reg2-1

November 21, 2023

1 ASSIGNMENT 3

NAME: RISHAV KUMAR

ROLL NO. 2301560042 My github account link : Github

2 QUESTION 2

Dataset source : zomato

2.0.1 Zomato dataset cleaning and visualization

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
data = pd.read_csv(r'C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\zomato.csv')
```

[3]:
```
data.head()
```

[3]:
```
                                                 url  \
0  https://www.zomato.com/bangalore/jalsa-banasha…
1  https://www.zomato.com/bangalore/spice-elephan…
2  https://www.zomato.com/SanchurroBangalore?cont…
3  https://www.zomato.com/bangalore/addhuri-udupi…
4  https://www.zomato.com/bangalore/grand-village…

                                             address                  name  \
0  942, 21st Main Road, 2nd Stage, Banashankari, …                 Jalsa
1  2nd Floor, 80 Feet Road, Near Big Bazaar, 6th …         Spice Elephant
2  1112, Next to KIMS Medical College, 17th Cross…        San Churro Cafe
3  1st Floor, Annakuteera, 3rd Stage, Banashankar…  Addhuri Udupi Bhojana
4  10, 3rd Floor, Lakshmi Associates, Gandhi Baza…          Grand Village

  online_order book_table   rate  votes                           phone  \
```

```
0          Yes        Yes  4.1/5    775    080 42297555\r\n+91 9743772233
1          Yes         No  4.1/5    787                     080 41714161
2          Yes         No  3.8/5    918                   +91 9663487993
3           No         No  3.7/5     88                   +91 9620009302
4           No         No  3.8/5    166  +91 8026612447\r\n+91 9901210005

       location            rest_type  \
0  Banashankari        Casual Dining
1  Banashankari        Casual Dining
2  Banashankari  Cafe, Casual Dining
3  Banashankari          Quick Bites
4  Basavanagudi        Casual Dining

                                        dish_liked  \
0  Pasta, Lunch Buffet, Masala Papad, Paneer Laja…
1  Momos, Lunch Buffet, Chocolate Nirvana, Thai G…
2  Churros, Cannelloni, Minestrone Soup, Hot Choc…
3                                     Masala Dosa
4                              Panipuri, Gol Gappe

                      cuisines approx_cost(for two people)  \
0  North Indian, Mughlai, Chinese                       800
1     Chinese, North Indian, Thai                       800
2          Cafe, Mexican, Italian                       800
3     South Indian, North Indian                        300
4       North Indian, Rajasthani                        600

                                     reviews_list menu_item  \
0  [('Rated 4.0', 'RATED\n  A beautiful place to …        []
1  [('Rated 4.0', 'RATED\n  Had been here for din…        []
2  [('Rated 3.0', "RATED\n  Ambience is not that …        []
3  [('Rated 4.0', "RATED\n  Great food and proper…        []
4  [('Rated 4.0', 'RATED\n  Very good restaurant …        []

  listed_in(type) listed_in(city)
0         Buffet    Banashankari
1         Buffet    Banashankari
2         Buffet    Banashankari
3         Buffet    Banashankari
4         Buffet    Banashankari
```

rows and columns in data set

[4]:
```
data.shape
```

[4]:
```
(51717, 17)
```

all the columns in dataset

[5]:
```
data.columns
```

[5]:
```
Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'phone', 'location', 'rest_type', 'dish_liked', 'cuisines',
       'approx_cost(for two people)', 'reviews_list', 'menu_item',
       'listed_in(type)', 'listed_in(city)'],
      dtype='object')
```

[6]:
```
df= data.copy()
```

dropping useless columns

[7]:
```
df.drop(['url',
         'address',
         "phone",
         'dish_liked',
         'menu_item',
         'reviews_list'], axis=1, inplace=True)
```

[8]:
```
df.columns
```

[8]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'cuisines', 'approx_cost(for two people)',
       'listed_in(type)', 'listed_in(city)'],
      dtype='object')
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   name                         51717 non-null  object
 1   online_order                 51717 non-null  object
 2   book_table                   51717 non-null  object
 3   rate                         43942 non-null  object
 4   votes                        51717 non-null  int64
 5   location                     51696 non-null  object
 6   rest_type                    51490 non-null  object
 7   cuisines                     51672 non-null  object
 8   approx_cost(for two people)  51371 non-null  object
 9   listed_in(type)              51717 non-null  object
 10  listed_in(city)              51717 non-null  object
dtypes: int64(1), object(10)
memory usage: 4.3+ MB
```

