**Project Python Foundations: FoodHub Data Analysis**

**Marks: 60 points**

**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 a Data Scientist at Foodhub 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\_of\_the\_order: 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

**Please read the instructions carefully before starting the project.**

This is a commented Jupyter IPython Notebook file in which all the instructions and tasks to be performed are mentioned. Read along carefully to complete the project.

* Blanks '**\_\_\_**' are provided in the notebook that needs to be filled with an appropriate code to get the correct result. Please replace the blank with the right code snippet. With every '**\_\_\_**' blank, there is a comment that briefly describes what needs to be filled in the blank space.
* Identify the task to be performed correctly, and only then proceed to write the required code.
* Fill the code wherever asked by the commented lines like "# write your code here" or "# complete the code". Running incomplete code may throw an error.
* Please run the codes in a sequential manner from the beginning to avoid any unnecessary errors.
* You can the results/observations derived from the analysis here and use them to create your final presentation.

**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

**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')*

In [ ]:



*# Read the data*

df **=** pd.read\_csv('foodhub\_order.csv\_\_\_\_\_') *## Fill the blank to read the data*

*# Returns the first 5 rows*

df.head(5)

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

In [ ]:



*# Check the shape of the dataset*

df.\_\_info\_\_\_\_\_ *## Fill in the blank*

**Question 2: What are the datatypes of the different columns in the dataset? [0.5 mark]**

In [2]:



df.info()

**---------------------------------------------------------------------------**

**NameError** Traceback (most recent call last)

**~\AppData\Local\Temp/ipykernel\_19212/3771845804.py** in <module>

**----> 1** df**.**info**()**

**NameError**: name 'df' is not defined

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

In [ ]:



*# Checking for missing values in the data*

df.'\_\_\_\_\_\_' *#Write the appropriate function to print the sum of null values for each column*

**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 [ ]:



*# Get the summary statistics of the numerical data*

df.'\_\_\_\_\_\_\_' *## Write the appropriate function to print the statitical summary of the data (Hint - you have seen this in the case studies before)*

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

In [ ]:



df['\_\_\_\_\_\_\_'].value\_counts() *## Complete the code*

**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]**

**Order ID**

In [ ]:



*# check unique order ID*

df['order\_id'].nunique()

**Customer ID**

In [ ]:



*# check unique customer ID*

df['customer\_id'].'\_\_\_\_\_' *## Complete the code to find out number of unique Customer ID*

**Restaurant name**

In [ ]:



*# check unique Restaurant Name*

df['restaurant\_name'].'\_\_\_\_\_' *## Complete the code to find out number of unique Restaurant Name*

**Cuisine type**

In [ ]:



*# Check unique cuisine type*

df['cuisine\_type'].'\_\_\_\_\_\_\_' *## Complete the code to find out number of unique cuisine type*

In [ ]:



plt.figure(figsize **=** (15,5))

sns.countplot(data **=** df, x **=** 'cuisine\_type') *## Create a countplot for cuisine type.*

**Cost of the order**

In [ ]:



sns.histplot(data**=**df,x**=**'cost\_of\_the\_order') *## Histogram for the cost of order*

plt.show()

sns.boxplot(data**=**df,x**=**'cost\_of\_the\_order') *## Boxplot for the cost of order*

plt.show()

**Day of the week**

In [ ]:



*# # Check the unique values*

df['day\_of\_the\_week'].'\_\_\_\_\_\_\_' *## Complete the code to check unique values for the 'day\_of\_the\_week' column*

In [ ]:



sns.countplot(data **=** df, x **=** '\_\_\_\_\_\_') *## Complete the code to plot a bar graph for 'day\_of\_the\_week' column*

**Rating**

In [ ]:



*# Check the unique values*

df['rating'].'\_\_\_\_\_\_\_' *## Complete the code to check unique values for the 'rating' column*

In [ ]:



sns.countplot(data **=** df, x **=** '\_\_\_\_\_\_') *## Complete the code to plot bar graph for 'rating' column*

**Food Preparation time**

In [ ]:



sns.histplot(data**=**df,x**=**'\_\_\_\_\_') *## Complete the code to plot the histogram for the cost of order*

plt.show()

sns.boxplot(data**=**df,x**=**'\_\_\_\_\_') *## Complete the code to plot the boxplot for the cost of order*

plt.show()

**Delivery time**

In [ ]:



sns.histplot(data**=**df,x**=**'\_\_\_\_\_') *## Complete the code to plot the histogram for the delivery time*

plt.show()

sns.boxplot(data**=**df,x**=**'\_\_\_\_\_') *## Complete the code to plot the boxplot for the delivery time*

plt.show()

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

In [ ]:



*# Get top 5 restaurants with highest number of orders*

df['restaurant\_name'].'\_\_\_\_\_\_\_' *## Complete the code*

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

In [ ]:



*# Get most popular cuisine on weekends*

df\_weekend **=** df[df['day\_of\_the\_week'] **==** 'Weekend']

df\_weekend['cuisine\_type'].'\_\_\_\_\_\_\_' *## Complete the code to check unique values for the cuisine type on weekend*

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

In [ ]:



*# Get orders that cost above 20 dollars*

df\_greater\_than\_20 **=** df[df['\_\_\_\_\_\_\_']**>**20] *## Write the appropriate column name to get the orders having cost above $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), '%')

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

In [ ]:



*# Get the mean delivery time*

mean\_del\_time **=** df['delivery\_time'].'\_\_\_\_\_\_\_' *## Write the appropriate function to obtain the mean delivery time*

​

print('The mean delivery time for this dataset is', round(mean\_del\_time, 2), 'minutes')

**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 [ ]:



*# Get the counts of each customer\_id*

df['\_\_\_\_\_\_\_'].value\_counts().head(3) *## Write the appropriate column name to get the top 5 cmost frequent customers*

**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]**

**Cuisine vs Cost of the order**

In [ ]:



*# 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()

**Cuisine vs Food Preparation time**

In [ ]:



*# Relationship between food preparation time and cuisine type*

plt.figure(figsize**=**(15,7))

sns.boxplot('\_\_\_\_\_\_\_') *## Complete the code to visualize the relationship between food preparation time and cuisine type using boxplot*

plt.xticks(rotation **=** 60)

plt.show()

**Day of the Week vs Delivery time**

In [ ]:



*# Relationship between day of the week and delivery time*

plt.figure(figsize**=**(15,7))

sns.boxplot('\_\_\_\_\_\_\_') *## Complete the code to visualize the relationship between day of the week and delivery time using boxplot*

plt.show()

**Run the below code and write your observations on the revenue generated by the restaurants.**

In [ ]:



df.groupby(['restaurant\_name'])['cost\_of\_the\_order'].sum().sort\_values(ascending **=** **False**).head(14)

**Rating vs Delivery time**

In [ ]:



*# Relationship between rating and delivery time*

plt.figure(figsize**=**(15, 7))

sns.pointplot(x **=** 'rating', y **=** 'delivery\_time', data **=** df)

plt.show()

**Rating vs Food preparation time**

In [ ]:



*# Relationship between rating and food preparation time*

plt.figure(figsize**=**(15, 7))

sns.pointplot('\_\_\_\_\_\_\_') *## Complete the code to visualize the relationship between rating and food preparation time using pointplot*

plt.show()

**Rating vs Cost of the order**

In [ ]:



*# Relationship between rating and cost of the order*

plt.figure(figsize**=**(15, 7))

sns.pointplot('\_\_\_\_\_\_\_') *## Complete the code to visualize the relationship between rating and cost of the order using pointplot*

plt.show()

**Correlation among variables**

In [ ]:



*# Plot the heatmap*

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()

**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 [ ]:



*# 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()

In [ ]:



*# Get the restaurant names that have rating count more than 50*

rest\_names **=** df\_rating\_count['\_\_\_\_\_\_\_\_\_\_\_\_\_\_']['restaurant\_name'] *## Complete the code to get the restaurant names having rating count more than 50*

​

*# 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(['\_\_\_\_\_\_\_'])['\_\_\_\_\_\_\_'].mean().sort\_values(ascending **=** **False**).reset\_index().dropna() *## Complete the code to find the mean rating*

​

**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 [ ]:



*#function to determine the revenue*

**def** compute\_rev(x):

**if** x **>** 20:

**return** x**\***0.25

**elif** x **>** 5:

**return** x**\***0.15

**else**:

**return** x**\***0

​

df['Revenue'] **=** df['\_\_\_\_\_\_\_\_'].apply(compute\_rev) *## Write the apprpriate column name to compute the revenue*

df.head()

In [ ]:



*# get the total revenue and print it*

total\_rev **=** df['Revenue'].'\_\_\_\_\_' *## Write the appropriate function to get the total revenue*

print('The net revenue is around', round(total\_rev, 2), 'dollars')

**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 [ ]:



*# Calculate total delivery time and add a new column to the dataframe df to store the total delivery time*

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 delivery time (see Question 9 for reference)*

​

​

​

​

**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 [ ]:



*# 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*

​

​

​

​

​

**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:**

**Recommendations:**