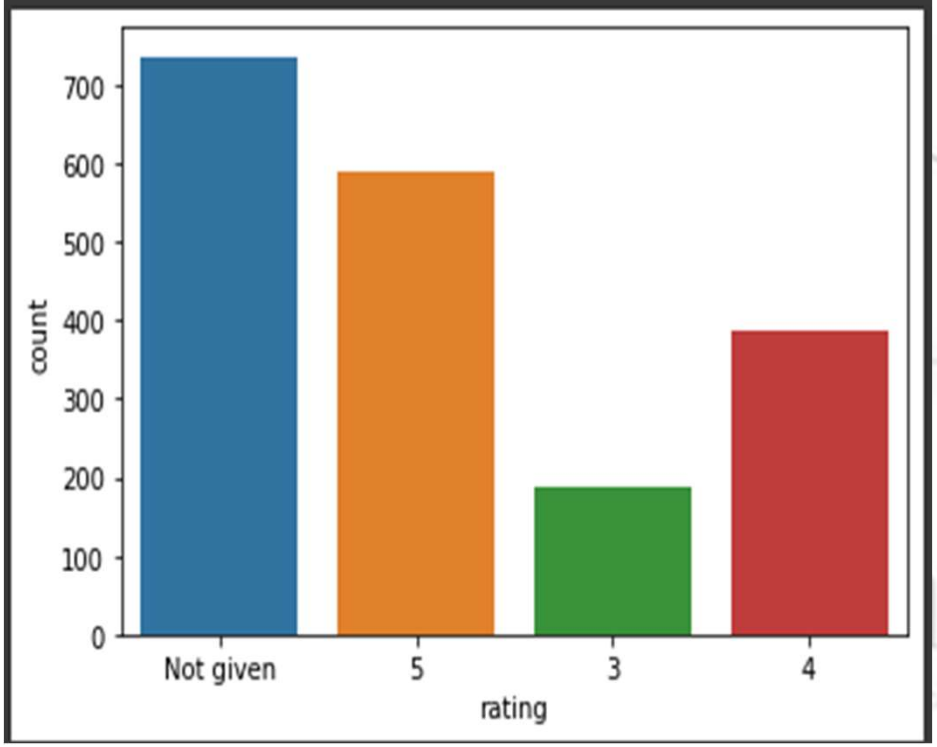
Weekend and weekday are the unique values of day_of_the_week

From the graph, we can say that maximum number of orders are placed on **weekends than weekdays**.

On the **weekends,** between more than **~1300 orders are placed**

Univariate Analysis- Rating

```
sns.countplot(data = df, x = 'rating'); ## Complete
```




```
# Check the unique values
df['rating'].unique() ## Complete the code to check

array(['Not given', '5', '3', '4'], dtype=object)
```

Observations:

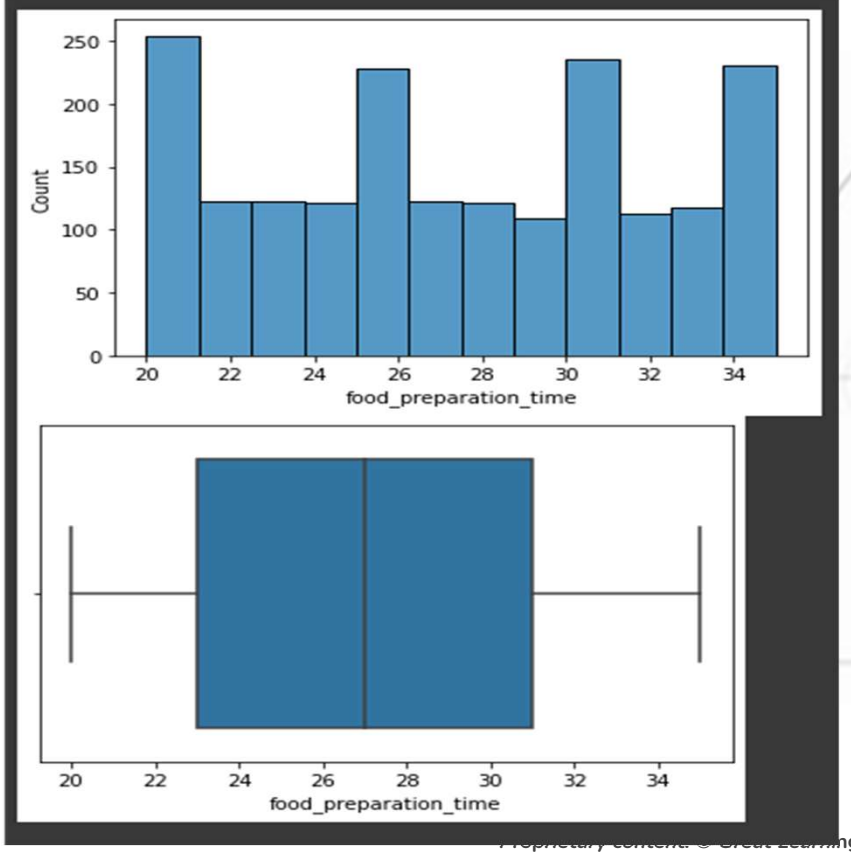
Unique values of rating- 'Not given', '5', '3', '4'.

It can be observed that there are **only 4 unique values** in the column **'rating'.**

From the graph, most customers do not give ratings and hence, **'Not given' is the maximum** counted rating followed by **'5' rating**. It can also be seen that none of the ratings are **below 3**.

Univariate Analysis- Food Preparation Time

```
sns.histplot(data=df,x='food_preparation_time')
plt.show()
sns.boxplot(data=df,x='food_preparation_time')
plt.show()
```




```
sns.histplot(data=df,x='food_preparation_time')
plt.show()
sns.boxplot(data=df,x='food_preparation_time')
plt.show()
```

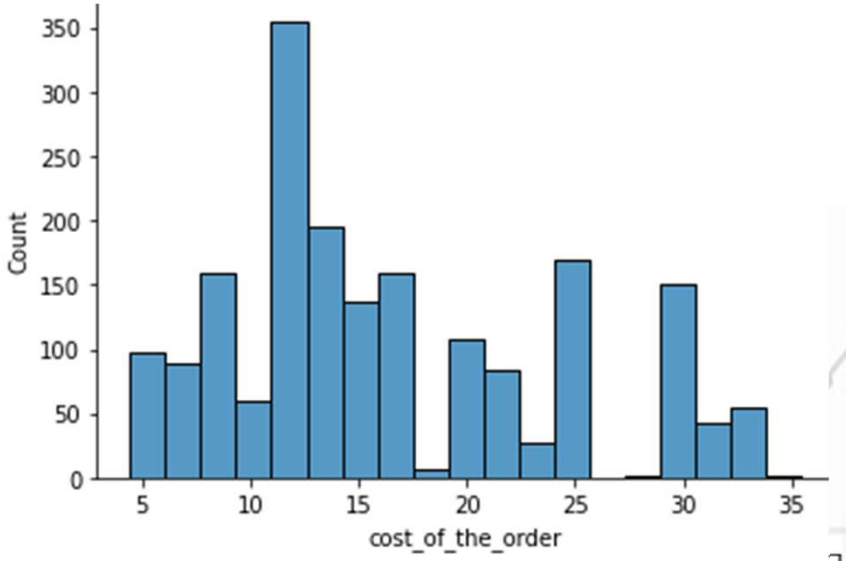

Observations:

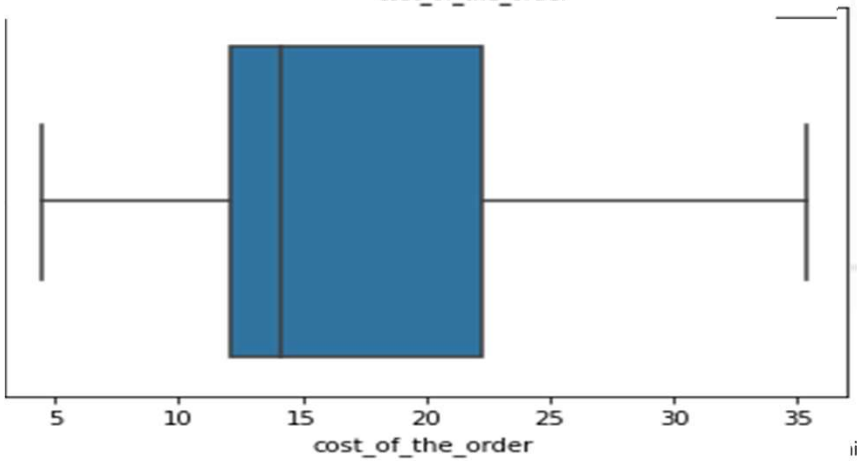
Maximum time taken for **food preparation is between 20-21 minutes.**

The **median time for food preparation** is somewhere around 27 minutes. It seems that **50%** of the data is concentrated **23 and 31 minutes.**

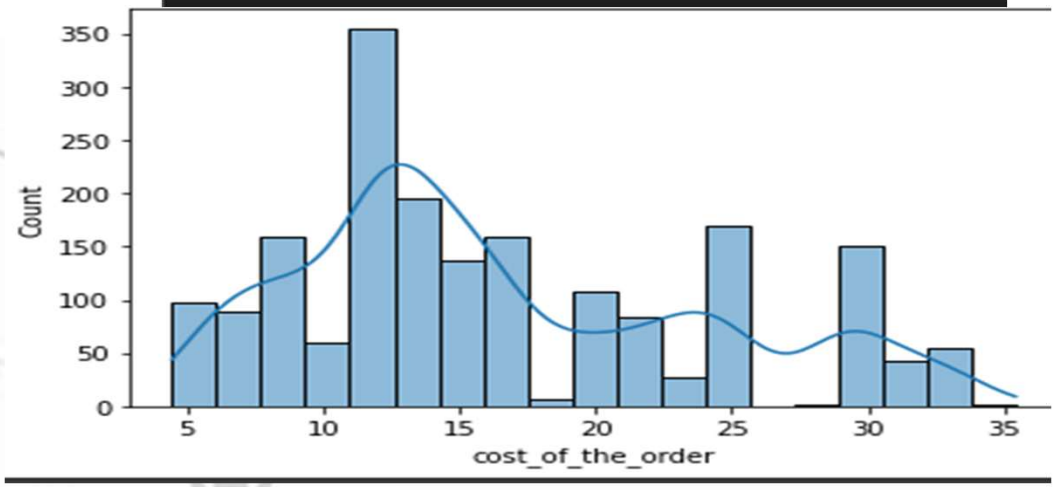
75% of data lies between **20 minutes and ~31 minutes**.

Univariate Analysis- Cost Of Order






```
sns.histplot(data=df,x='cost_of_the_order') ## Comple
plt.show()
sns.boxplot(data=df,x='cost_of_the_order') ## Complet
plt.show()
sns.histplot(data=df,x='cost_of_the_order',kde=True)
plt.show()
```



Observations:

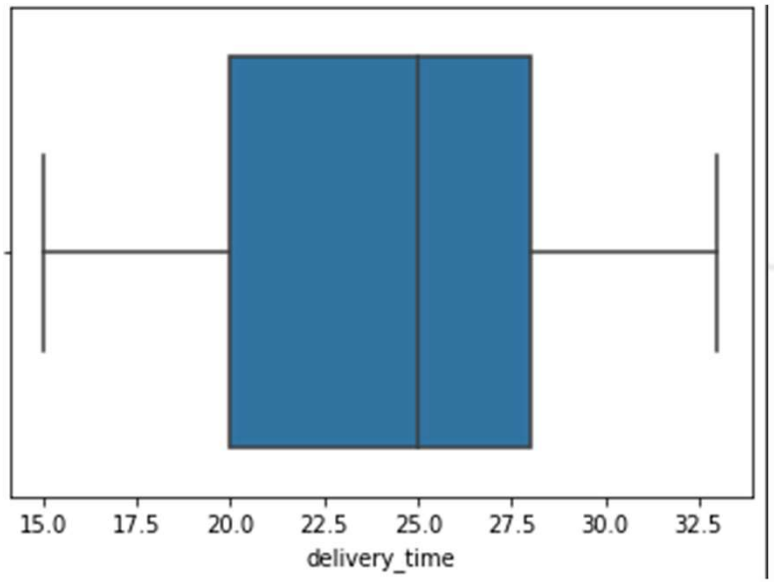
Cost of order variable seems **right-skewed**

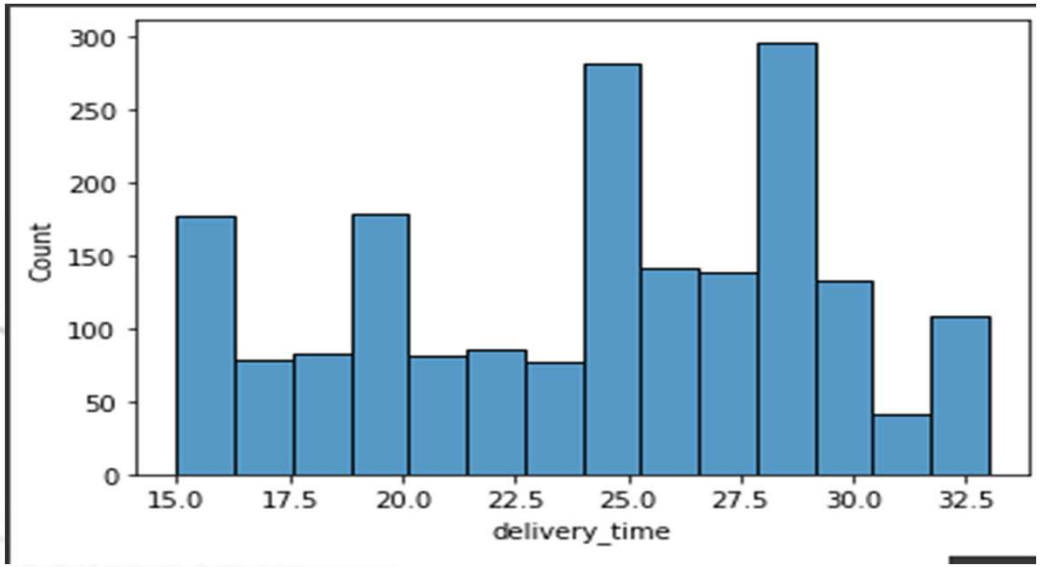
The median is less than $15 and the maximum is $35.