checking for null values

[10]:
```
df.isna().sum()
```

[10]:
```
name                            0
online_order                    0
book_table                      0
rate                         7775
votes                           0
location                       21
rest_type                     227
cuisines                       45
approx_cost(for two people)   346
listed_in(type)                 0
listed_in(city)                 0
dtype: int64
```

number of unique character in rate

[11]:
```
rate_uni=len(df.rate.unique())
rate_uni
```

[11]:
```
65
```

[12]:
```
df.rate.unique()
```

[12]:
```
array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
       '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
       '4.3/5', 'NEW', '2.9/5', '3.5/5', nan, '2.6/5', '3.8 /5', '3.4/5',
       '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
       '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
       '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
       '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
       '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
       '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
       '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

[13]:
```
def clean_rate(d):
    if d== "NEW" or d=="-":
        return np.nan
    else:
        d=str(d).split('/')
        d=d[0]
        return float(d)
```

[14]:
```
df.rate= df.rate.apply(clean_rate)
```

[15]:
```
df.rate.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 51717 entries, 0 to 51716
Series name: rate
Non-Null Count  Dtype
--------------  -----
41665 non-null  float64
dtypes: float64(1)
memory usage: 404.2 KB
```

[16]:
```
df.rate.isna().sum()
```

[16]:
```
10052
```

[17]:
```
df.rate= df.rate.fillna(df.rate.mode()[0])
```

[18]:
```
df.rate
```

[18]:
```
0        4.1
1        4.1
2        3.8
3        3.7
4        3.8
        ...
51712    3.6
51713    3.9
51714    3.9
51715    4.3
51716    3.4
Name: rate, Length: 51717, dtype: float64
```

[19]:
```
df.rate.isna().sum()
```

[19]:
```
0
```

[20]:
```
df.isna().sum()
```

[20]:
```
name                            0
online_order                    0
book_table                      0
rate                            0
votes                           0
location                       21
rest_type                     227
cuisines                       45
approx_cost(for two people)   346
listed_in(type)                 0
listed_in(city)                 0
```

```
dtype: int64
```

we can drop other columns which contains nan values

[21]:
```
df.dropna(inplace=True)
```

[22]:
```
df.isna().sum()
```

[22]:
```
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
location                       0
rest_type                      0
cuisines                       0
approx_cost(for two people)    0
listed_in(type)                0
listed_in(city)                0
dtype: int64
```

[23]:
```
df.rename(columns={'approx_cost(for two people)':'cost_for_2',␣
 ↪'listed_in(type)':'type'}, inplace=True)
```

[24]:
```
df.columns
```

[24]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'cuisines', 'cost_for_2', 'type', 'listed_in(city)'],
      dtype='object')
```

[25]:
```
df['listed_in(city)'].unique()
```

[25]:
```
array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
       'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
       'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
       'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
       'Koramangala 4th Block', 'Koramangala 5th Block',
       'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
       'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
       'Old Airport Road', 'Rajajinagar', 'Residency Road',
       'Sarjapur Road', 'Whitefield'], dtype=object)
```

