Project Python Foundations: FoodHub Data Analysis

Marks: 60

Context

The number of restaurants in New York is increasing day by day. Lots of students and busy professionals rely on those restaurants due to their hectic lifestyles. Online food delivery service is a great option for them. It provides them with good food from their favorite restaurants. A food aggregator company FoodHub offers access to multiple restaurants through a single smartphone app.

The app allows the restaurants to receive a direct online order from a customer. The app assigns a delivery person from the company to pick up the order after it is confirmed by the restaurant. The delivery person then uses the map to reach the restaurant and waits for the food package. Once the food package is handed over to the delivery person, he/she confirms the pick-up in the app and travels to the customer's location to deliver the food. The delivery person confirms the drop-off in the app after delivering the food package to the customer. The customer can rate the order in the app. The food aggregator earns money by collecting a fixed margin of the delivery order from the restaurants.

Objective

The food aggregator company has stored the data of the different orders made by the registered customers in their online portal. They want to analyze the data to get a fair idea about the demand of different restaurants which will help them in enhancing their customer experience. Suppose you are hired as a Data Scientist in this company and the Data Science team has shared some of the key questions that need to be answered. Perform the data analysis to find answers to these questions that will help the company to improve the business.

Data Description

The data contains the different data related to a food order. The detailed data dictionary is given below.

Data Dictionary

- order_id: Unique ID of the order
- customer_id: ID of the customer who ordered the food
- restaurant_name: Name of the restaurant
- cuisine_type: Cuisine ordered by the customer
- cost: Cost of the order
- day_of_the_week: Indicates whether the order is placed on a weekday or weekend (The weekday is from Monday to Friday and the weekend is Saturday and Sunday)
- rating: Rating given by the customer out of 5
- food_preparation_time: Time (in minutes) taken by the restaurant to prepare the food. This is calculated by taking the difference between the timestamps of the restaurant's order confirmation and the delivery person's pick-up confirmation.
- delivery_time: Time (in minutes) taken by the delivery person to deliver the food package. This is calculated by taking the difference between the timestamps of the delivery person's pick-up confirmation and drop-off information

Let us start by importing the required libraries

In [ ]:
```
# import libraries for data manipulation
import numpy as np
import pandas as pd

# import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns


%matplotlib inline
```

Understanding the structure of the data

In [ ]:
```
# uncomment and run the following lines for Google Colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", for
ce_remount=True).
```

In [ ]:
```
# read the data
df = pd.read_csv('/content/drive/MyDrive/foodhub_order.csv')
```

```
# returns the first 5 rows
df.head()
```

Out[ ]:

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	delivery_time
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	25	20
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	25	23
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5	23	28
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3	25	15
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4	25	24

Observations:

The DataFrame has 9 columns as mentioned in the Data Dictionary. Data in each row corresponds to the order placed by a customer.

Question 1: How many rows and columns are present in the data? [0.5 mark]

In [ ]:
```
# Write your code here
df.shape
```

Out[ ]:
```
(1898, 9)
```

Observations:

The data has 1898 rows and 9 columns

Question 2: What are the datatypes of the different columns in the dataset? (The info() function can be used) [0.5 mark]

In [ ]:
```
# Use info() to print a concise summary of the DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1898 entries, 0 to 1897
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   order_id               1898 non-null   int64  
 1   customer_id            1898 non-null   int64  
 2   restaurant_name        1898 non-null   object 
 3   cuisine_type           1898 non-null   object 
 4   cost_of_the_order      1898 non-null   float64
 5   day_of_the_week        1898 non-null   object 
 6   rating                 1898 non-null   object 
 7   food_preparation_time  1898 non-null   int64  
 8   delivery_time          1898 non-null   int64  
dtypes: float64(1), int64(4), object(4)
memory usage: 133.6+ KB
```

Observations:

There are attibutes of different types - int, float, and object.

Question 3: Are there any missing values in the data? If yes, treat them using an appropriate method. [1 mark]

In [ ]:
```
# Write your code here
df.isnull().sum()
```

Out[ ]:
```
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
```

```
   # This is formatted as code
```

Observations: There are no missing values.

Question 4: 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? [2 marks]

In [ ]:
```
# Write your code here
df.describe(include = 'all').T
```

Out[ ]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	ma
order_id	1898.0	NaN	NaN	NaN	1477495.5	548.049724	1476547.0	1477021.25	1477495.5	1477969.75	1478444
customer_id	1898.0	NaN	NaN	NaN	171168.478398	113698.139743	1311.0	77787.75	128600.0	270525.0	405334
restaurant_name	1898	178	Shake Shack	219	NaN	NaN	NaN	NaN	NaN	NaN	Na
cuisine_type	1898	14	American	584	NaN	NaN	NaN	NaN	NaN	NaN	Na
cost_of_the_order	1898.0	NaN	NaN	NaN	16.498851	7.483812	4.47	12.08	14.14	22.2975	35.4
day_of_the_week	1898	2	Weekend	1351	NaN	NaN	NaN	NaN	NaN	NaN	Na
rating	1898	4	Not given	736	NaN	NaN	NaN	NaN	NaN	NaN	Na
food_preparation_time	1898.0	NaN	NaN	NaN	27.37197	4.632481	20.0	23.0	27.0	31.0	35
delivery_time	1898.0	NaN	NaN	NaN	24.161749	4.972637	15.0	20.0	25.0	28.0	33

Observations:

Food preparation time in minutes= Minimum: 20, Maximum: 35, Average: 27.37.

