zomato

October 20, 2024

[3]:
```
#Python Project On Top 1000 Bollywood Movies And Their Box Office
import seaborn as sns
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

[4]:
```
zom=pd.read_csv("Zomatodata.csv")
zom
```

[4]:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1/5	775	800	Buffet
1	Spice Elephant	Yes	No	4.1/5	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8/5	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	300	Buffet
4	Grand Village	No	No	3.8/5	166	600	Buffet
..	…	…	…	…	…	…	…
143	Melting Melodies	No	No	3.3/5	0	100	Dining
144	New Indraprasta	No	No	3.3/5	0	150	Dining
145	Anna Kuteera	Yes	No	4.0/5	771	450	Dining
146	Darbar	No	No	3.0/5	98	800	Dining
147	Vijayalakshmi	Yes	No	3.9/5	47	200	Dining

[148 rows x 7 columns]

[5]:
```
#Data Cleaning And Data Processing
#1 to modify the rate by only printing rating that 4.1,3,7 etc removing "/"␣
 ↪part [Converting The Data Type Of Rate Column]
def rating(value):
    value=str(value).split("/")
    value=value[0];
    return float(value)

zom["rate"]=zom["rate"].apply(rating)
zom
```

[5]:

	name	online_order	book_table	rate	votes	\
0	Jalsa	Yes	Yes	4.1	775	
1	Spice Elephant	Yes	No	4.1	787	
2	San Churro Cafe	Yes	No	3.8	918	
3	Addhuri Udupi Bhojana	No	No	3.7	88	