[26]:
```
df['location'].unique()
```

[26]:
```
array(['Banashankari', 'Basavanagudi', 'Mysore Road', 'Jayanagar',
       'Kumaraswamy Layout', 'Rajarajeshwari Nagar', 'Vijay Nagar',
       'Uttarahalli', 'JP Nagar', 'South Bangalore', 'City Market',
```

```
'Nagarbhavi', 'Bannerghatta Road', 'BTM', 'Kanakapura Road',
'Bommanahalli', 'CV Raman Nagar', 'Electronic City', 'HSR',
'Marathahalli', 'Wilson Garden', 'Shanti Nagar',
'Koramangala 5th Block', 'Koramangala 8th Block', 'Richmond Road',
'Koramangala 7th Block', 'Jalahalli', 'Koramangala 4th Block',
'Bellandur', 'Sarjapur Road', 'Whitefield', 'East Bangalore',
'Old Airport Road', 'Indiranagar', 'Koramangala 1st Block',
'Frazer Town', 'RT Nagar', 'MG Road', 'Brigade Road',
'Lavelle Road', 'Church Street', 'Ulsoor', 'Residency Road',
'Shivajinagar', 'Infantry Road', 'St. Marks Road',
'Cunningham Road', 'Race Course Road', 'Commercial Street',
'Vasanth Nagar', 'HBR Layout', 'Domlur', 'Ejipura',
'Jeevan Bhima Nagar', 'Old Madras Road', 'Malleshwaram',
'Seshadripuram', 'Kammanahalli', 'Koramangala 6th Block',
'Majestic', 'Langford Town', 'Central Bangalore', 'Sanjay Nagar',
'Brookefield', 'ITPL Main Road, Whitefield',
'Varthur Main Road, Whitefield', 'KR Puram',
'Koramangala 2nd Block', 'Koramangala 3rd Block', 'Koramangala',
'Hosur Road', 'Rajajinagar', 'Banaswadi', 'North Bangalore',
'Nagawara', 'Hennur', 'Kalyan Nagar', 'New BEL Road', 'Jakkur',
'Rammurthy Nagar', 'Thippasandra', 'Kaggadasapura', 'Hebbal',
'Kengeri', 'Sankey Road', 'Sadashiv Nagar', 'Basaveshwara Nagar',
'Yeshwantpur', 'West Bangalore', 'Magadi Road', 'Yelahanka',
'Sahakara Nagar', 'Peenya'], dtype=object)
```

Both location and listed_in(city) have same/similar value so we can drop one

[27]:
```
df.drop(['listed_in(city)'], axis=1, inplace=True)
```

[28]:
```
df.columns
```

[28]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'cuisines', 'cost_for_2', 'type'],
      dtype='object')
```

[29]:
```
df.cost_for_2.unique()
```

[29]:
```
array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
       '900', '200', '750', '150', '850', '100', '1,200', '350', '250',
       '950', '1,000', '1,500', '1,300', '199', '80', '1,100', '160',
       '1,600', '230', '130', '50', '190', '1,700', '1,400', '180',
       '1,350', '2,200', '2,000', '1,800', '1,900', '330', '2,500',
       '2,100', '3,000', '2,800', '3,400', '40', '1,250', '3,500',
       '4,000', '2,400', '2,600', '120', '1,450', '469', '70', '3,200',
       '60', '560', '240', '360', '6,000', '1,050', '2,300', '4,100',
       '5,000', '3,700', '1,650', '2,700', '4,500', '140'], dtype=object)
```

[30]:
```
# def rem_coma(d):
#     d=d.replace(',',"")
#     return int(d)
```

[31]:
```
df.cost_for_2= df.cost_for_2.apply(lambda d: int(d.replace(',',"")))
```

[32]:
```
df.cost_for_2.info()
```

```
<class 'pandas.core.series.Series'>
Index: 51148 entries, 0 to 51716
Series name: cost_for_2
Non-Null Count  Dtype
--------------  -----
51148 non-null  int64
dtypes: int64(1)
memory usage: 799.2 KB
```

[33]:
```
df.rest_type.value_counts()
```

[33]:
```
rest_type
Quick Bites                   19046
Casual Dining                 10273
Cafe                           3687
Delivery                       2578
Dessert Parlor                 2245
                              ...
Dessert Parlor, Kiosk             2
Food Court, Beverage Shop         2
Dessert Parlor, Food Court        2
Quick Bites, Kiosk                1
Sweet Shop, Dessert Parlor        1
Name: count, Length: 93, dtype: int64
```

[34]:
```
rest_typ=df.rest_type.value_counts(ascending= False)
ltt= rest_typ[rest_typ<1000]
ltt
```

[34]:
```
rest_type
Beverage Shop                 865
Bar                           686
Food Court                    619
Sweet Shop                    468
Bar, Casual Dining            415
                             ...
Dessert Parlor, Kiosk           2
Food Court, Beverage Shop       2
Dessert Parlor, Food Court      2
```