Question 5: How many orders are not rated? [1 mark]

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

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

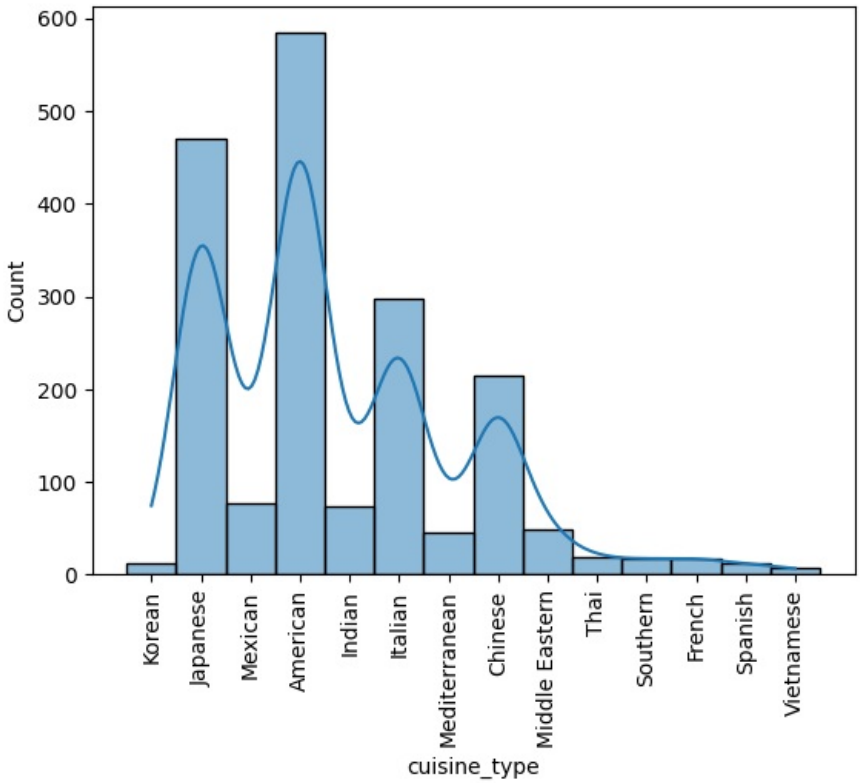
Observations: 736 orders are not rated

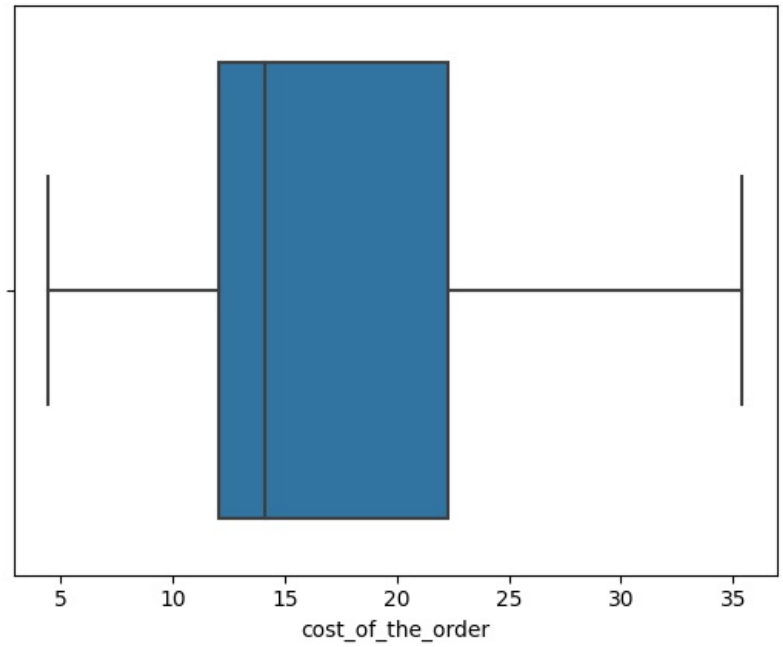
Exploratory Data Analysis (EDA)