4	Grand Village	No	No	3.8	166	
..	…	…	…	…	…	
143	Melting Melodies	No	No	3.3	0	
144	New Indraprasta	No	No	3.3	0	
145	Anna Kuteera	Yes	No	4.0	771	
146	Darbar	No	No	3.0	98	
147	Vijayalakshmi	Yes	No	3.9	47	

	approx_cost(for two people)	listed_in(type)
0	800	Buffet
1	800	Buffet
2	800	Buffet
3	300	Buffet
4	600	Buffet
..	…	…
143	100	Dining
144	150	Dining
145	450	Dining
146	800	Dining
147	200	Dining

[148 rows x 7 columns]

[6]:
```
zom.head(10)
```

[6]:

	name	online_order	book_table	\
0	Jalsa	Yes	Yes	
1	Spice Elephant	Yes	No	
2	San Churro Cafe	Yes	No	
3	Addhuri Udupi Bhojana	No	No	
4	Grand Village	No	No	

```
5                                  Timepass Dinner          Yes          No
6  Rosewood International Hotel - Bar & Restaurant           No          No
7                                           Onesta          Yes         Yes
8                                  Penthouse Cafe          Yes          No
9                                       Smacznego          Yes          No

   rate  votes  approx_cost(for two people) listed_in(type)
0   4.1    775                         800          Buffet
1   4.1    787                         800          Buffet
2   3.8    918                         800          Buffet
3   3.7     88                         300          Buffet