```
Quick Bites, Kiosk              1
Sweet Shop, Dessert Parlor      1
Name: count, Length: 85, dtype: int64
```

[35]:
```
def cat_rest(d):
    if d in ltt:
        return 'other'
    else:
        return d
```

[36]:
```
df.rest_type= df.rest_type.apply(cat_rest)
```

[37]:
```
df.rest_type.value_counts(ascending=False)
```

[37]:
```
rest_type
Quick Bites           19046
Casual Dining         10273
other                  9028
Cafe                   3687
Delivery               2578
Dessert Parlor         2245
Takeaway, Delivery     2014
Bakery                 1141
Casual Dining, Bar     1136
Name: count, dtype: int64
```

[38]:
```
loc= df.location.value_counts(ascending=True)
```

[39]:
```
loc
```

[39]:
```
location
Peenya                      1
Rajarajeshwari Nagar        2
Jakkur                      3
Yelahanka                   5
West Bangalore              6
                          ...
Whitefield               2109
JP Nagar                 2219
Koramangala 5th Block    2481
HSR                      2496
BTM                      5071
Name: count, Length: 93, dtype: int64
```

[40]:
```
ll= loc[loc<300]
```

[41]:
```
def cat_loc(d):
    if d in ll:
        return "other"
    else:
        return d
```

[42]:
```
df.location= df.location.apply(cat_loc)
```

[43]:
```
df.location.value_counts()
```

[43]:
```
location
BTM                    5071
other                  4962
HSR                    2496
Koramangala 5th Block  2481
JP Nagar               2219
Whitefield             2109
Indiranagar            2033
Jayanagar              1916
Marathahalli           1808
Bannerghatta Road      1611
Bellandur              1271
Electronic City        1248
Koramangala 1st Block  1237
Brigade Road           1218
Koramangala 7th Block  1176
Koramangala 6th Block  1129
Sarjapur Road          1049
Koramangala 4th Block  1017
Ulsoor                 1017
Banashankari            904
MG Road                 894
Kalyan Nagar            841
Richmond Road           804
Malleshwaram            724
Frazer Town             720
Basavanagudi            684
Residency Road          674
Brookefield             656
Banaswadi               645
New BEL Road            644
Kammanahalli            640
Rajajinagar             591
Church Street           569
Lavelle Road            523
Shanti Nagar            511
Shivajinagar            499
```

```
Cunningham Road           491
Domlur                    482
Old Airport Road          437
Ejipura                   434
Commercial Street         370
St. Marks Road            343
Name: count, dtype: int64
```

[44]:
```
cuis= df.cuisines.value_counts(ascending=False)
```

[45]:
```
cuis
```

[45]:
```
cuisines
North Indian                                  2858
North Indian, Chinese                         2355
South Indian                                  1822
Biryani                                        906
Bakery, Desserts                               899
                                              ...
Beverages, Burger                                1
North Indian, Mughlai, Lucknowi                  1
Continental, Thai, North Indian, Chinese         1
North Indian, Bengali, Chinese, Beverages        1
North Indian, Chinese, Arabian, Momos            1
Name: count, Length: 2704, dtype: int64
```

[46]:
```
cl= cuis[cuis<100]
```

[47]:
```
def cat_cuis(d):
    if d in cl:
        return 'other'
    else:
        return d
```

[48]:
```
df.cuisines= df.cuisines.apply(cat_cuis)
```

[49]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 51148 entries, 0 to 51716
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   name          51148 non-null  object
 1   online_order  51148 non-null  object
 2   book_table    51148 non-null  object
 3   rate          51148 non-null  float64
```