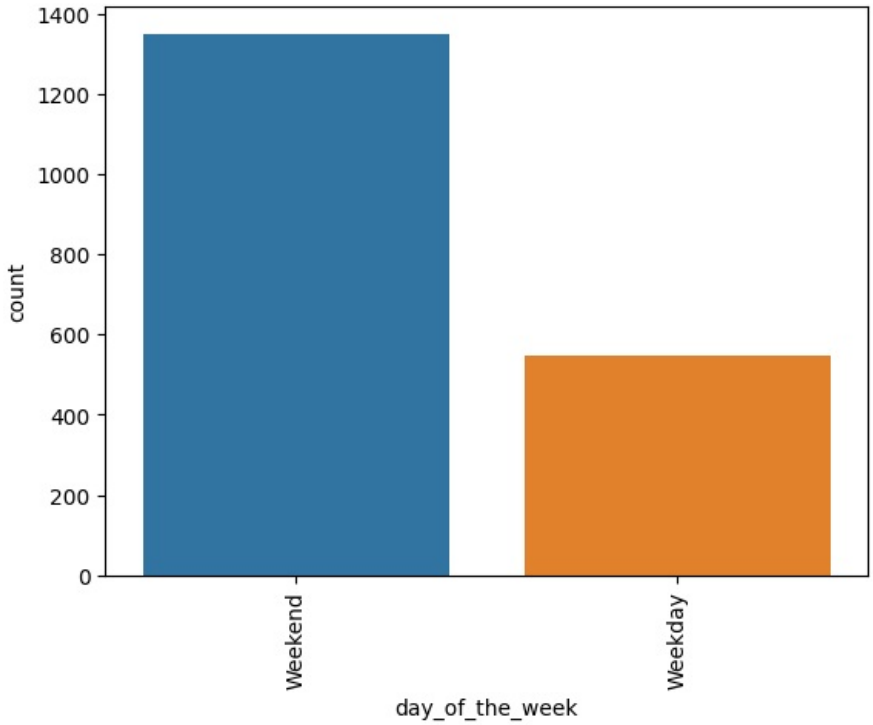
Univariate Analysis

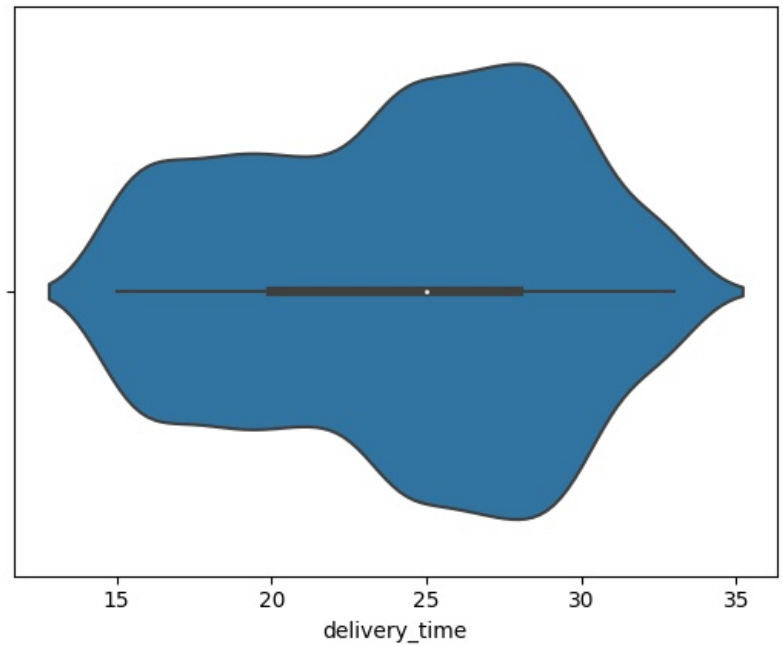
Question 6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.) [9 marks]

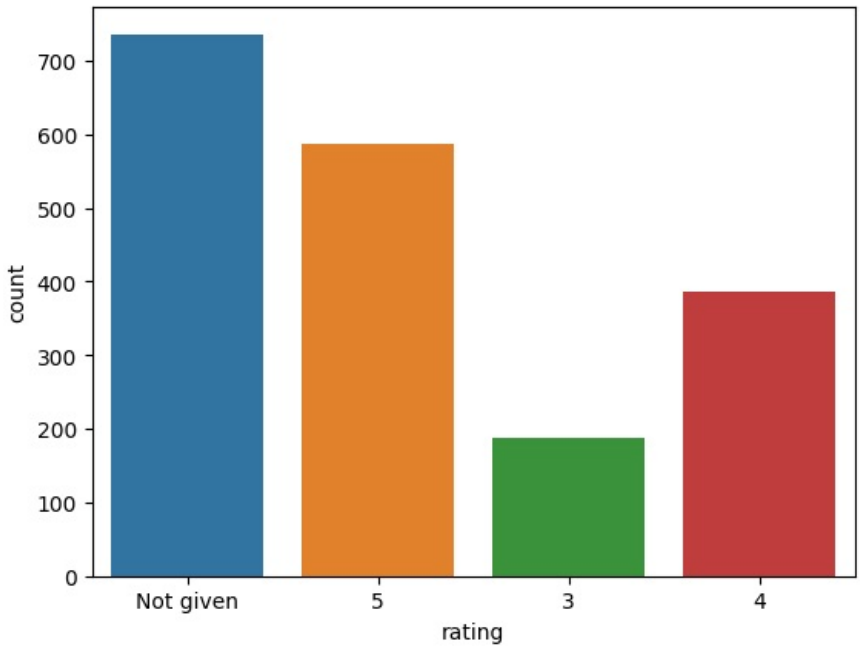
In [ ]:
```
# Write the code here

sns.histplot(data = df, x = 'cuisine_type', kde = True)
plt.xticks(rotation = 90);
plt.show()
sns.boxplot(df, x = 'cost_of_the_order')
plt.show()
sns.countplot(data = df, x = 'day_of_the_week')
plt.xticks(rotation = 90);
plt.show()
sns.violinplot(data = df, x = 'delivery_time')
plt.show()
sns.countplot(data = df, x = 'rating')
plt.show()
plt.show()
sns.countplot(data = df, x = 'food_preparation_time')
plt.show()
sns.histplot(data = df, x = 'delivery_time', kde = True)
plt.show()
plt.figure(figsize = (20,7))
```