4   3.8    166                         600          Buffet
5   3.8    286                         600          Buffet
6   3.6      8                         800          Buffet
7   4.6   2556                         600           Cafes
8   4.0    324                         700           other
9   4.2    504                         550           Cafes
```

[99]:
```
#To Check Null Values In Columns True/False
zom.isnull()
```

[99]:
```
      name  online_order  book_table   rate  votes  \
0    False         False       False  False  False
1    False         False       False  False  False
2    False         False       False  False  False
3    False         False       False  False  False
4    False         False       False  False  False
..     …             …          …     …     …
143  False         False       False  False  False
144  False         False       False  False  False
145  False         False       False  False  False
146  False         False       False  False  False
147  False         False       False  False  False

     approx_cost(for two people)  listed_in(type)
0                          False            False
1                          False            False
2                          False            False
3                          False            False
4                          False            False
..                           …                …
143                        False            False
144                        False            False
145                        False            False
146                        False            False
147                        False            False
```

```
[148 rows x 7 columns]
```

[ ]:
```
#All Columns Are False Means No Null Values
```

[7]:
```
#DATA CLEANING
#to check missing or null values
zom.info()
```

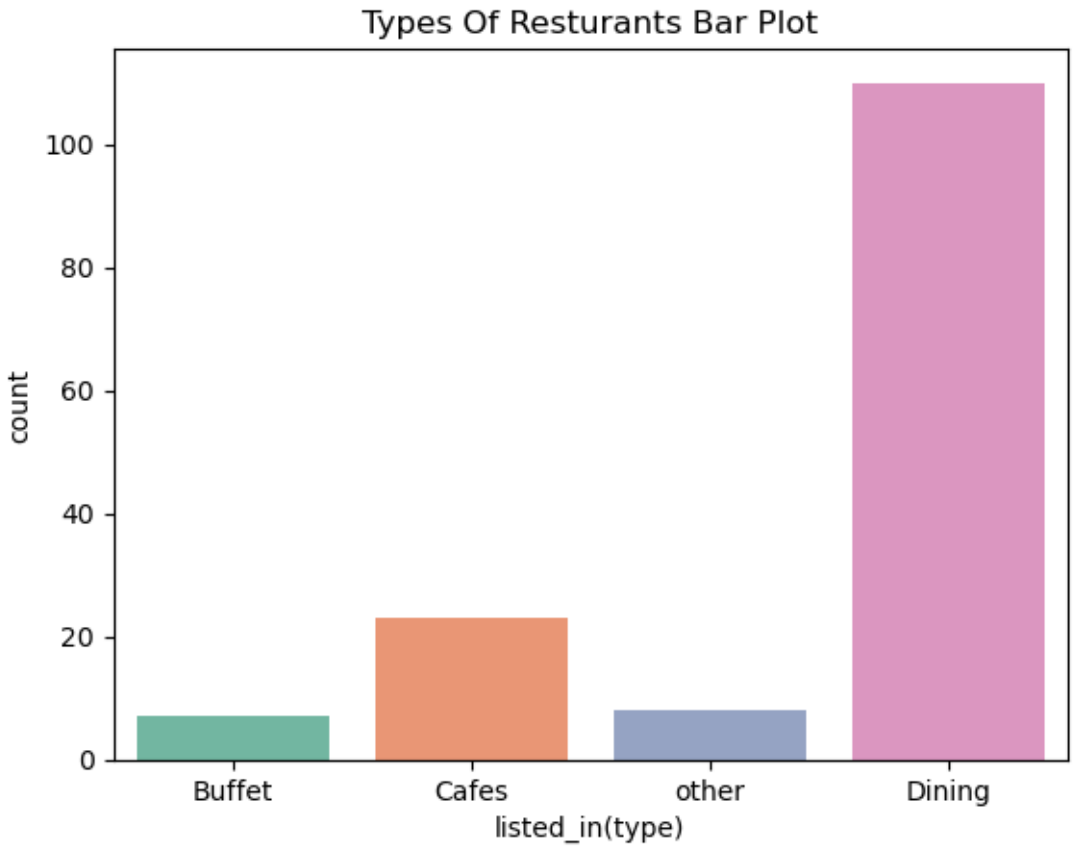
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   name                         148 non-null    object 
 1   online_order                 148 non-null    object 
 2   book_table                   148 non-null    object 
 3   rate                         148 non-null    float64
 4   votes                        148 non-null    int64  
 5   approx_cost(for two people)  148 non-null    int64  
 6   listed_in(type)              148 non-null    object 
dtypes: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

[101]:
```
#To Check Null Values As Per Order
zom.isnull().sum()
```

[101]:
```
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
approx_cost(for two people)    0
listed_in(type)                0
dtype: int64
```

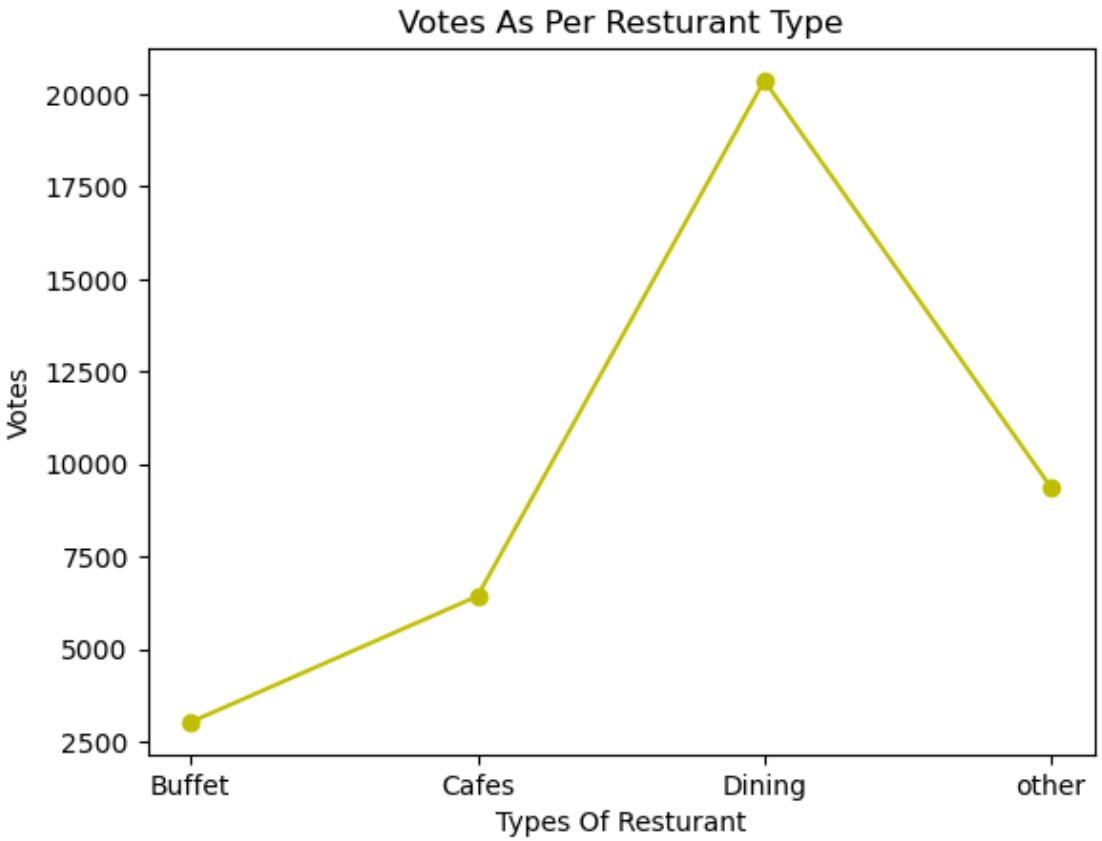
[8]:
```
#NO NULL VALUES PRESENT
```

[43]:
```
#Q1 Types Of Resturant Which types is used by cosumer more?
#We Can Show It By Bar Graph Seaborn Library

plt.title("Types Of Resturants Bar Plot")
sns.
 ↪countplot(x=zom["listed_in(type)"],hue=zom["listed_in(type)"],palette="Set2")
plt.show()
```



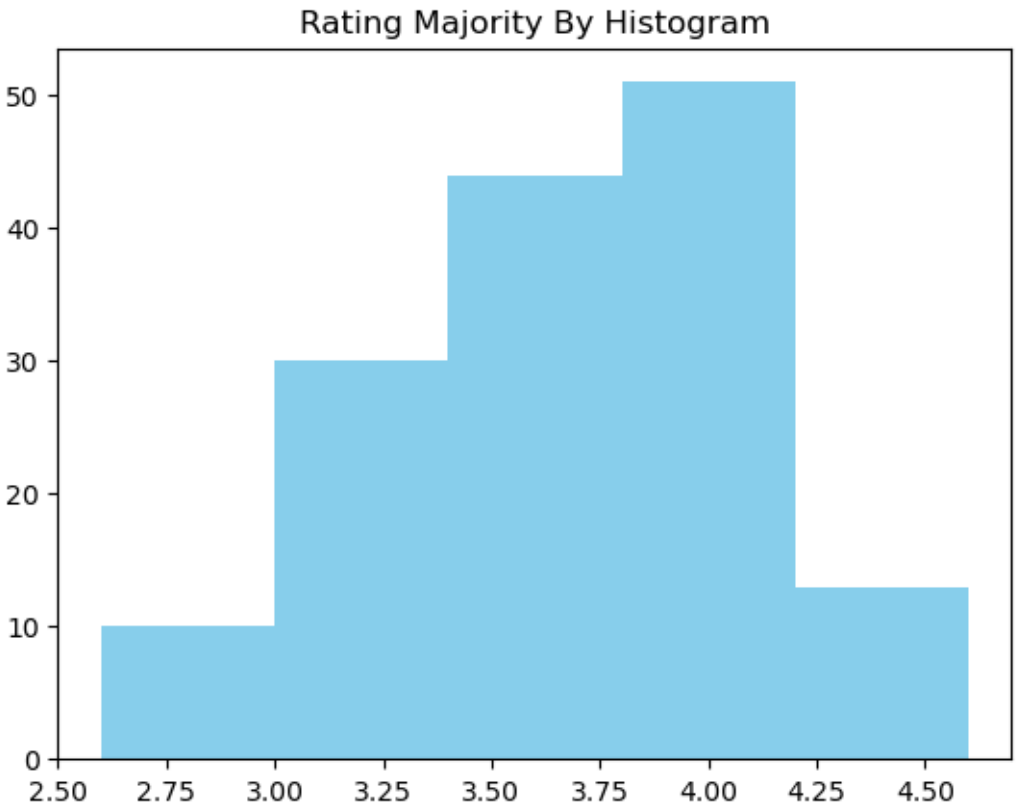
[59]:
```
#Conclusion-Customer prefer the dining type in resturant
```

[83]:
```
#Q2 How many votes has each type of resturant recieved from customer
#We can show it by line plot/graph "x=types of resturant","y=Votes"

plt.title("Votes As Per Resturant Type")
group=zom.groupby("listed_in(type)")["votes"].sum()
result=pd.DataFrame({"votes":group})
plt.xlabel("Types Of Resturant")
plt.ylabel("Votes")
plt.plot(result,c="y",marker="o")
plt.show()
```



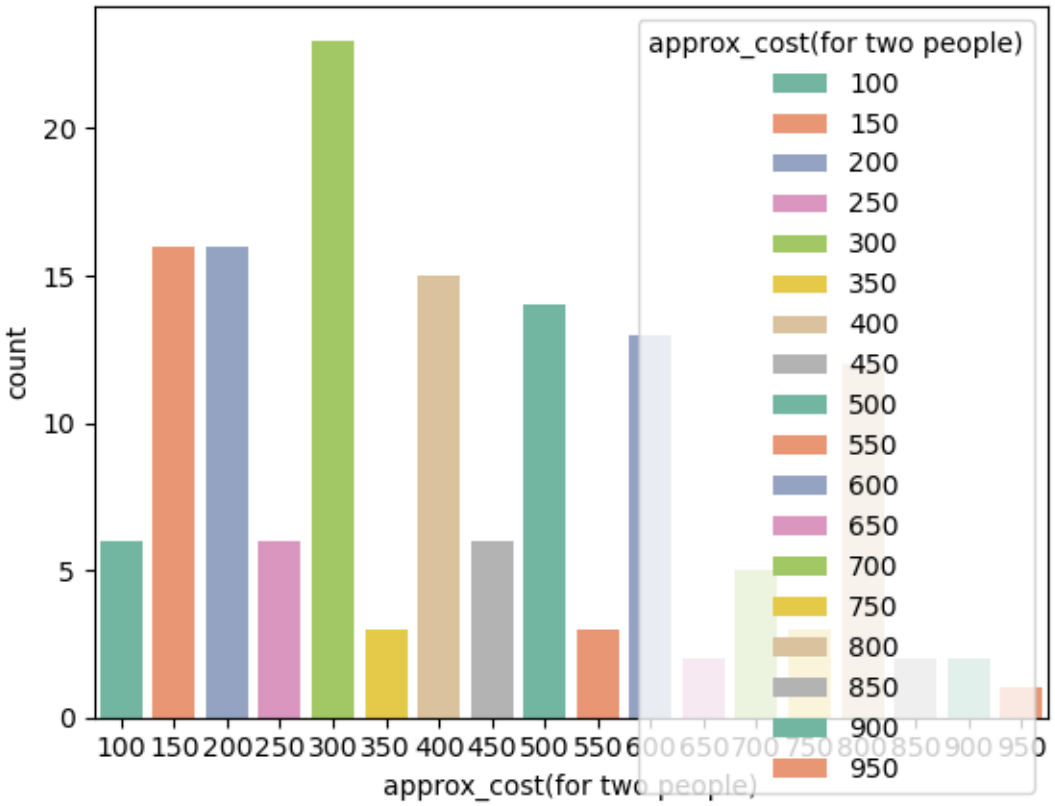
[73]:
```
#Conclusion-Dining Has Got More Votes From All Of Other Types
```

[53]:
```
#Q3 What are the rating majority of resturant recieved by the customer
#We Will Show It By Histogram

plt.title("Rating Majority By Histogram")
plt.hist(zom["rate"],bins=5,color="skyblue")
plt.show()
```



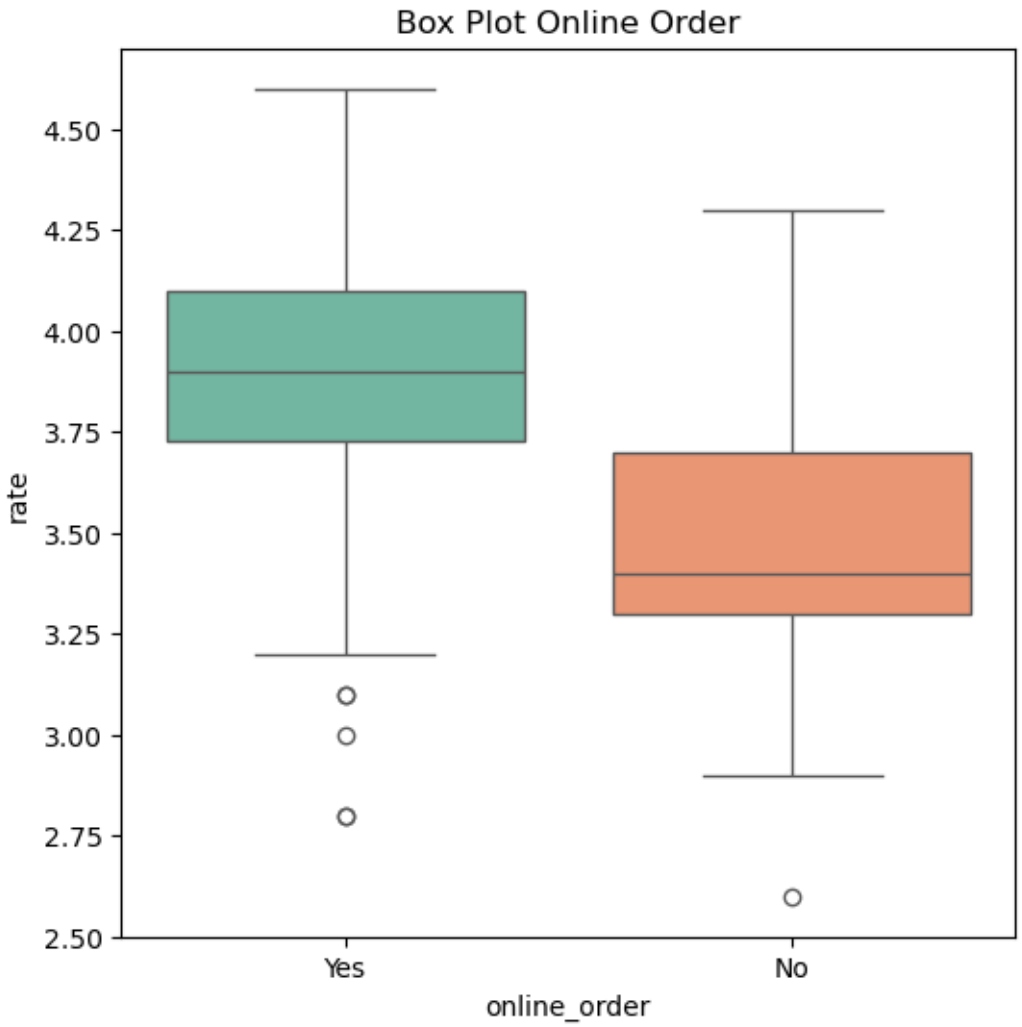
[87]:
```
#Conclusion-The Higest Rating Is From 3.50 To 4.25
```

[59]:
```
#Q4 What is the average spending money on couple orders
sns.countplot(x=zom["approx_cost(for two people)"],hue=zom["approx_cost(for two␣
↪people)"],palette="Set2")
plt.show()
```



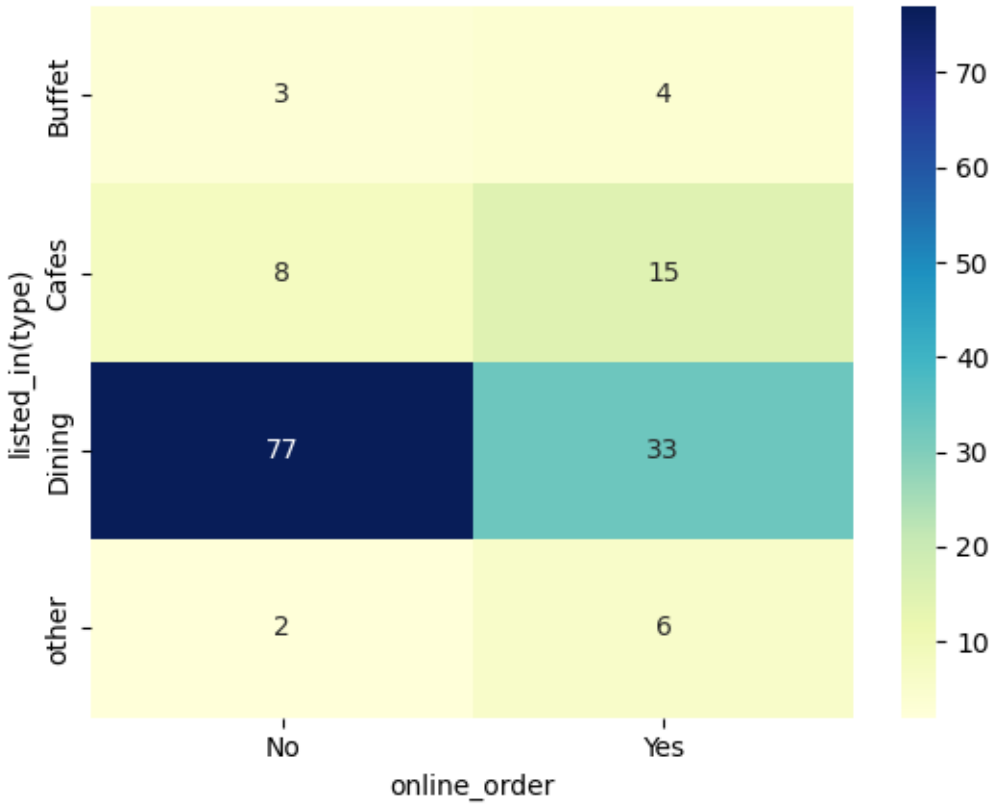
[61]:
```
#Conclusion-Here Couple Spends 300 Rupees To Order Food Online
```

[79]:
```
#Q5-Which Mode has recieved the maximum rating online or offline
#we can see it using boxplot
plt.figure(figsize=(6,6))
plt.title("Box Plot Online Order")
sns.boxplot(x="online_order",y="rate",␣
 ↪hue="online_order",data=zom,palette="Set2")
plt.show()
```



[81]:
```
#Conclusion-As per box plot we can say that online order has maximum rating
```

[97]:
```
#Q6-Which restaurant recieved more offline order so that zomato can provide␣
 ↪customer good offers
pivot_table=zom.
 ↪pivot_table(index="listed_in(type)",columns="online_order",aggfunc="size",fill_value=0)
sns.heatmap(pivot_table,annot=True,cmap="YlGnBu",fmt="d")
plt.show()
```



[ ]:
```
#Conclusion:Dinning Mainly Accept Offline Orders Whereas Cafe Primarily Recieve␣
↪Online Orders.This Tells That Clients Prefers Orders In Person At␣
↪Restaurants,But Prefer Online Order At Cafes!!!
```