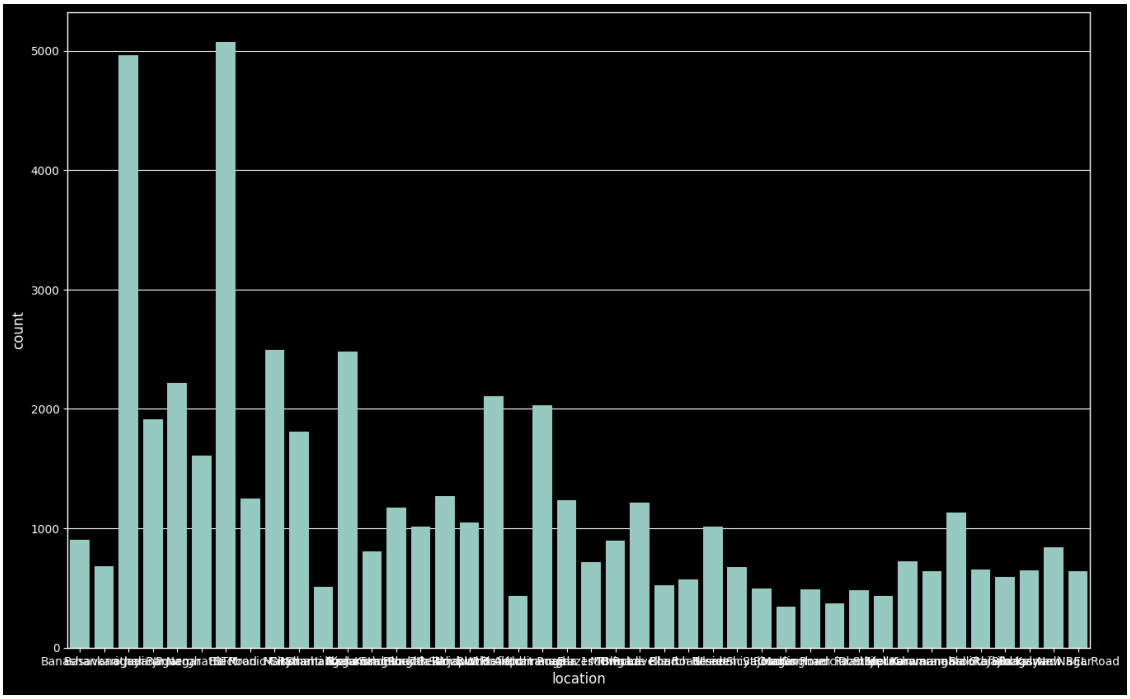
```
 4   votes          51148 non-null  int64
 5   location       51148 non-null  object
 6   rest_type      51148 non-null  object
 7   cuisines       51148 non-null  object
 8   cost_for_2     51148 non-null  int64
 9   type           51148 non-null  object
dtypes: float64(1), int64(2), object(7)
memory usage: 4.3+ MB
```

2.1 visualization

[50]:
```
plt.style.use("ggplot")
plt.style.use("dark_background")
```

2.1.1 location vs number of restarant

[51]:
```
plt.figure(figsize=(16,10))
ax= sns.countplot(data= df, x= "location")
plt.xticks(rotation=0)
plt.show()
```