50% of the data lies between ~$14 to ~$23

Univariate Analysis- Delivery time

```
sns.histplot(data=df,x='delivery_time')
plt.show()
sns.boxplot(data=df,x='delivery_time')
plt.show()
```





Observations:

maximum deliveries were **between** times **~27.8 to ~29** minutes

75% of the data lies **below ~27.8 minutes**

Univariate Analysis

Q7. Which are the top 5 restaurants in terms of the number of orders received?

```
df['restaurant_name'].value_counts().head(5)
```

The top 5 restaurants in terms of number of orders received is:

Shake shack = **219 orders**
The Meatball Shop= **132 orders**
Blue ribbon sushi= **119 orders**
Blue ribbon fried chicken= **96 orders**
Parm= **68 orders**

Q9. What percentage of the orders cost more than 20 dollars?

```
# Get orders that cost above 20 dollars
df_greater_than_20 = df[df['cost_of_the_order']>20]

# Calculate the number of total orders where the cost is above 20 dollars
print('The number of total orders that cost above 20 dollars is:', df_greater_than_20.shape[0])

# Calculate percentage of such orders in the dataset
percentage = (df_greater_than_20.shape[0] / df.shape[0]) * 100

print("Percentage of orders above 20 dollars:", round(percentage, 2), '%')
```

Observation:
The number of total orders that cost above 20 dollars is: **555**
Percentage of orders above 20 dollars: **29.24 %**

Univariate Analysis

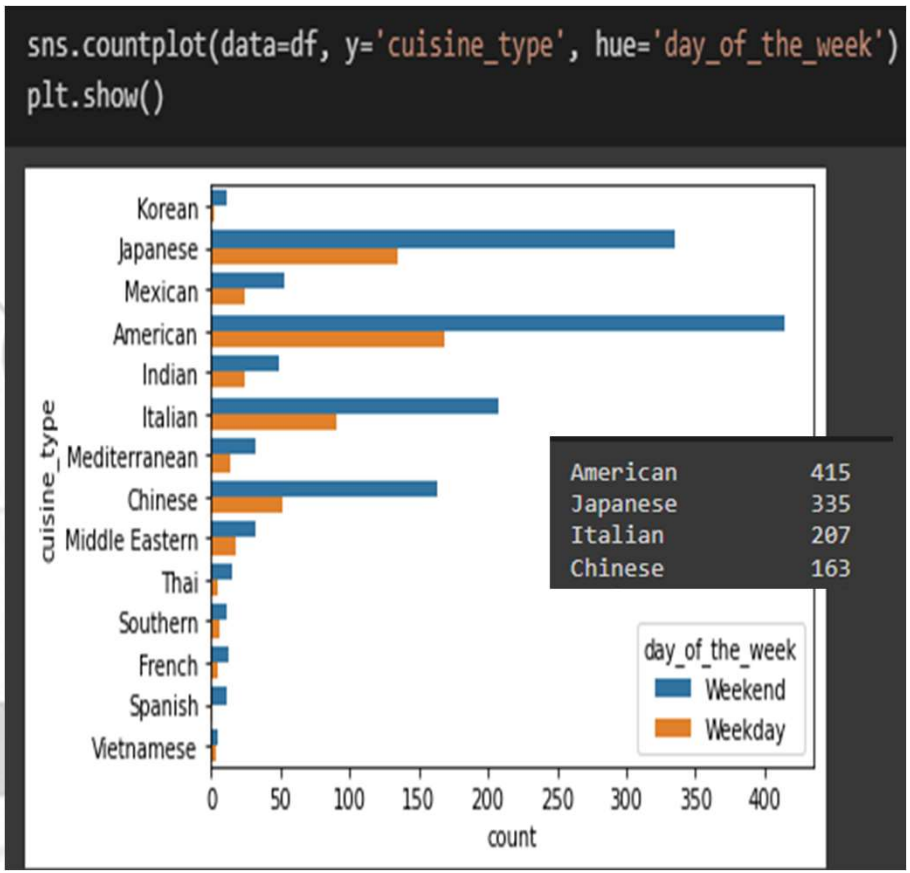
Q8. Which is the most popular cuisine on weekends?

```
# Get most popular cuisine on weekends
df_weekend = df[df['day_of_the_week'] == 'Weekend']
df_weekend['cuisine_type'].unique() ## Complete the co
```

```
array(['Korean', 'Japanese', 'American', 'Italian', 'Mexican',
       'Mediterranean', 'Chinese', 'Indian', 'Thai', 'Southern', 'French',
       'Spanish', 'Middle Eastern', 'Vietnamese'], dtype=object)
```

Observation:
American cuisine is the most popular cuisine type ordered **on weekends**

```
sns.countplot(data=df, y='cuisine_type', hue='day_of_the_week')
plt.show()
```



Univariate Analysis

Q10. What is the mean order delivery time?

```
# Get the mean delivery time
mean_del_time = df['delivery_time'].mean()

print('The mean delivery time for this dataset is', round(mean_del_time, 2), 'minutes')
```

Observation:

The mean delivery time for this dataset is **24.16 minutes.**

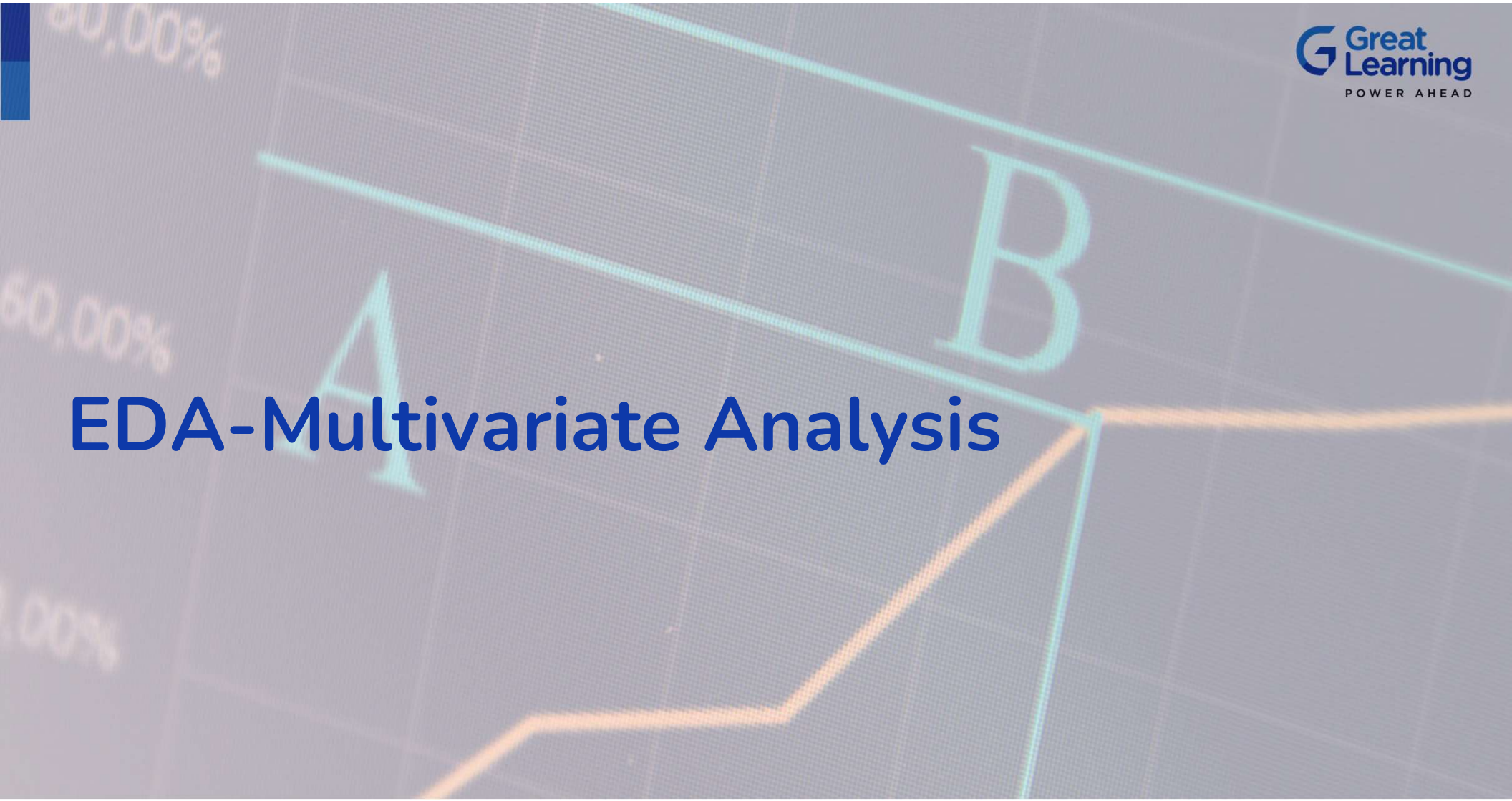
Q11. The company has decided to give 20% discount vouchers to the top 3 most frequent customers. Find the IDs of these customers and the number of orders they placed.

```
# Get the counts of  each customer_id
df['customer_id'].value_counts().head(3)
```

```
52832    13
47440    10
83287     9
Name: customer_id, dtype: int64
```

Observation: Top 3 most frequent customers and number of orders placed:

Customer_id **52832** placed **13** orders
Customer_id **47440** placed **10** orders
Customer_id **83287** placed **9** orders

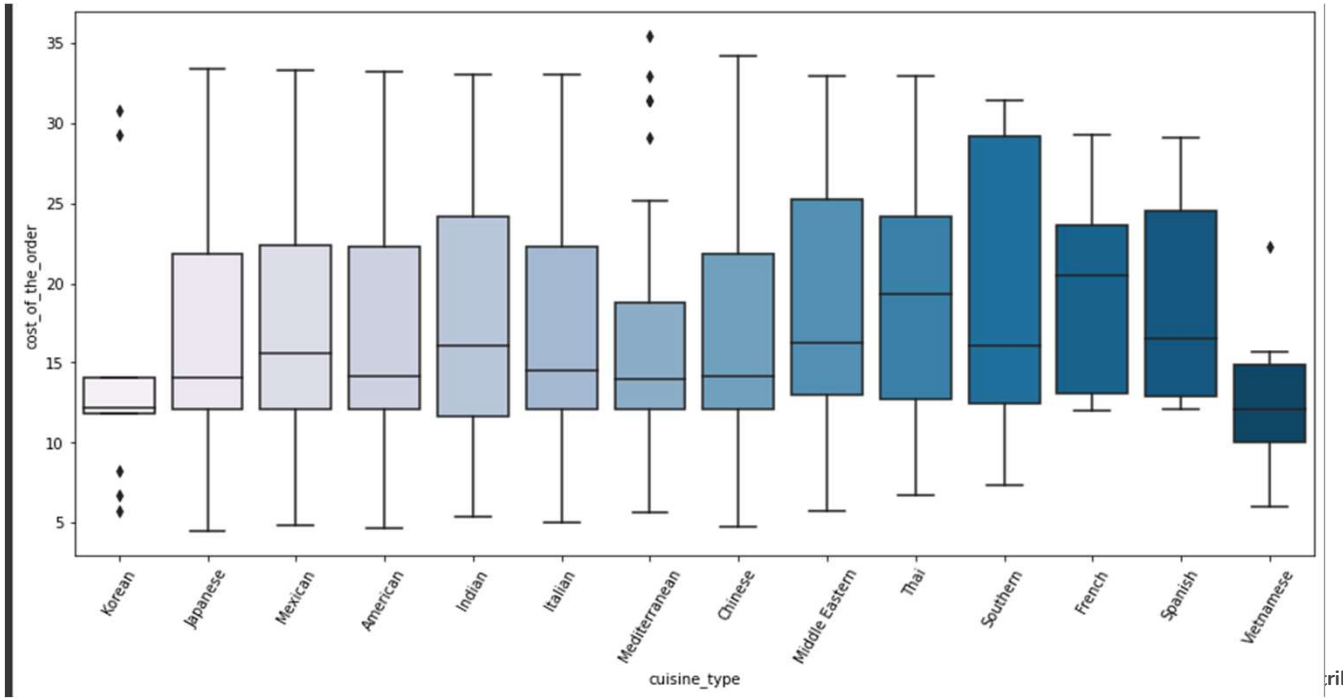


EDA-Multivariate Analysis

Multivariate Analysis- Cuisine vs Cost of the order

Q12. Perform a multivariate analysis to explore relationships between the important variables in the dataset. (It is a good idea to explore relations between numerical variables as well as relations between numerical and categorical variables)

```
# Relationship between cost of the order and cuisine type
plt.figure(figsize=(15,7))
sns.boxplot(x = "cuisine_type", y = "cost_of_the_order", data = df, palette = 'PuBu')
plt.xticks(rotation = 60)
plt.show()
```



Observation:

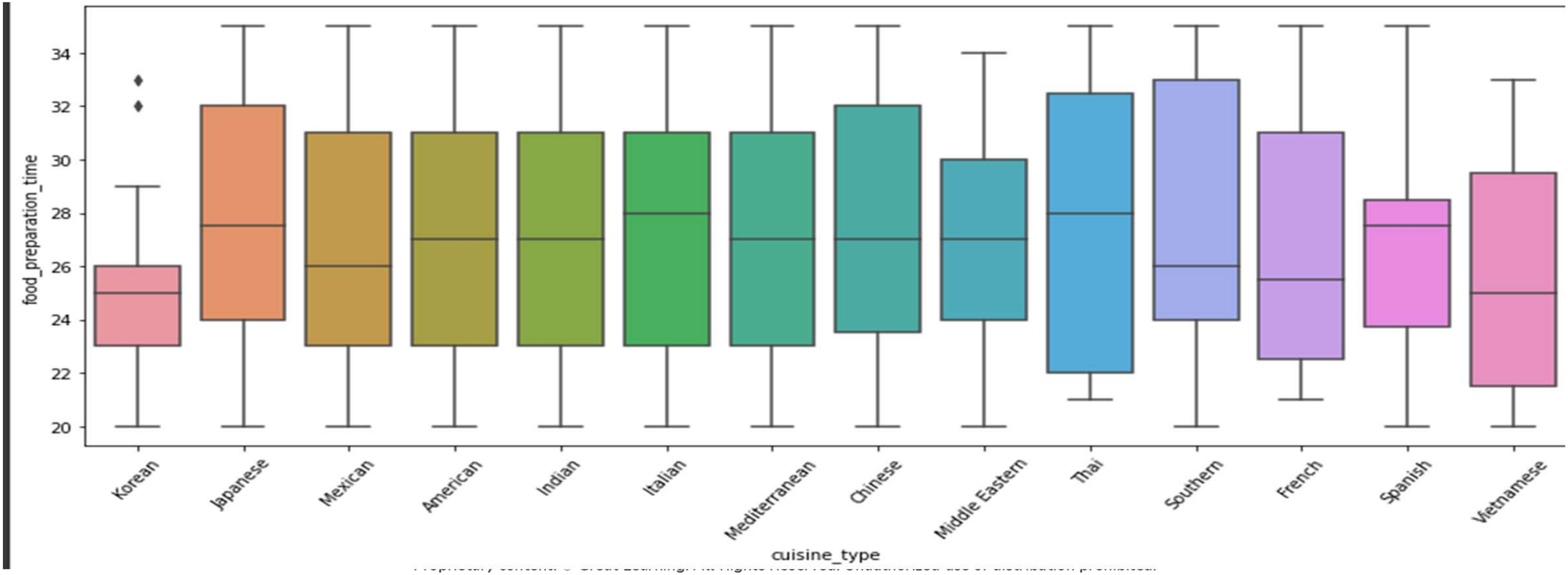
- Cost of order of French cuisine has the highest median ~$21 and Vietnamese cuisine has the lowest Median.
- Korean, Mediterranean and Vietnamese cuisines have cost outliers.

Multivariate Analysis- Cuisine vs Food Preparation time

```
# Relationship between food preparation time and cuisine type
plt.figure(figsize=(15,6))
sns.boxplot(data=df,x = 'cuisine_type', y = 'food_preparation_time')
plt.xticks(rotation = 50)
plt.show()
```

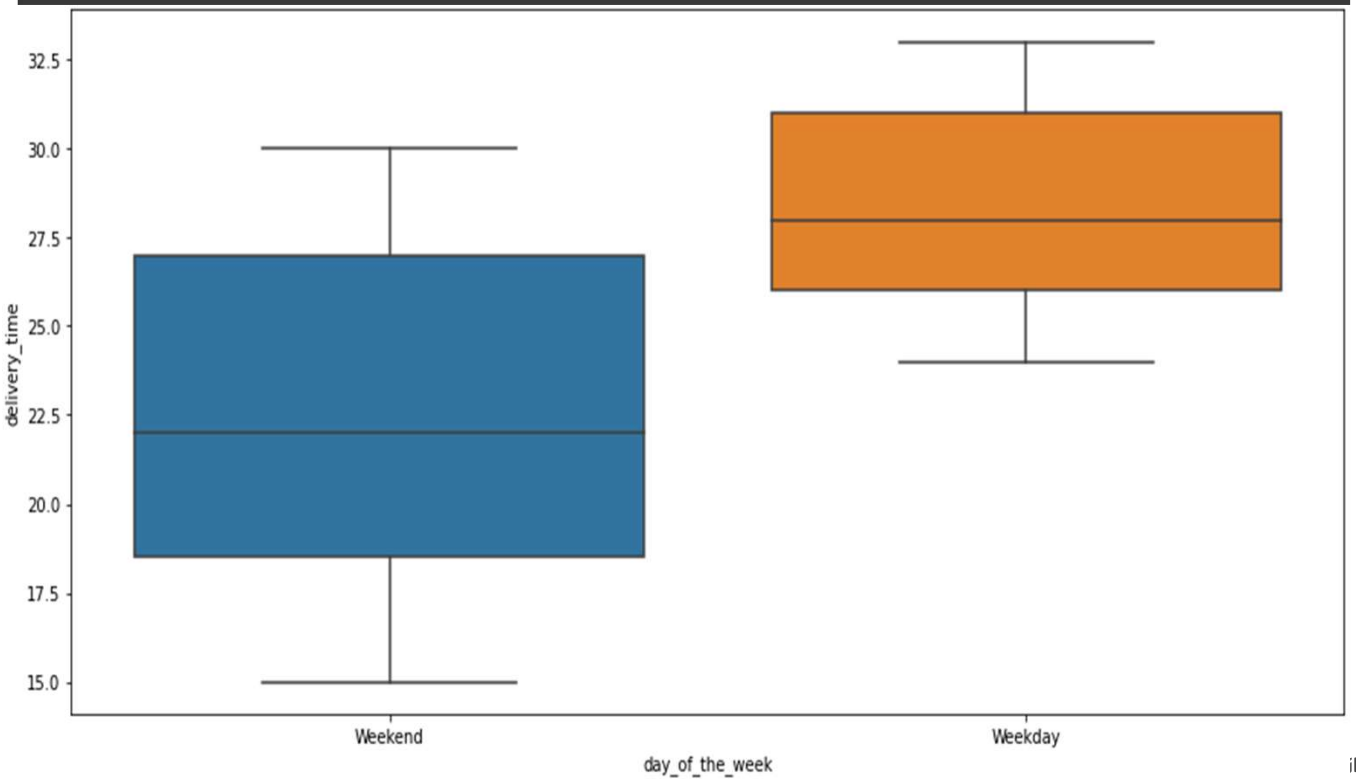
Observation:

Median of food preparation time of Thai and Italian cuisine seems higher than other cuisines.



Multivariate Analysis- Day of the Week vs Delivery time

```
# Relationship between day of the week and delivery time
plt.figure(figsize=(15,7))
sns.boxplot(data=df, x='day_of_the_week', y= 'delivery_time')
plt.show()
```



Observation:

- Median of weekday delivery time is more than a weekend delivery time.
- The RANGE of weekend delivery time also has a wider spread than the weekday delivery time

Multivariate Analysis- Revenue generated by the restaurants

```
Revenue= df.groupby(['restaurant_name'])['cost_of_the_order'].sum().sort_values(ascending=False).head(10)
print(Revenue)
```

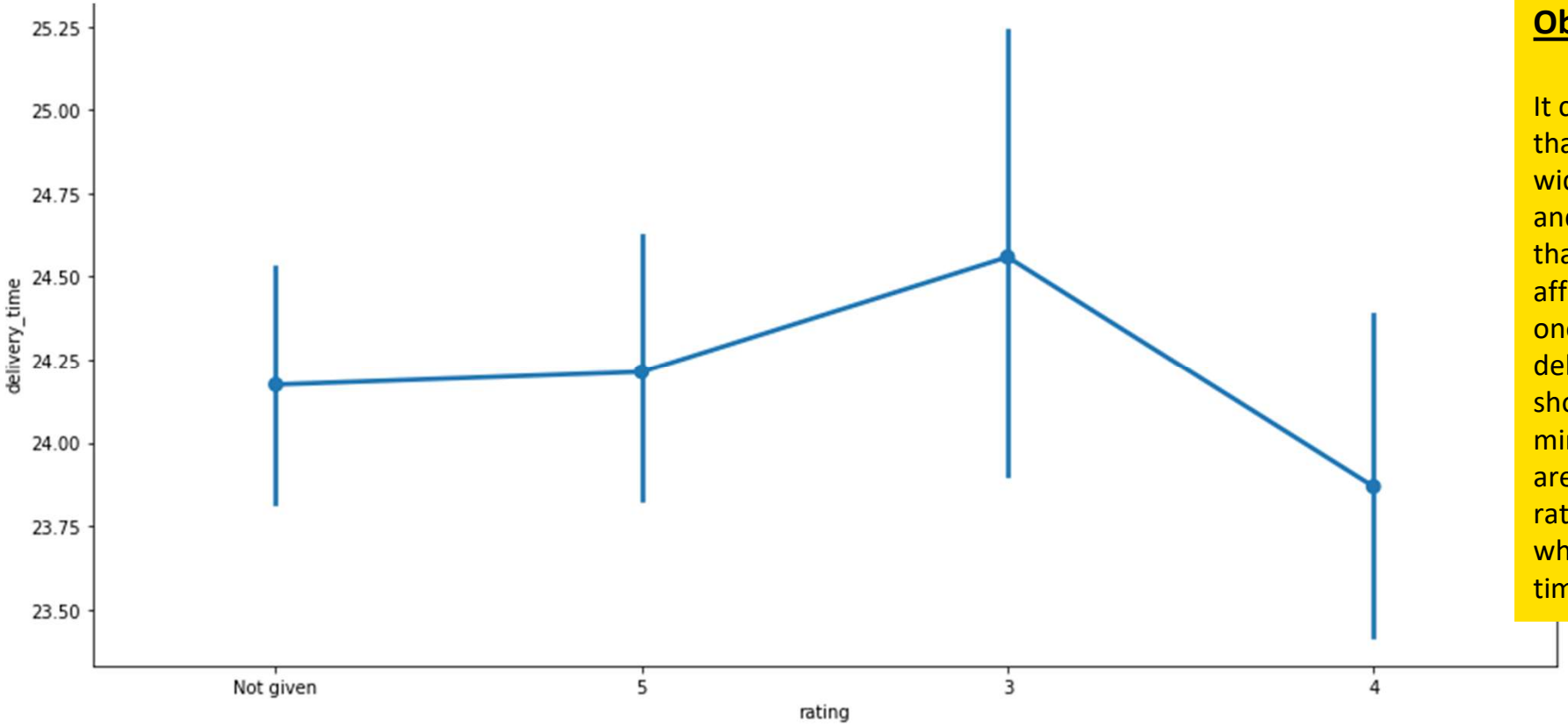
```
restaurant_name
Shake Shack                      3579.53
The Meatball Shop                2145.21
Blue Ribbon Sushi                1903.95
Blue Ribbon Fried Chicken        1662.29
Parm                             1112.76
RedFarm Broadway                  965.13
RedFarm Hudson                    921.21
TAO                               834.50
Han Dynasty                       755.29
Blue Ribbon Sushi Bar & Grill     666.62
Name: cost_of_the_order, dtype: float64
```

Observation:

Shake Shack generates the highest revenue, followed by **The Meatball Shop amongst the top 10 revenue generators.**

Multivariate Analysis- Rating vs Delivery time

```
# Relationship between rating and delivery time
plt.figure(figsize=(15, 7))
sns.pointplot(x = 'rating', y = 'delivery_time', data = df)
plt.show()
```

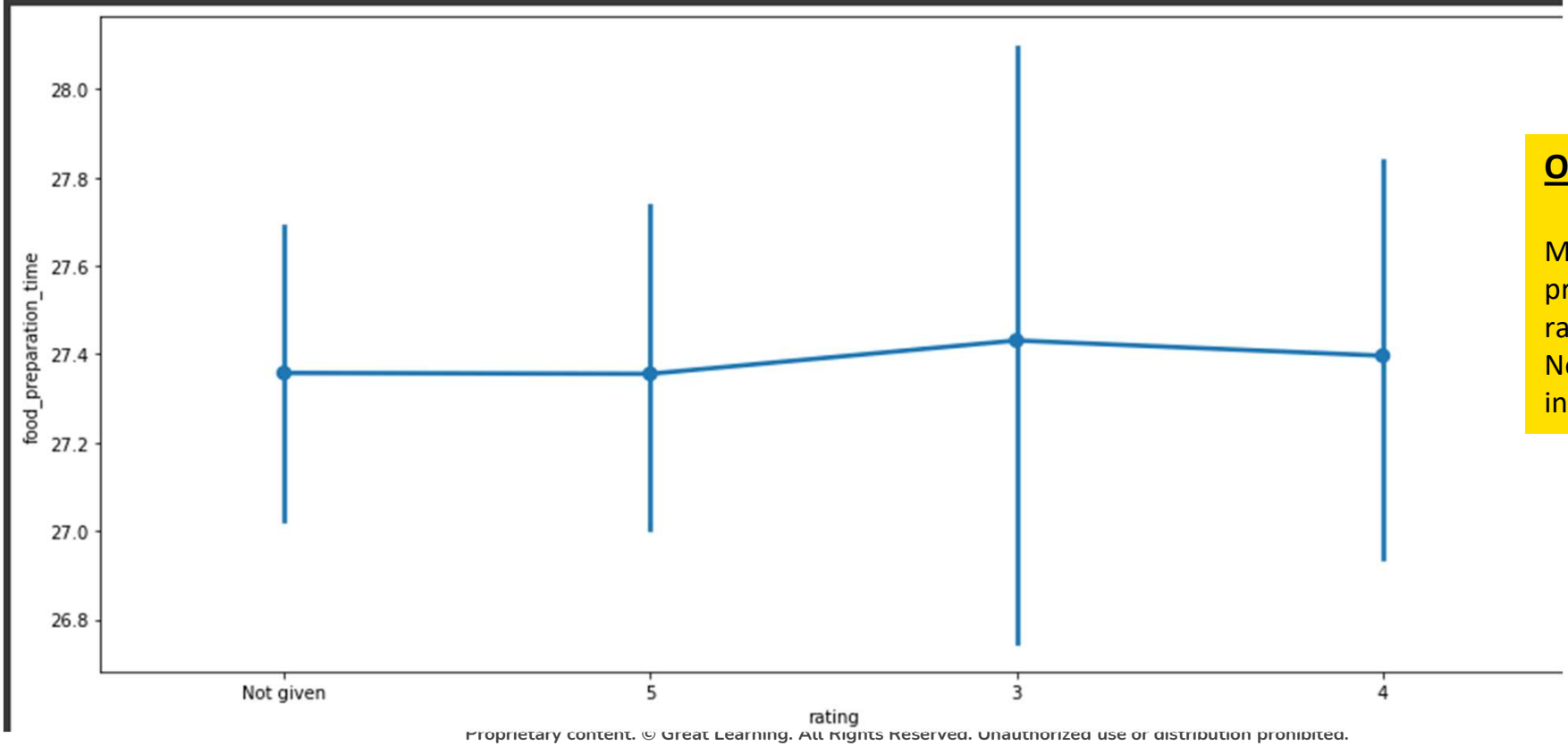


Observation:

It can be observed that rating 3 has the widest range of data and can also be seen that delivery time affects the ratings. On one side when the delivery timings are shorter than 24 minutes, the ratings are higher, and the ratings are lower when the delivery time is more

Multivariate Analysis- Rating vs Food preparation time

```
# Relationship between rating and food preparation time
plt.figure(figsize=(15, 7))
sns.pointplot(data=df, x='rating',y='food_preparation_time')
plt.show()
```

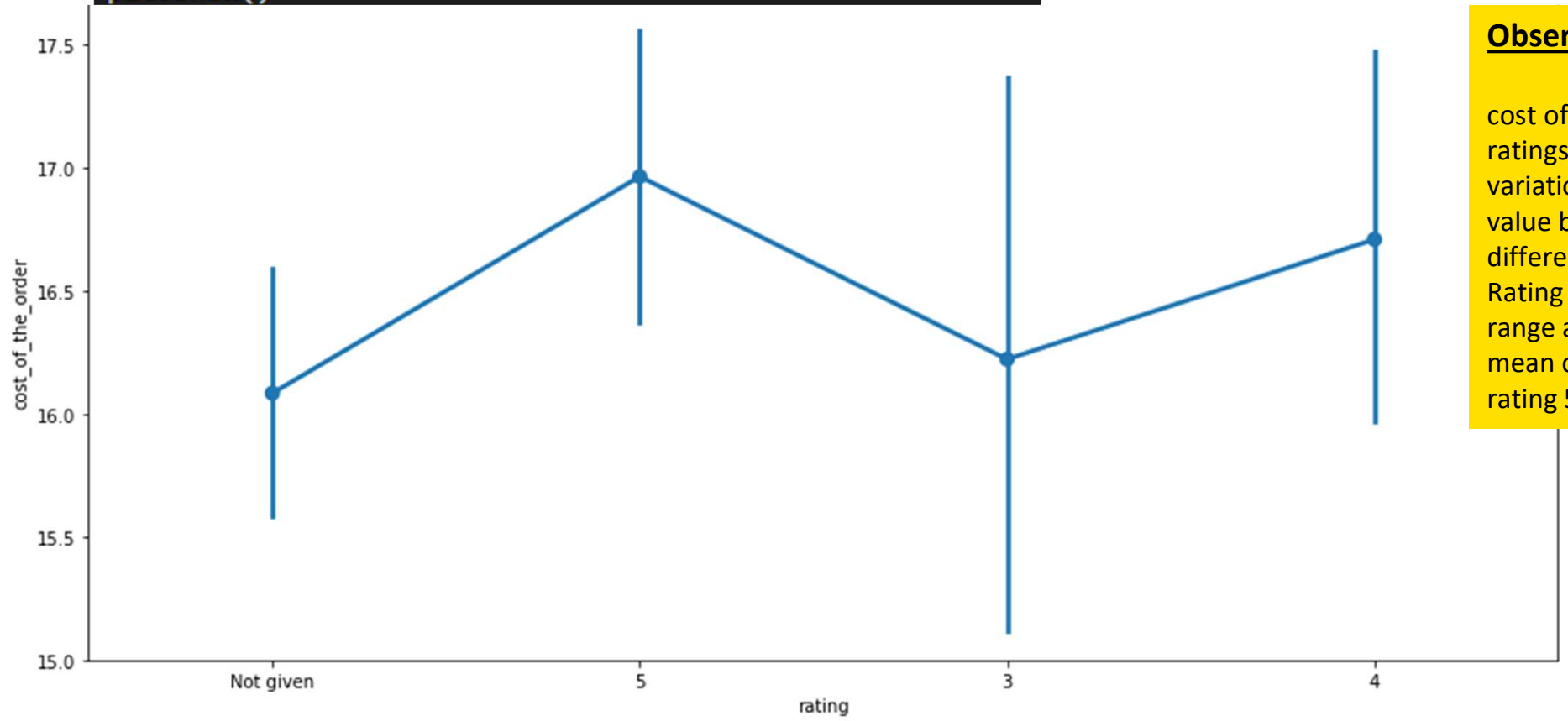


Observation:

Most of the food preparation time ratings are 3.
Not much difference in the means

Multivariate Analysis- Rating vs Cost of the order

```
# Relationship between rating and cost of the order
plt.figure(figsize=(15, 7))
sns.pointplot(data=df,x='rating',y='cost_of_the_order')
plt.show()
```

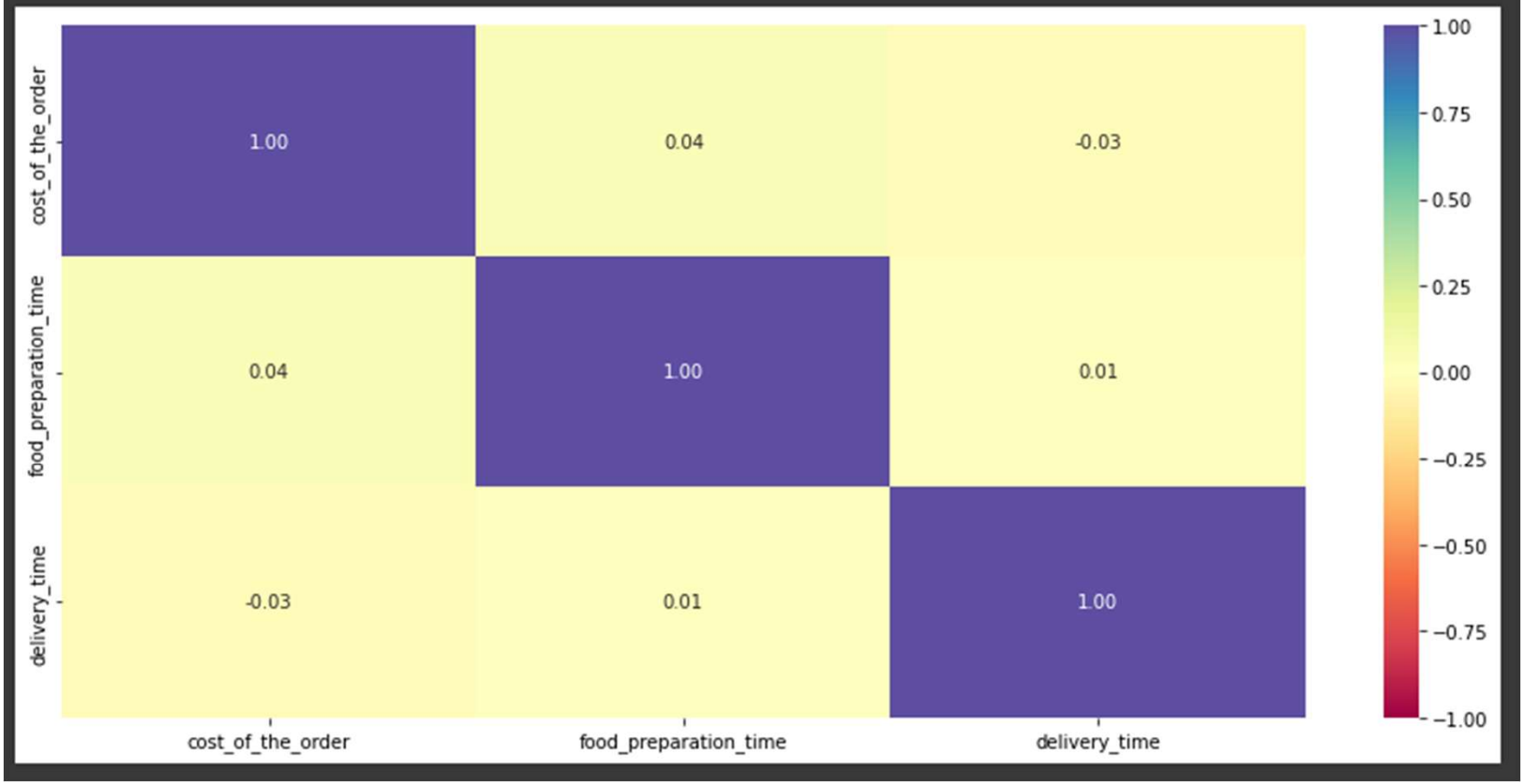


Observation:

cost of order vs ratings show huge variations in mean value between different ratings. Rating 3 has a larger range and lower mean compared to rating 5 and 4.

Multivariate Analysis- Correlation among variables

```
col_list = ['cost_of_the_order', 'food_preparation_time', 'delivery_time']
plt.figure(figsize=(15, 7))
sns.heatmap(df[col_list].corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral")
plt.show()
```



Observation:

Food preparation time has a slightly more positive correlation with cost of order than delivery time. Cost of order has a very weak association with the food preparation time but greater than the association between delivery time and food preparation time.

Q13. **The company wants to provide a promotional offer in the advertisement of the restaurants. The condition to get the offer is that the restaurants must have a rating count of more than 50 and the average rating should be greater than 4. Find the restaurants fulfilling the criteria to get the promotional offer.**

```
# Filter the rated restaurants
df_rated = df[df['rating'] != 'Not given'].copy()

# Convert rating column from object to integer
df_rated['rating'] = df_rated['rating'].astype('int')

# Create a dataframe that contains the restaurant names with their rating counts
df_rating_count = df_rated.groupby(['restaurant_name'])['rating'].count().sort_values(ascending = False).reset_index()
df_rating_count.head()
```

	restaurant_name	rating
0	Shake Shack	133
1	The Meatball Shop	84
2	Blue Ribbon Sushi	73
3	Blue Ribbon Fried Chicken	64
4	RedFarm Broadway	41

```
# Get the restaurant names that have rating count more than 50
rest_names = df_rating_count[df_rating_count['rating']>50]['restaurant_name'] ## Complete the code to get the restau

# Filter to get the data of restaurants that have rating count more than 50
df_mean_4 = df_rated[df_rated['restaurant_name'].isin(rest_names)].copy()

# Group the restaurant names with their ratings and find the mean rating of each restaurant
df_mean_4.groupby(['restaurant_name'])['rating'].mean().sort_values(ascending = False).reset_index().dropna() ## Com
```

	restaurant_name	rating
0	The Meatball Shop	4.511905
1	Blue Ribbon Fried Chicken	4.328125
2	Shake Shack	4.278195
3	Blue Ribbon Sushi	4.219178

Observation:

The restaurants are :

1.The meatball shop=4.5 avg

2. Blue Ribbon Fried Chicken= 4.3

3. Shake Shack=4.27

4. Blue Ribbon Sushi= 4.21

Q14. The company charges the restaurant 25% on the orders having cost greater than 20 dollars and 15% on the orders having cost greater than 5 dollars. Find the net revenue generated by the company across all orders.

```
# get the total revenue and print it
total_rev = df['Revenue'].sum() ## Write the appropriate function to get the t
print('The net revenue is around', round(total_rev, 2), 'dollars')
```

```
The net revenue is around 6166.3 dollars
```

Observation:

The net revenue is around **6166.3 dollars.**

Q15. The company wants to analyze the total time required to deliver the food. What percentage of orders take more than 60 minutes to get delivered from the time the order is placed? (The food has to be prepared and then delivered.)

```
# Calculate total delivery time and add a new column to the dataframe df to store the total deliver
df['total_time'] = df['food_preparation_time'] + df['delivery_time']

## Write the code below to find the percentage of orders that have more than 60 minutes of total de

total_of_orders= df[df['total_time']>60]
percentage_of_orders= (total_of_orders.shape[0]/df.shape[0])*100
print('Percentage of orders with more than 60 minutes of total delivery time is:', round(percentage_of_orders,2),'%')
```

Observation:

Percentage of orders with more than 60 minutes of total delivery time is: **10.54 %**

Q16. The company wants to analyze the delivery time of the orders on weekdays and weekends. How does the mean delivery time vary during weekdays and weekends?

```
# Get the mean delivery time on weekdays and print it
print('The mean delivery time on weekdays is around',
      round(df[df['day_of_the_week'] == 'Weekday']['delivery_time'].mean()),
     'minutes')


## Write the code below to get the mean delivery time on weekends and print it

print('The mean delivery time on weekends is around',
      round(df[df['day_of_the_week'] == 'Weekend']['delivery_time'].mean()),
     'minutes')
```

Observation:

The mean delivery time on weekdays is around **28 minutes**
The mean delivery time on weekends is around **22 minutes**

APPENDIX

Microsoft Excel
ma Separated Val

Food hub data

C:\Users\shikh\
nloads\PYF_Projec

Python low code version

Great Learning

Happy Learning !