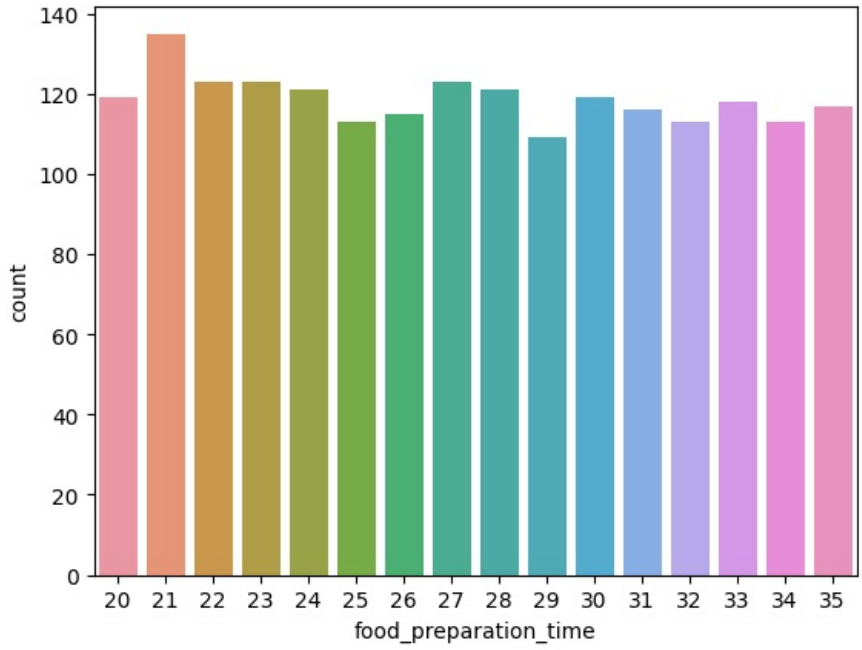

600
500
400
300
200
100
0
Count
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese
cuisine_type

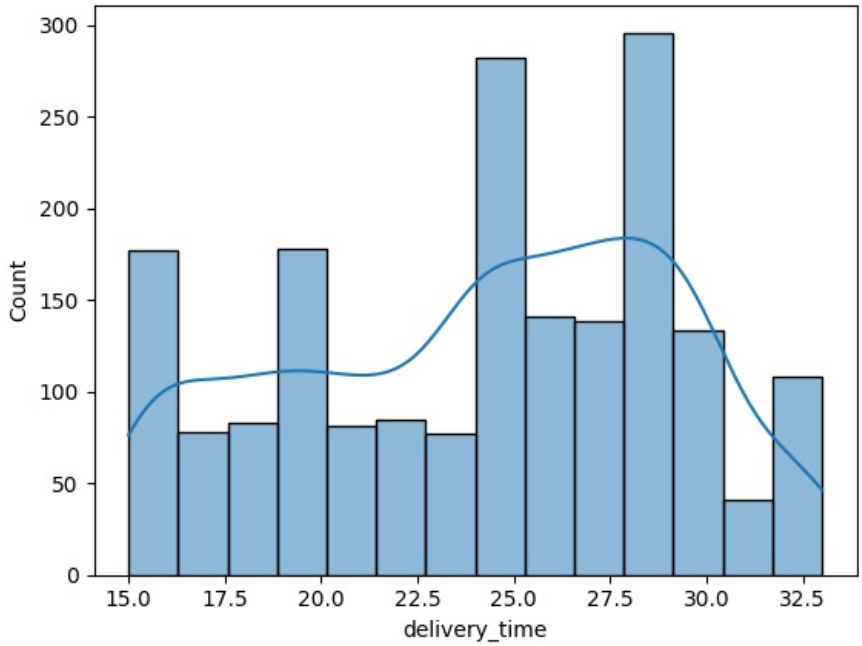

5
10
15
20
25
30
35
cost_of_the_order


1400
1200
1000
800
600
400
200
0
count
Weekend
Weekday
day_of_the_week


15
20
25
30
35
delivery_time


700
600
500
400
300
200
100
0
count
Not given
5
3
4
rating





Out[ ]:
```
<Figure size 2000x700 with 0 Axes>
```


```
<Figure size 2000x700 with 0 Axes>
```


Obervations:

- The highest number of orders are placed in American cuisines
- The mean cost of order is around $14.
- Most orders are placed between $12 to 23$.
- The number of oders that are not rated is higher than number of order with any other ratings.
- The graph of food preparation time looks more or less consistant.
- The delivery time of most of the orders is 38 mins.

Question 7: Which are the top 5 restaurants in terms of the number of orders received? [1 mark]

In [ ]:
```
# Write the code here
df['restaurant_name'].value_counts().head(5)
```

Out[ ]:
```
Shake Shack                  219
The Meatball Shop            132
Blue Ribbon Sushi            119
Blue Ribbon Fried Chicken     96
Parm                          68
Name: restaurant_name, dtype: int64
```

Observations: The top 5 restuarants with highest number of orders received are Shake Shack, The Meatball

Shop, Blue Ribbon Sushi, Blue Ribbon Fried Chicken, and Parm. The American restuarant Shake Shack comes in the top.

Question 8: Which is the most popular cuisine on weekends? [1 mark]

> Indented block

In [ ]:
```
# Write the code here
df_cuisine = df [["cuisine_type", "day_of_the_week"]].copy()
pop_cuisine = df_cuisine[df_cuisine["day_of_the_week"]== "Weekend"]
print (pop_cuisine["cuisine_type"].value_counts())
```

```
American          415
Japanese          335
Italian           207
Chinese           163
Mexican            53
Indian             49
Mediterranean      32
Middle Eastern     32
Thai               15
French             13
Korean             11
Southern           11
Spanish            11
Vietnamese          4
Name: cuisine_type, dtype: int64
```

Observations: The most popular cuisine in weekends is American, second is Japanese, and third comes Italian.

Question 9: What percentage of the orders cost more than 20 dollars? [2 marks]

In [ ]:
```
# Write the code here
orders =len(df[(df["cost_of_the_order"] > 20)])
orders_20 = len(df[(df["cost_of_the_order"] > 20)])/len(df)*100
print("Percentage of orders above $20 =", orders_20,'%')
```

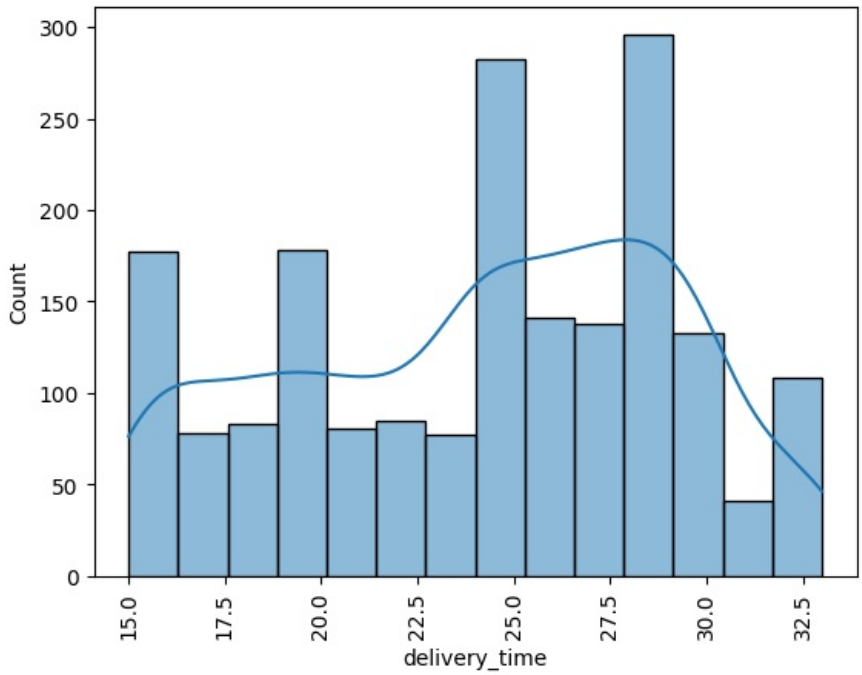
```
Percentage of orders above $20 = 29.24130663856691 %
```

Observations: 29.24% of the total number of orders places are above $20.

Question 10: What is the mean order delivery time? [1 mark]

In [ ]:
```
# Write the code here
time = df["delivery_time"].mean()
print("Delivery time average:", time)
sns.histplot(data = df, x = 'delivery_time', kde = True)
plt.xticks(rotation = 90);
plt.show()
```

```
Delivery time average: 24.161749209694417
```



Observations: The mean delivery time is 24.16 mins.

Question 11: 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. [1 mark]

In [ ]:
```
# Write the code here
reg_customer = df [["customer_id"]].copy()
print (reg_customer["customer_id"].value_counts())
```

```
52832     13
47440     10
83287      9
250494     8
259341     7
          ..
385426     1
254913     1
289597     1
74412      1
397537     1
Name: customer_id, Length: 1200, dtype: int64
```

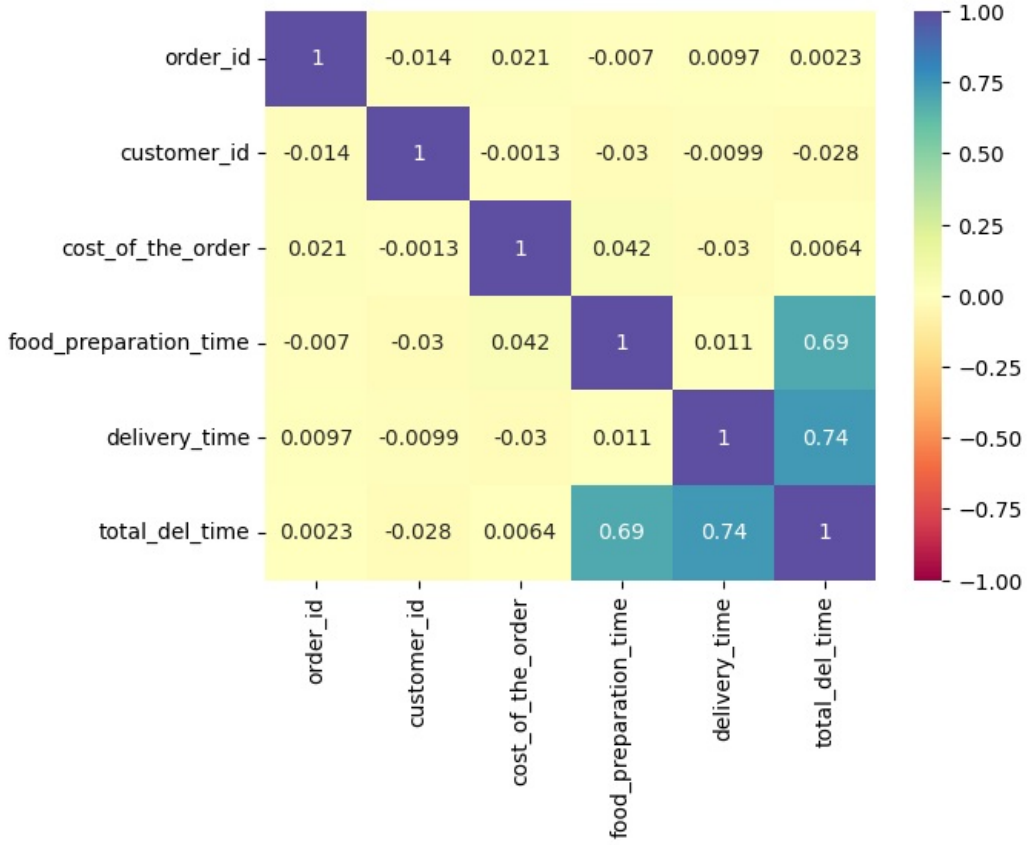
Observations: The top 3 frequent customers are with customer ID 52832, 47440, and 83287 who places 13, 10, and 9 orders respectively.

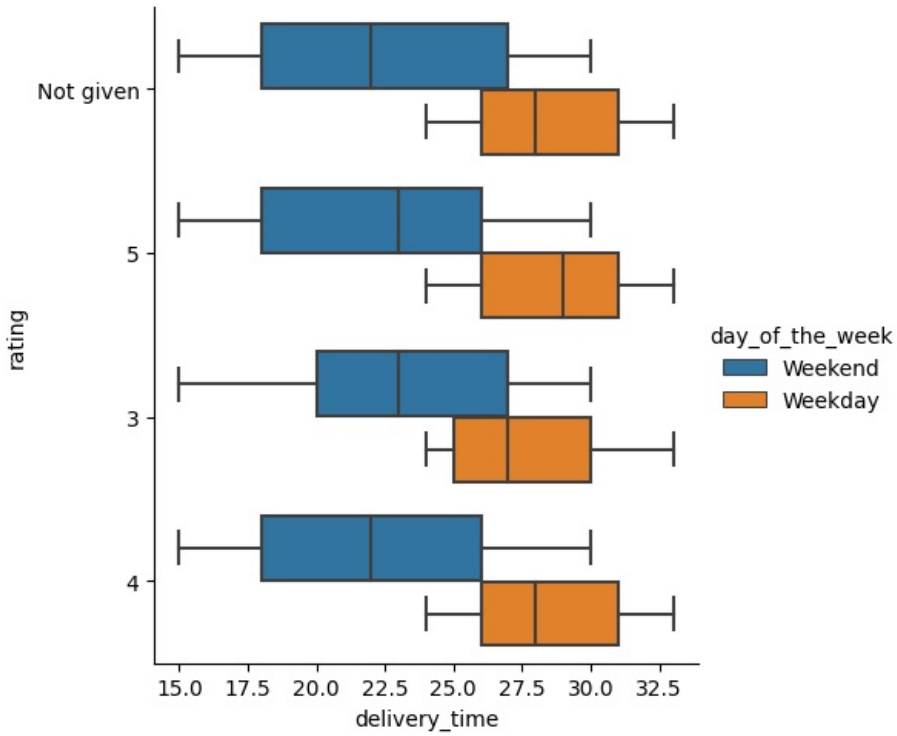
Multivariate Analysis

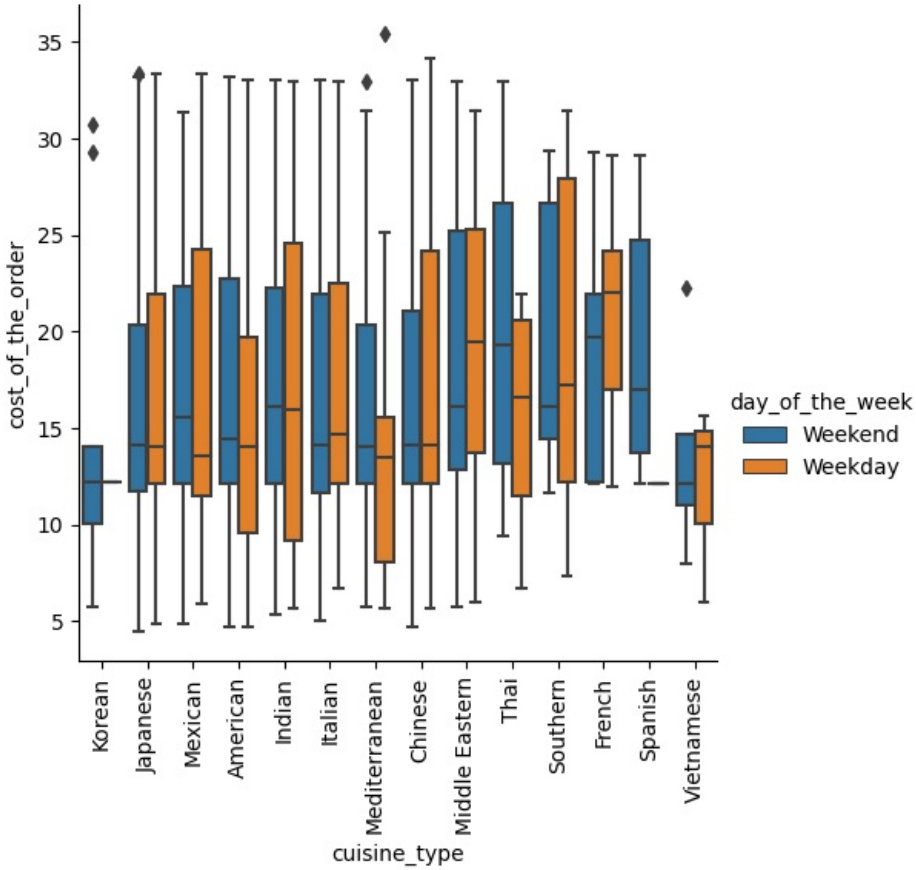
Question 12: Perform a multivariate analysis to explore relationships between the important variables in the dataset. (It is a good idea to explore relations between numerical variables as well as relations between numerical and categorical variables) [10 marks]

In [80]:
```
# Write the code here
sns.heatmap(df.corr(), annot = True, cmap='Spectral', vmin=-1, vmax=1);
sns.catplot(kind="box",data=df, y="rating",x = "delivery_time", hue="day_of_the_week");
sns.catplot(data=df, kind="box", x="cuisine_type", y = "cost_of_the_order", hue="day_of_the_week");
plt.xticks(rotation=90);
```

```
<ipython-input-80-0fe61fd16922>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprec
ated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  sns.heatmap(df.corr(), annot = True, cmap='Spectral', vmin=-1, vmax=1);
```







Observations:

- The delivery time in weekdays are more than weekends
- The cost of order in Southern cuisine is the highest during weekdays.
- The cost of orders in korean cuisine is lowest in weekends.
- The restuarants with rating 5 have delivery time around 28 mins.
- The high correlated data in the heatmap is cost of the order and delivery time. And the lowest correlated data are food preparation time-customer ID, cost of order-delivery time, and food preparation time-delivery time.

Question 13: 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. [3 marks]

In [ ]:
```
# Write the code here
rate = df[df['rating'] != 'Not given'].copy()
rate['rating'] = rate['rating'].astype('int')
rate_count = rate.groupby(['restaurant_name'])['rating'].count().sort_values(ascending = False).reset_index()
rate_count.head(10)
rest =rate_count[rate_count['rating']>50]['restaurant_name']
df_mean_4 = rate[rate['restaurant_name'].isin(rest)].copy()
df_mean_4.groupby(['restaurant_name'])['rating'].mean().sort_values(ascending = False).reset_index().dropna()
```

Out[ ]:

	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

Observations: There are 4 restuarants with more than 50 ratings and have average rating more than 4 which are The Meatball Shop, Blue Ribbon Fried Chicken, Shake Shack, and Blue Ribbon Sushi.

Question 14: 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. [3 marks]

In [ ]:
```
# Write the code here
Over_twenty = df[df["cost_of_the_order"] > 20]
twenty_profit = df["cost_of_the_order"].sum()*25/100
Over_five = df[(df["cost_of_the_order"] > 5) & (df["cost_of_the_order"] <20)]
five_profit = df["cost_of_the_order"].sum()*15/100
revenue = twenty_profit+five_profit
print("Revenue Generated = $", revenue)
```

```
Revenue Generated = $ 12525.928
```

Observations: If 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 the net revenue becomes $12525.93

Question 15: 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.) [2 marks]

In [ ]:
```
# Write the code here
df["total_del_time"] = df["delivery_time"]+ df["food_preparation_time"]
over_sixty = len(df[(df["total_del_time"] > 60)])/len(df)*100
print("Percentage of required delivery time:", over_sixty,"%")
```

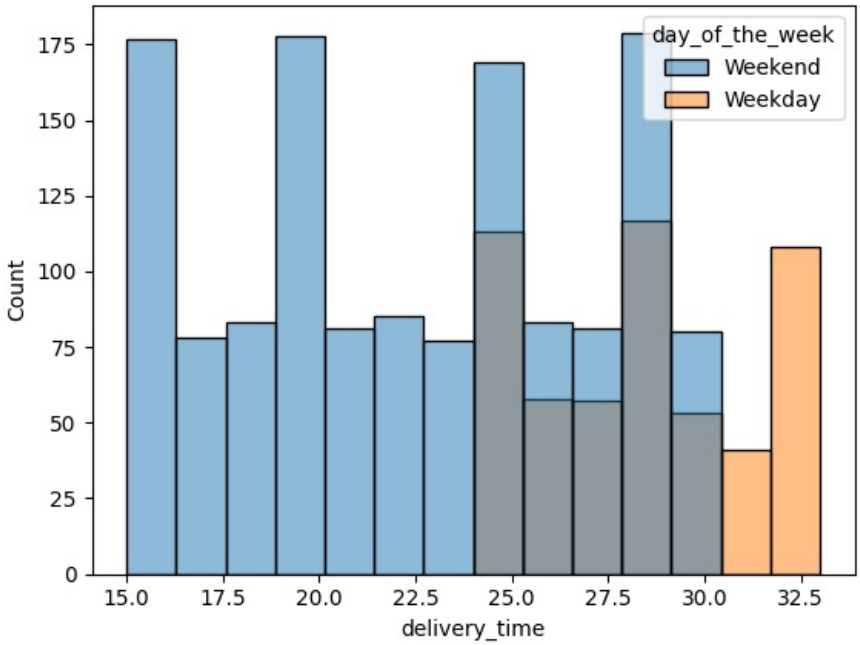
```
Percentage of required delivery time: 10.537407797681771 %
```

Observations: The total required to deliver food consists of food preparation time and delivery time. Here, the percentage of orders take more than 60 minutes to get delivered from the time the order is placed is about 10.54 % which is very low.

Question 16: 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? [2 marks]

In [ ]:
```
# Write the code here
new_cols = df[["delivery_time", "day_of_the_week"]].copy()
week_day = new_cols.groupby("day_of_the_week").mean()
sns.histplot(new_cols,x = "delivery_time", color = 'violet', hue = "day_of_the_week");
print(week_day)
```

```
                delivery_time
day_of_the_week
Weekday             28.340037
Weekend             22.470022
```



Observations: The mean delivery time vary during the weekdays is 5.87 mins more than the weekends.

Conclusion and Recommendations

Question 17: What are your conclusions from the analysis? What recommendations would you like to share to help improve the business? (You can use cuisine type and feedback ratings to drive your business recommendations.) [6 marks]

Conclusions:

- American restuarants are the most popular in New York city.
- From the data set total number of order with 'Not Given' rating is 736.
- The weekdays have longer food delivery time than weekends. Moreover, the percentage of orders with delivery time more than an hour is 10.54%.
- The food preparation time and cost of order have the highest positive correlation which is 0.042.
- The highest negatively correlated data and customer ID and food preparation time.

Recommendations:

- It is observed that 736 of 1897 orders are not rated. Customers should be encouraged to rate orders.
- To reduce delivery time and provide better customer service more employees and delivery drivers can be hired for busy hours in weekdays.
- Discount can be given to new customers after every 3-4 orders. This will help to grow business in the restuarants with least number of customers.

Processing math: 100%