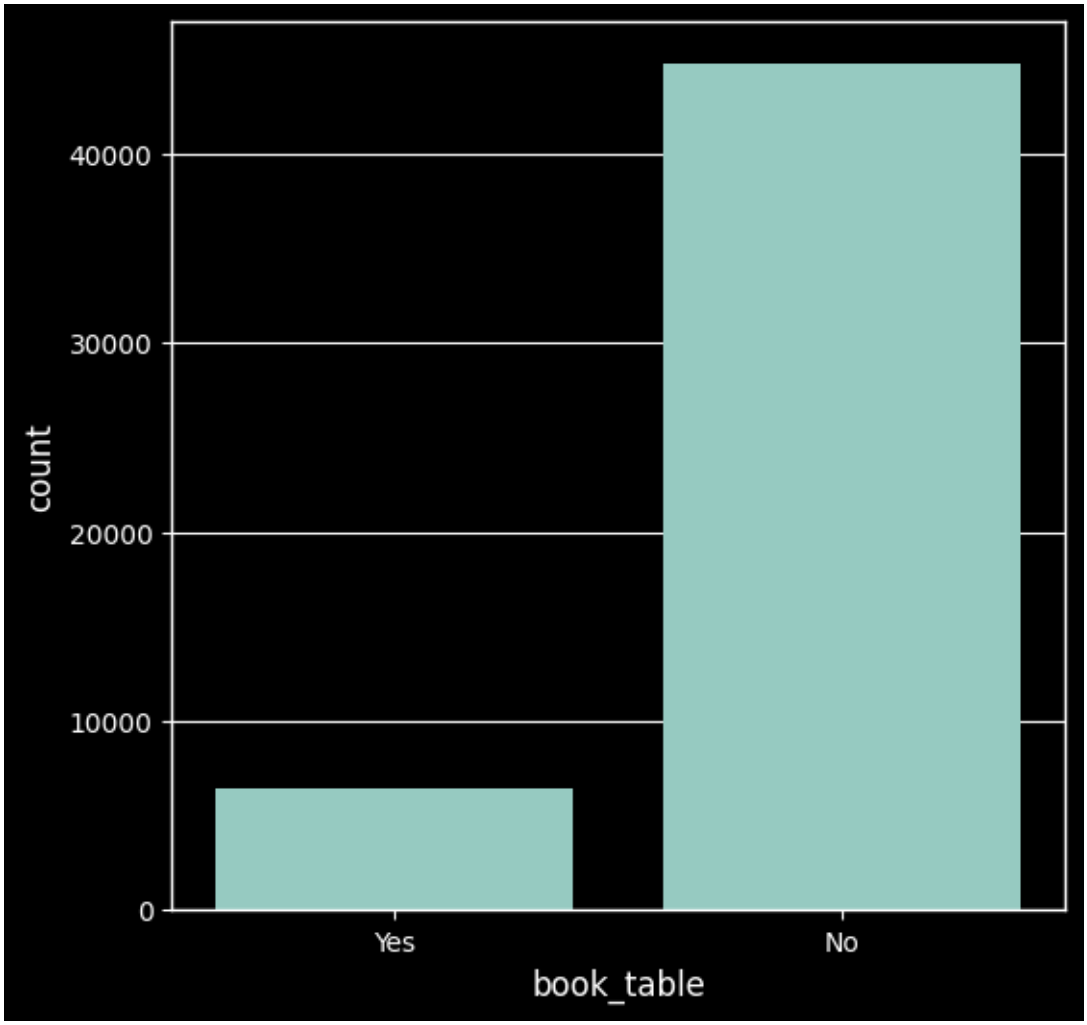


2.1.2 support online or not

[52]:
```
plt.figure(figsize= (6,4))
ax= sns.countplot(data= df, x='online_order')
plt.xticks(rotation=0)
plt.show()
```

2.1.3 tables are booked or not

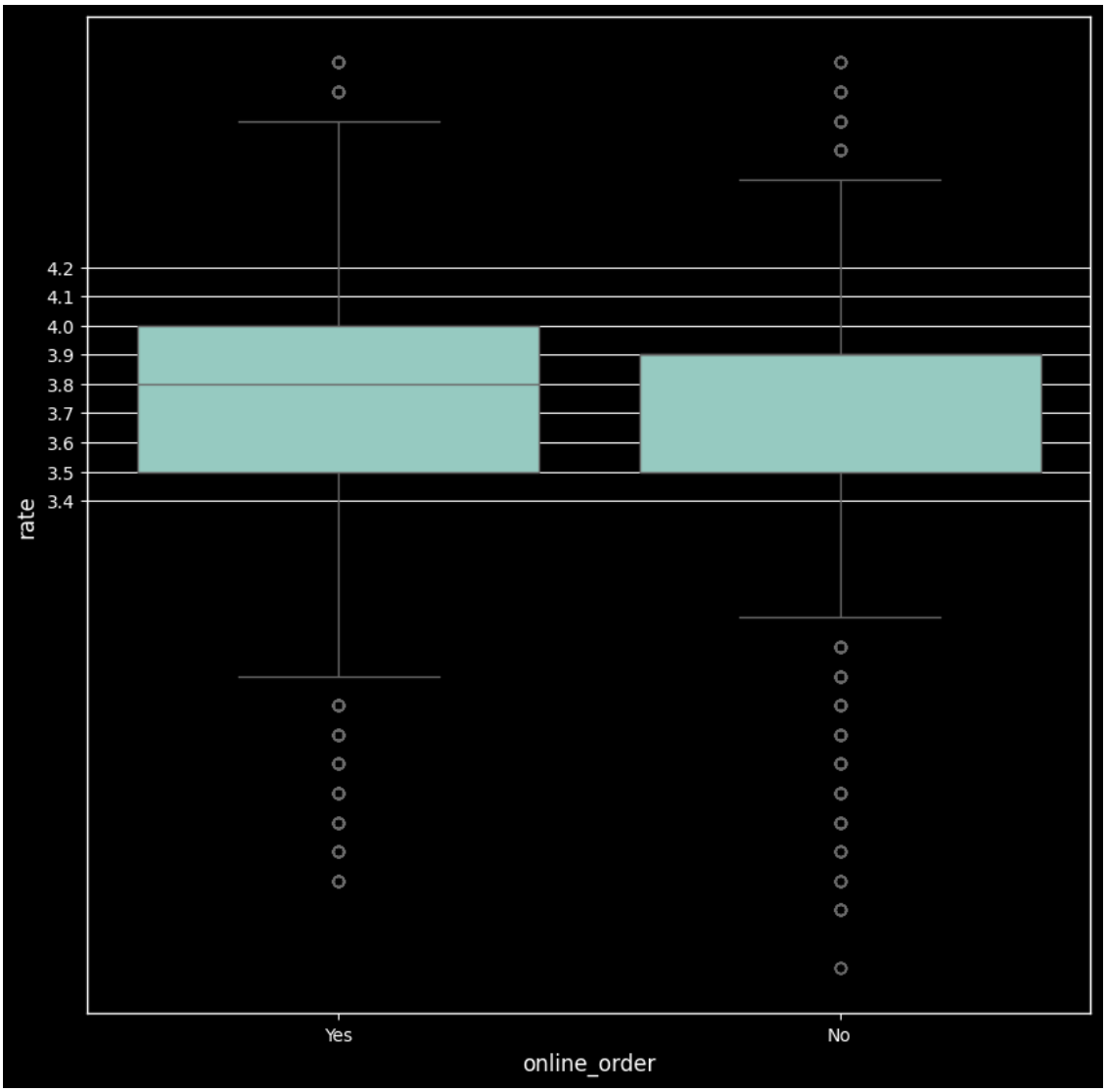
[53]:
```
plt.figure(figsize = (6,6))
sns.countplot(data=df,x="book_table")
plt.show()
```



2.1.4 online order and rating

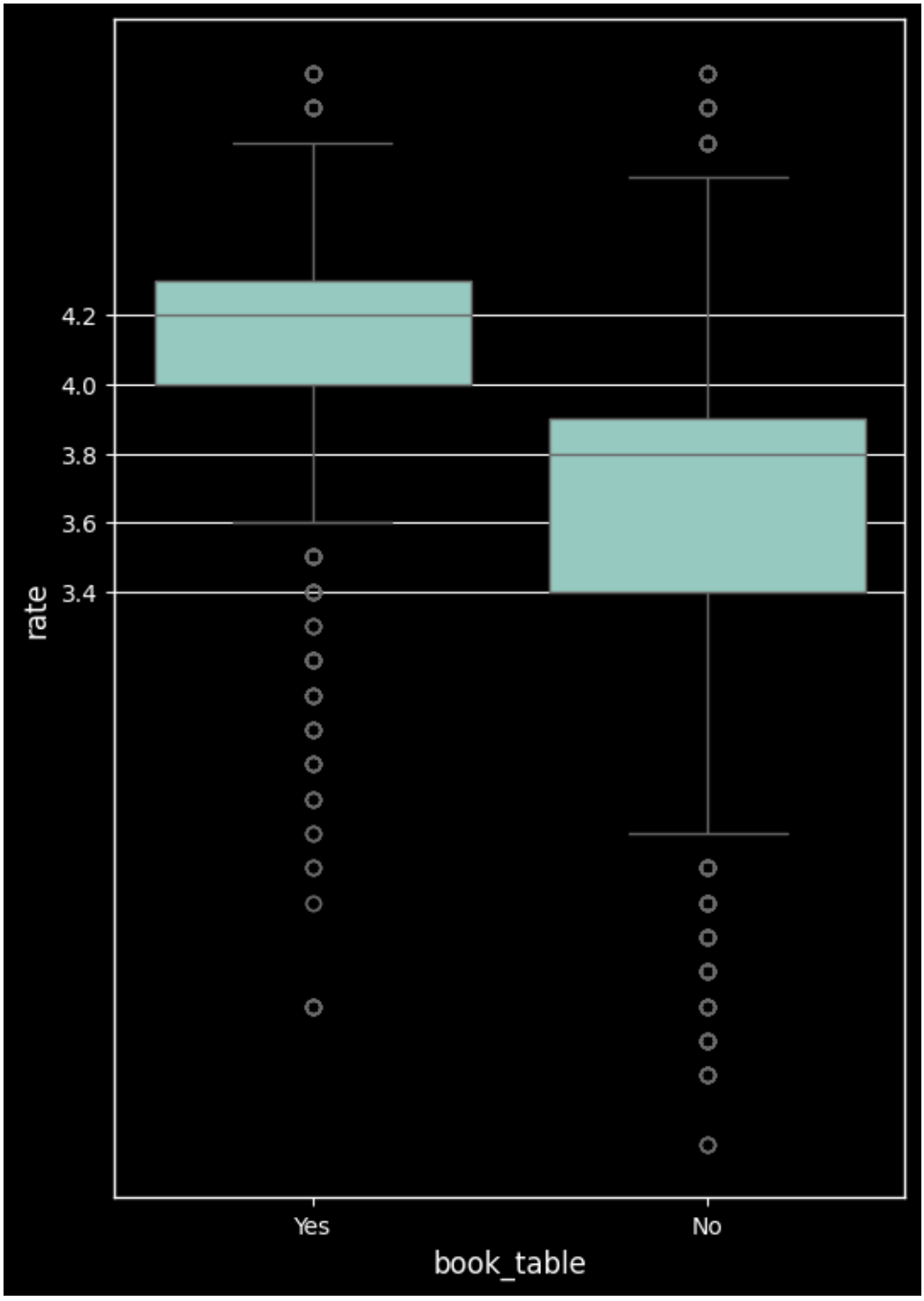
[54]:
```
plt.figure(figsize=(10,10))
sns.boxplot(data= df, x='online_order', y="rate")
li=[]
x=3.4
while(x<4.2):
    li.append(x)
    x+=.1

plt.yticks(li)
plt.show()
```



2.1.5 booktable vs rating

[55]:
```
plt.figure(figsize = (6,9))
sns.boxplot(data = df, x = "book_table", y= "rate")
plt.yticks([3.4,3.6,3.8,4.0, 4.2])
plt.show()
```


4.2
4.0
3.8
3.6
3.4
rate
Yes
No
book_table

[ ]: