

Loading the Dataset

```
In [1]: import numpy as np #NumPy is a general-purpose array-processing package.
import pandas as pd #It contains high-level data structures and manipulation tools des
import matplotlib.pyplot as plt #It is a Plotting Library
import seaborn as sns #Seaborn is a Python data visualization library based on matplot
from sklearn.linear_model import LogisticRegression #Logistic Regression is a Machine
from sklearn.linear_model import LinearRegression #Linear Regression is a Machine Lear
from sklearn.model_selection import train_test_split #Splitting of Dataset
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import r2_score
```

Reading the dataset

```
In [2]: #reading the dataset
zomato_orgnl=pd.read_csv("F:\Zomato.csv")
zomato_orgnl.head() #This function returns the first n rows for the object based on po
```

	url	address	name	online_order	book_table	rate	votes	
	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	42297555\rf97437
	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 417
	//www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 96634
	is://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 96200
	tps://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	8026612447\rf99012

Deleting Unnnecessary Columns

```
In [3]: #Deleting Unnnecessary Columns
zomato=zomato_orign1.drop(['url','dish_liked','phone'],axis=1)
```

Removing the Duplicates

```
In [4]: #Removing the Duplicates
zomato.duplicated().sum()
zomato.drop_duplicates(inplace=True)
```

Remove the NaN values from the dataset

```
In [5]: #Remove the NaN values from the dataset
zomato.isnull().sum()
zomato.dropna(how='any',inplace=True)
zomato.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43499 entries, 0 to 51716
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   address                             43499 non-null  object
1   name                                43499 non-null  object
2   online_order                        43499 non-null  object
3   book_table                          43499 non-null  object
4   rate                                43499 non-null  object
5   votes                               43499 non-null  int64
6   location                            43499 non-null  object
7   rest_type                           43499 non-null  object
8   cuisines                            43499 non-null  object
9   approx_cost(for two people)         43499 non-null  object
10  reviews_list                        43499 non-null  object
11  menu_item                           43499 non-null  object
12  listed_in(type)                     43499 non-null  object
13  listed_in(city)                     43499 non-null  object
dtypes: int64(1), object(13)
memory usage: 5.0+ MB
```

Changing the Columns Names

```
In [6]: #Changing the Columns Names
zomato.columns
zomato = zomato.rename(columns={'approx_cost(for two people)':'cost','listed_in(type)':'listed_in(city)':'city'})
zomato.columns
```

```
Out[6]: Index(['address', 'name', 'online_order', 'book_table', 'rate', 'votes',
              'location', 'rest_type', 'cuisines', 'cost', 'reviews_list',
              'menu_item', 'type', 'city'],
              dtype='object')
```

Some Transformations

```
In [7]: #Some Transformations
zomato['cost'] = zomato['cost'].astype(str)
zomato['cost'] = zomato['cost'].apply(lambda x: x.replace(',','.'))
```

```
zomato['cost'] = zomato['cost'].astype(float)
zomato.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43499 entries, 0 to 51716
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   address         43499 non-null  object
1   name            43499 non-null  object
2   online_order    43499 non-null  object
3   book_table      43499 non-null  object
4   rate            43499 non-null  object
5   votes           43499 non-null  int64
6   location        43499 non-null  object
7   rest_type       43499 non-null  object
8   cuisines        43499 non-null  object
9   cost            43499 non-null  float64
10  reviews_list    43499 non-null  object
11  menu_item       43499 non-null  object
12  type            43499 non-null  object
13  city            43499 non-null  object
dtypes: float64(1), int64(1), object(12)
memory usage: 5.0+ MB
```

Removing '/5' from Rates

```
In [8]: #Removing '/5' from Rates
zomato['rate'].unique()
zomato = zomato.loc[zomato.rate != 'NEW']
zomato = zomato.loc[zomato.rate != '-'].reset_index(drop=True)
remove_slash = lambda x: x.replace('/5', '') if type(x) == np.str else x
zomato.rate = zomato.rate.apply(remove_slash).str.strip().astype('float')
zomato['rate'].head()
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_1356\3634058824.py:5: DeprecationWarning: `np.str` is a deprecated alias for the builtin `str`. To silence this warning, use `str` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.str_` here.
Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
Out[8]: remove_slash = lambda x: x.replace('/5', '') if type(x) == np.str else x
0      4.1
1      4.1
2      3.8
3      3.7
4      3.8
Name: rate, dtype: float64
```

Adjust the column names

```
In [9]: # Adjust the column names
zomato.name = zomato.name.apply(lambda x:x.title())
zomato.online_order.replace(('Yes', 'No'), (True, False), inplace=True)
zomato.book_table.replace(('Yes', 'No'), (True, False), inplace=True)
zomato.cost.unique()
```

```
Out[9]: array([[800. , 300. , 600. , 700. , 550. , 500. , 450. , 650. ,
        400. , 900. , 200. , 750. , 150. , 850. , 100. , 1.2 ,
        350. , 250. , 950. , 1. , 1.5 , 1.3 , 199. , 1.1 ,
        1.6 , 230. , 130. , 1.7 , 1.35, 2.2 , 1.4 , 2. ,
        1.8 , 1.9 , 180. , 330. , 2.5 , 2.1 , 3. , 2.8 ,
        3.4 , 50. , 40. , 1.25, 3.5 , 4. , 2.4 , 2.6 ,
        1.45, 70. , 3.2 , 240. , 6. , 1.05, 2.3 , 4.1 ,
        120. , 5. , 3.7 , 1.65, 2.7 , 4.5 , 80. ]])
```

Encode the input Variables

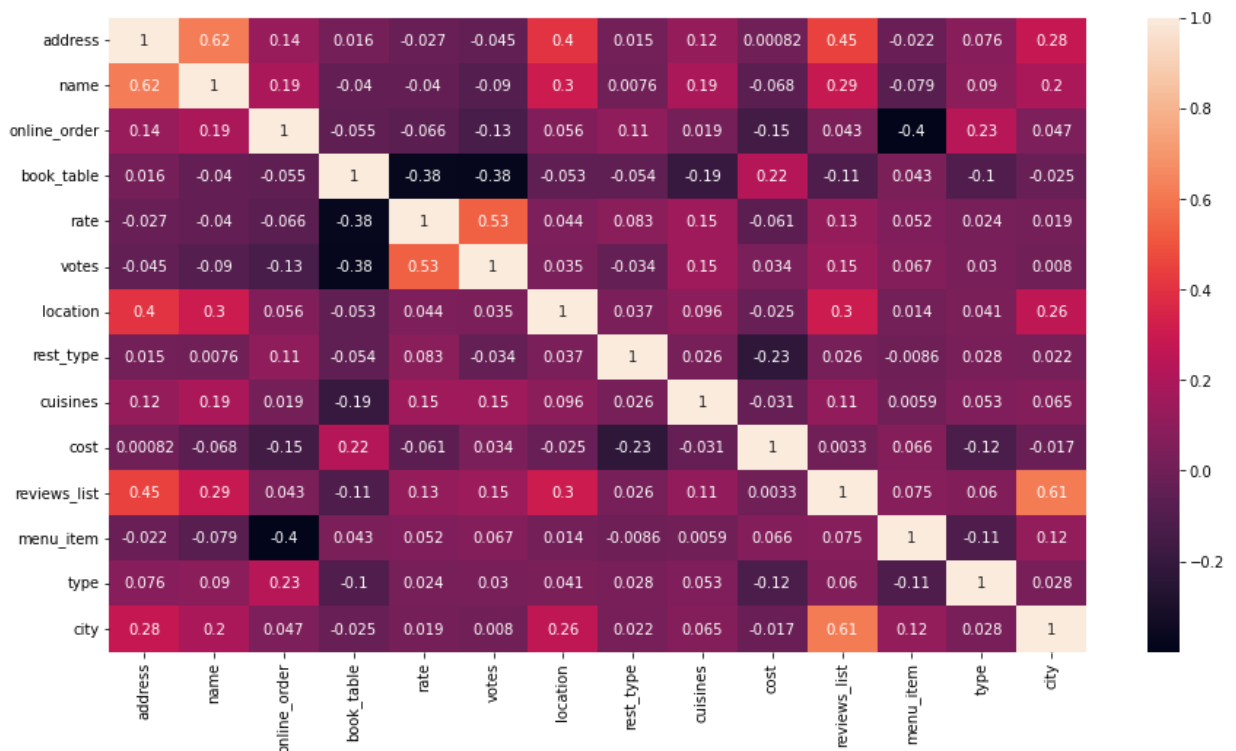
```
In [10]: #Encode the input Variables
def Encode(zomato):
    for column in zomato.columns[~zomato.columns.isin(['rate', 'cost', 'votes'])]:
        zomato[column] = zomato[column].factorize()[0]
    return zomato

zomato_en = Encode(zomato.copy())
```

Get Correlation between different variables

```
In [11]: #Get Correlation between different variables
corr = zomato_en.corr(method='kendall')
plt.figure(figsize=(15,8))
sns.heatmap(corr, annot=True)
zomato_en.columns
```

```
Out[11]: Index(['address', 'name', 'online_order', 'book_table', 'rate', 'votes',
        'location', 'rest_type', 'cuisines', 'cost', 'reviews_list',
        'menu_item', 'type', 'city'],
        dtype='object')
```



Defining the independent variables and dependent variables

```
In [13]: #Defining the independent variables and dependent variables
x = zomato_en.iloc[:,[2,3,5,6,7,8,9,11]]
y = zomato_en['rate']
#Getting Test and Training Set
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.1,random_state=353)
x_train.head()
```

```
Out[13]:
```

	online_order	book_table	votes	location	rest_type	cuisines	cost	menu_item
16950	0	1	0	8	2	5	250.0	0
767	0	1	131	8	4	278	400.0	190
6750	0	1	137	45	2	1295	250.0	0
9471	0	1	74	16	0	537	1.0	0
25162	0	1	61	12	2	1860	350.0	0

```
In [14]: y_train.head()
```

```
Out[14]: 16950    3.9
767      3.7
6750     4.0
9471     3.8
25162    3.7
Name: rate, dtype: float64
```

Regression Analysis

Linear Regression

```
In [15]: #Prepare a Linear REgression Model
reg=LinearRegression()
reg.fit(x_train,y_train)
y_pred=reg.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

```
Out[15]: 0.27362337221038613
```

Decision Tree Regression

```
In [16]: #Prepairng a Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.1,random_state=105)
DTree=DecisionTreeRegressor(min_samples_leaf=.0001)
DTree.fit(x_train,y_train)
y_predict=DTree.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

```
Out[16]: 0.8544435619824873
```

Random Forest Regression

```
In [19]: #Preparing Random Forest REgression
from sklearn.ensemble import RandomForestRegressor
RForest=RandomForestRegressor(n_estimators=500,random_state=329,min_samples_leaf=.0001)
RForest.fit(x_train,y_train)
y_predict=RForest.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

Out[19]: 0.8774279501872895

Extra Tree Regression

```
In [20]: #Preparing Extra Tree Regression
from sklearn.ensemble import ExtraTreesRegressor
ETree=ExtraTreesRegressor(n_estimators = 100)
ETree.fit(x_train,y_train)
y_predict=ETree.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

Out[20]: 0.9384354116714239

Data Visualization

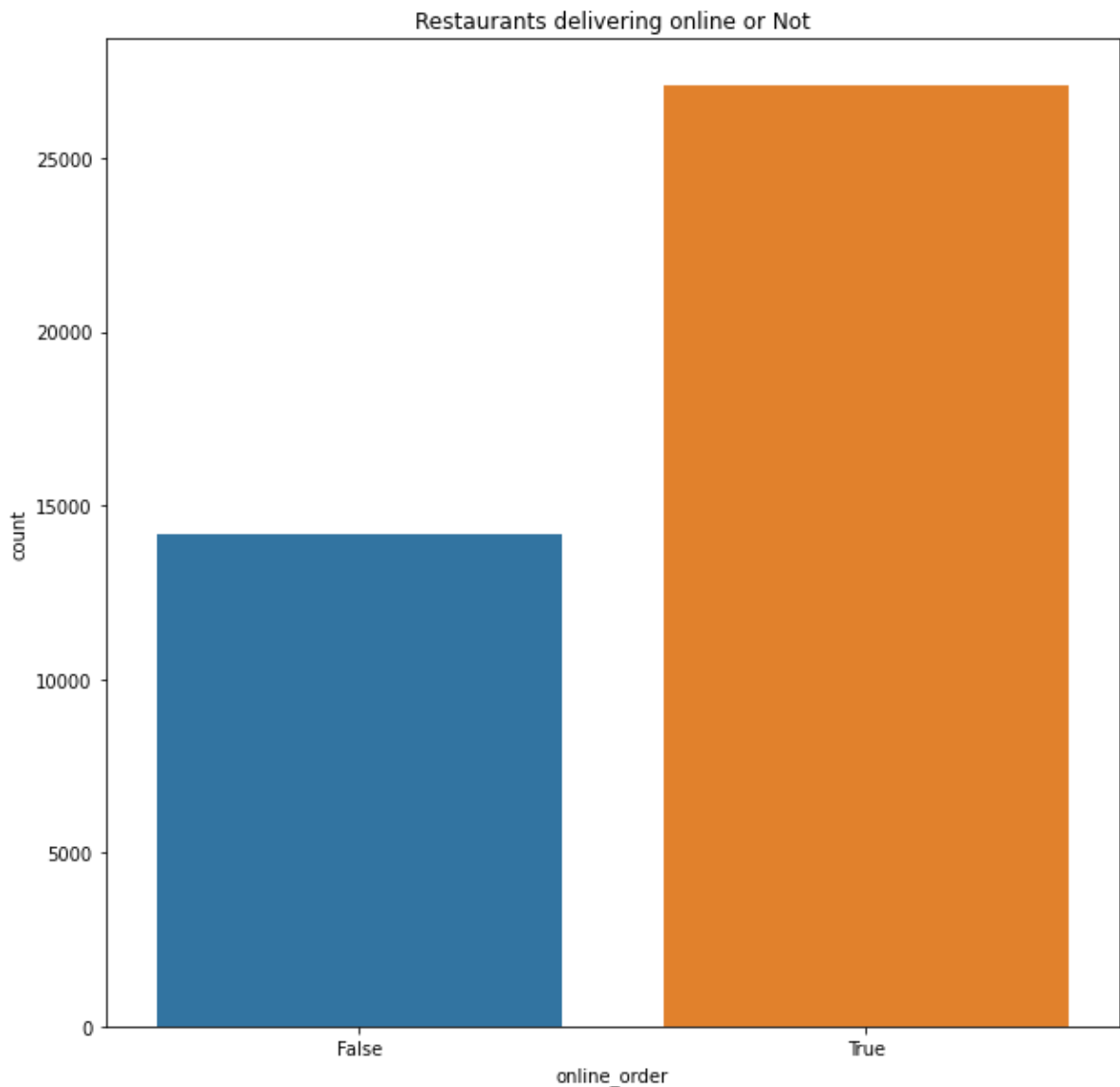
Restaurants delivering Online or not

```
In [21]: #Restaurants delivering Online or not
sns.countplot(zomato['online_order'])
fig = plt.gcf()
fig.set_size_inches(10,10)
plt.title('Restaurants delivering online or Not')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

Out[21]: Text(0.5, 1.0, 'Restaurants delivering online or Not')



Restaurants allowing table booking or not

```
In [22]: #Restaurants allowing table booking or not
sns.countplot(zomato['book_table'])
fig = plt.gcf()
fig.set_size_inches(10,10)
plt.title('Restaurants allowing table booking or not')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[22]: Text(0.5, 1.0, 'Restaurants allowing table booking or not')
```

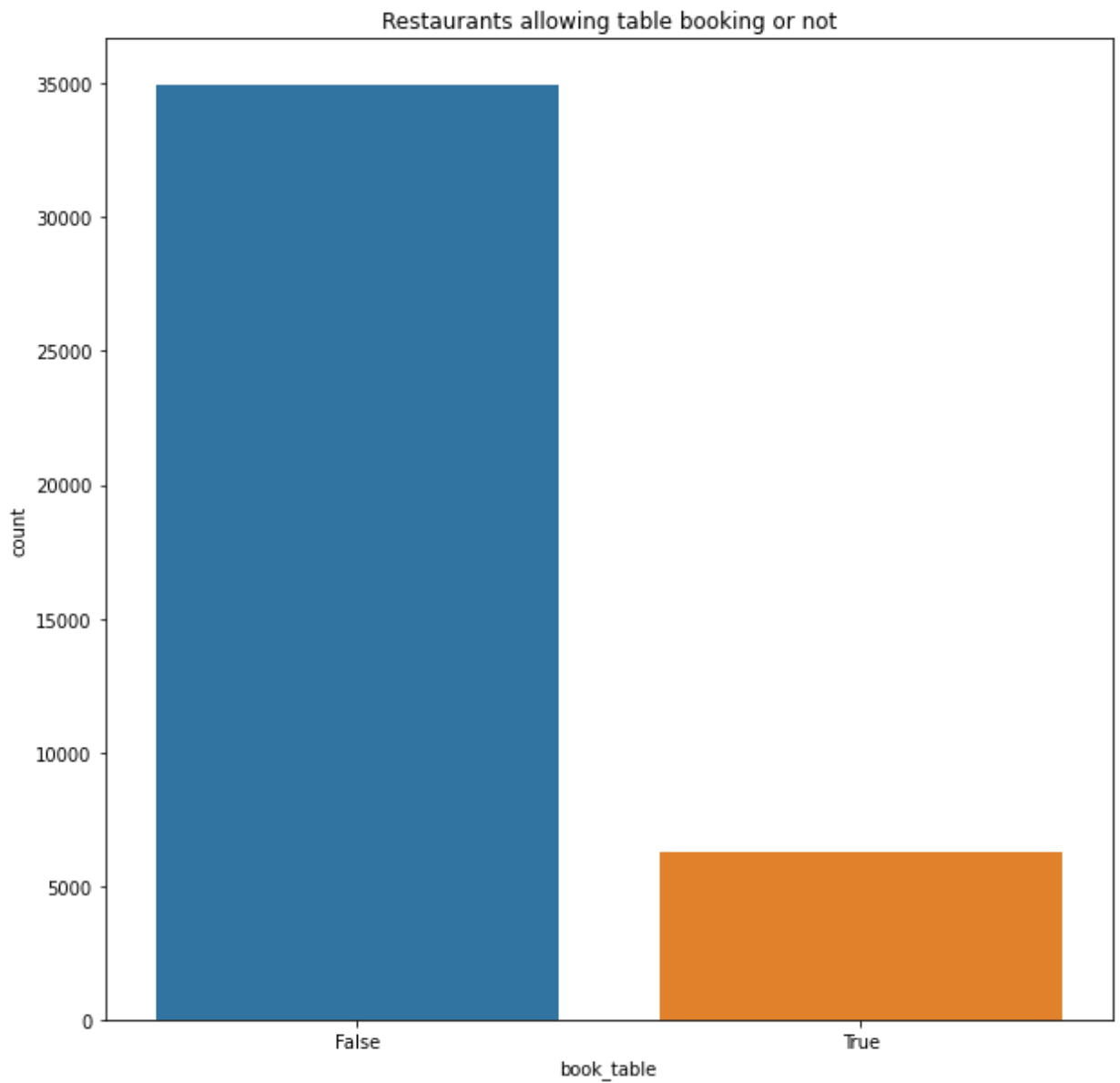
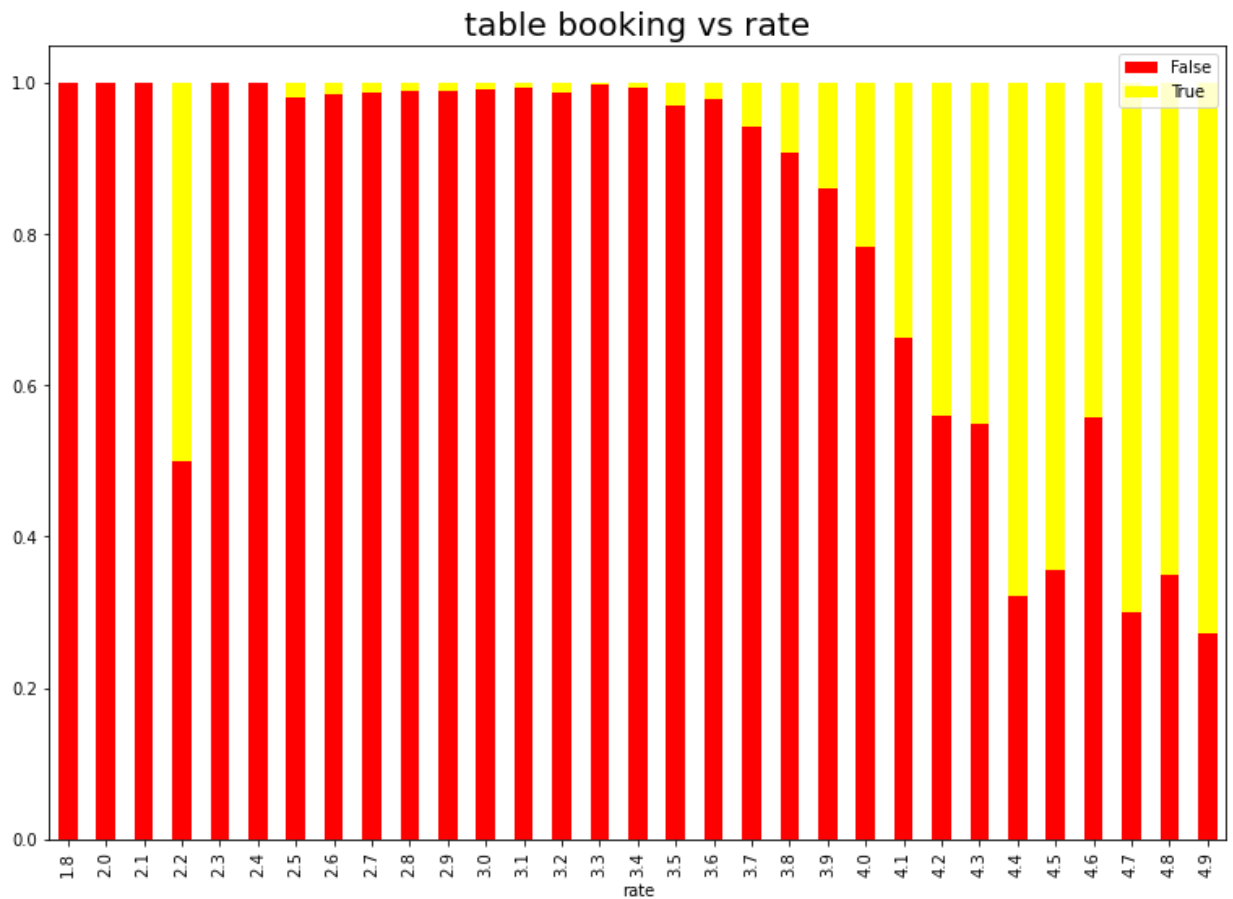


Table booking Rate vs Rate

```
In [23]: #Table booking Rate vs Rate
plt.rcParams['figure.figsize'] = (13, 9)
Y = pd.crosstab(zomato['rate'], zomato['book_table'])
Y.div(Y.sum(1).astype(float), axis = 0).plot(kind = 'bar', stacked = True,color=['red', 'blue'])
plt.title('table booking vs rate', fontweight = 30, fontsize = 20)
plt.legend(loc="upper right")
plt.show()
```

Location

```
In [24]: # Location
sns.countplot(zomato['city'])
sns.countplot(zomato['city']).set_xticklabels(sns.countplot(zomato['city']).get_xticklabels())
fig = plt.gcf()
fig.set_size_inches(13,13)
plt.title('Location')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

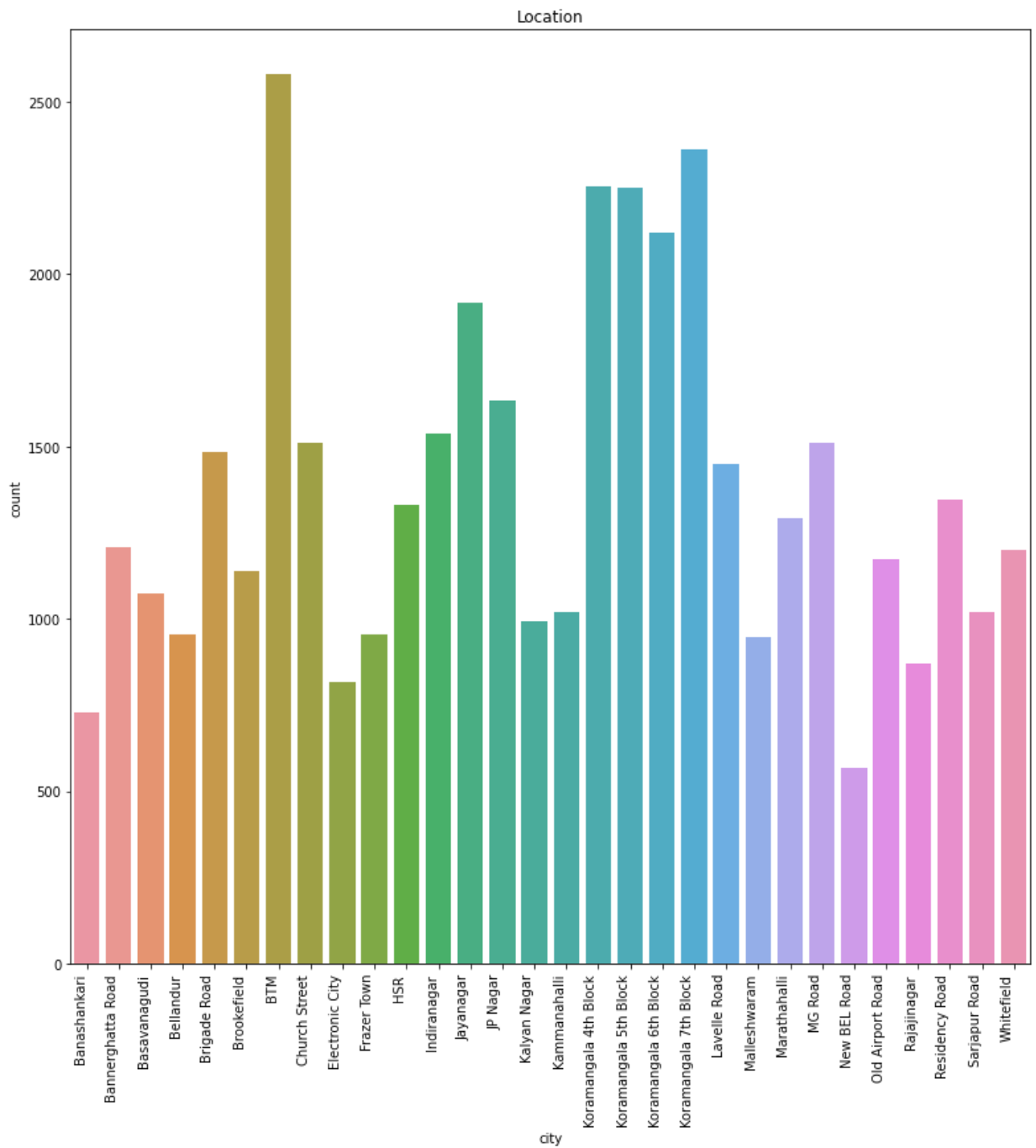
C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

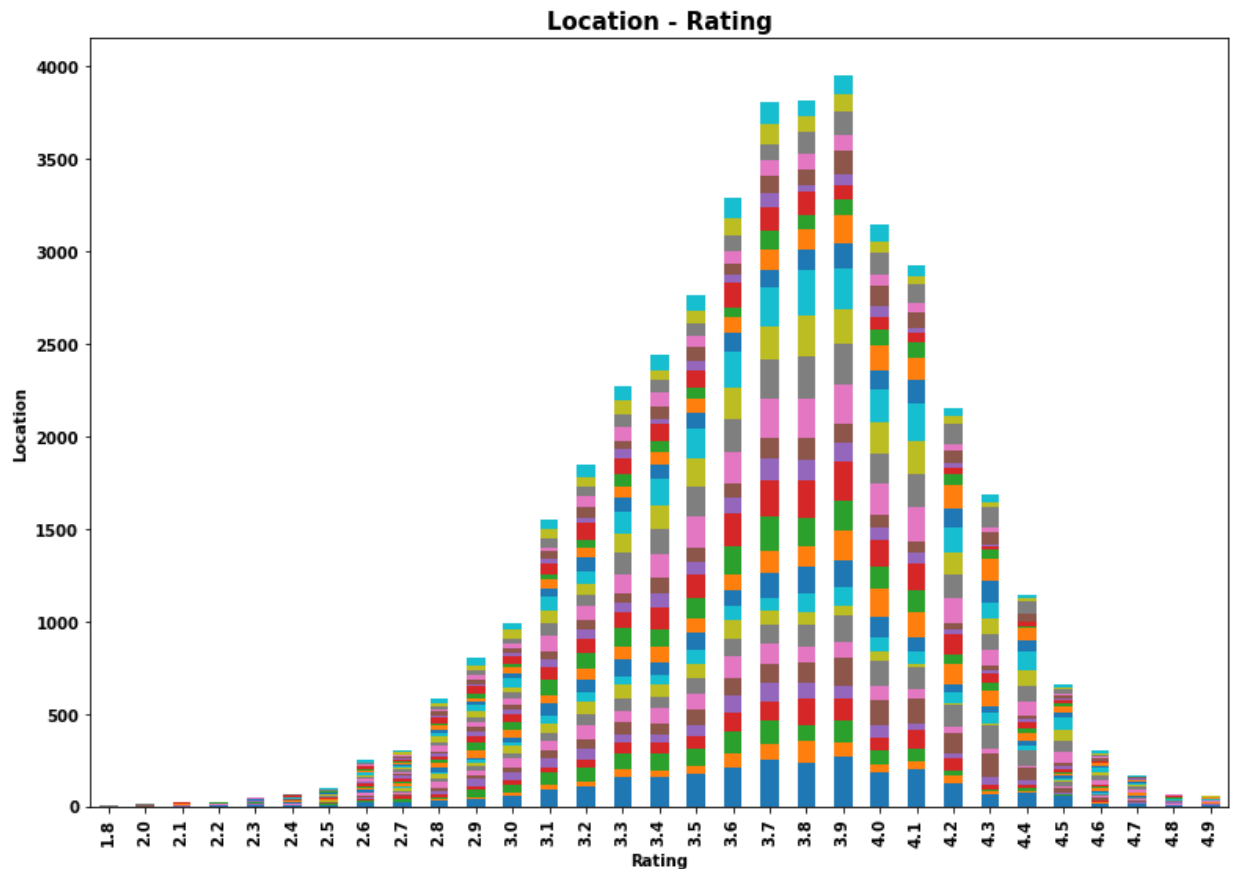
warnings.warn(

```
Out[24]: Text(0.5, 1.0, 'Location')
```



Location and Rating

```
In [25]: #Location and Rating
loc_plt=pd.crosstab(zomato['rate'],zomato['city'])
loc_plt.plot(kind='bar',stacked=True);
plt.title('Location - Rating',fontsize=15,fontweight='bold')
plt.ylabel('Location',fontsize=10,fontweight='bold')
plt.xlabel('Rating',fontsize=10,fontweight='bold')
plt.xticks(fontsize=10,fontweight='bold')
plt.yticks(fontsize=10,fontweight='bold');
plt.legend().remove();
```



Restaurant Type

```
In [26]: #Restaurant Type
sns.countplot(zomato['rest_type'])
sns.countplot(zomato['rest_type']).set_xticklabels(sns.countplot(zomato['rest_type']).
fig = plt.gcf()
fig.set_size_inches(15,15)
plt.title('Restuarant Type')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

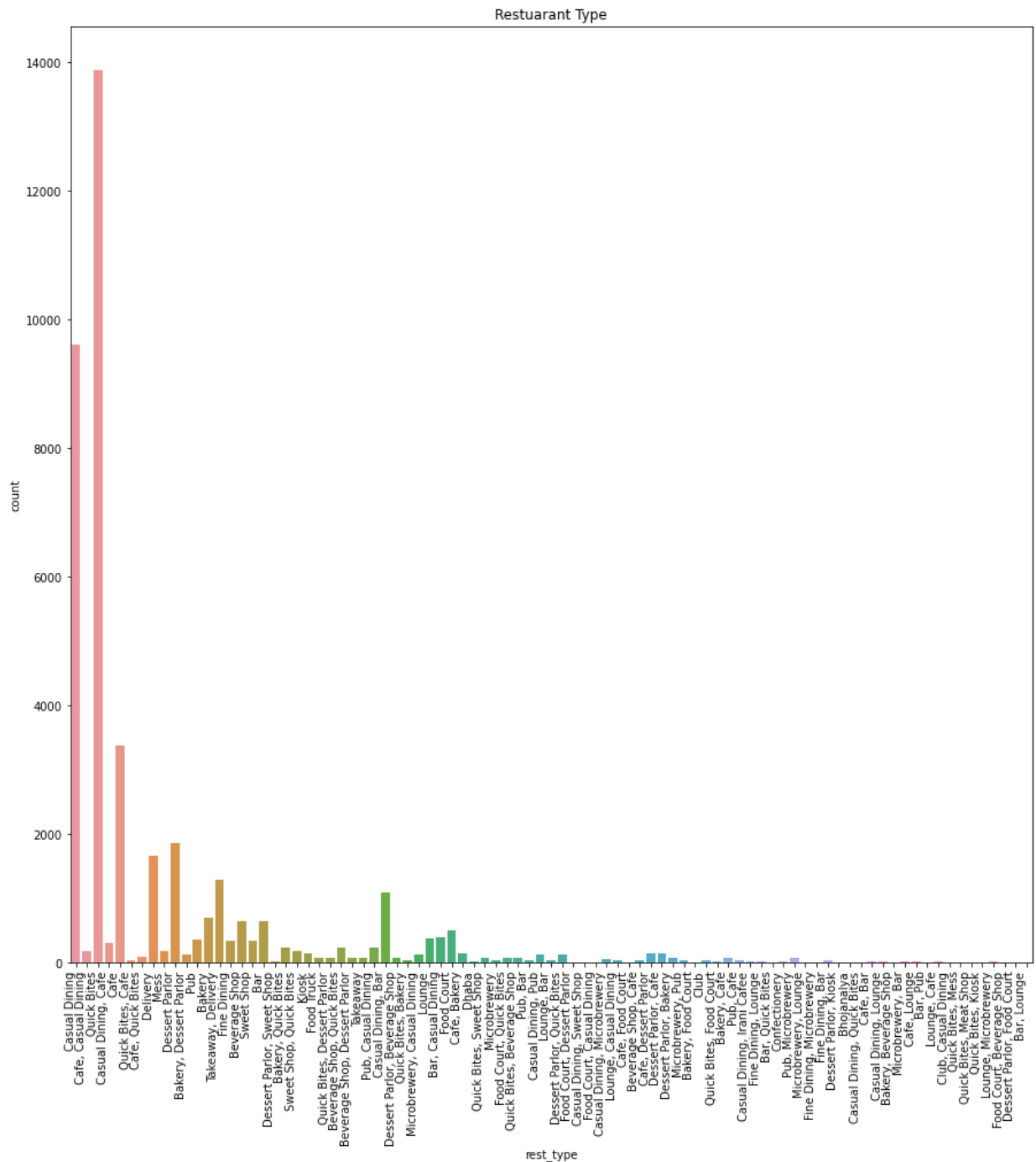
C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

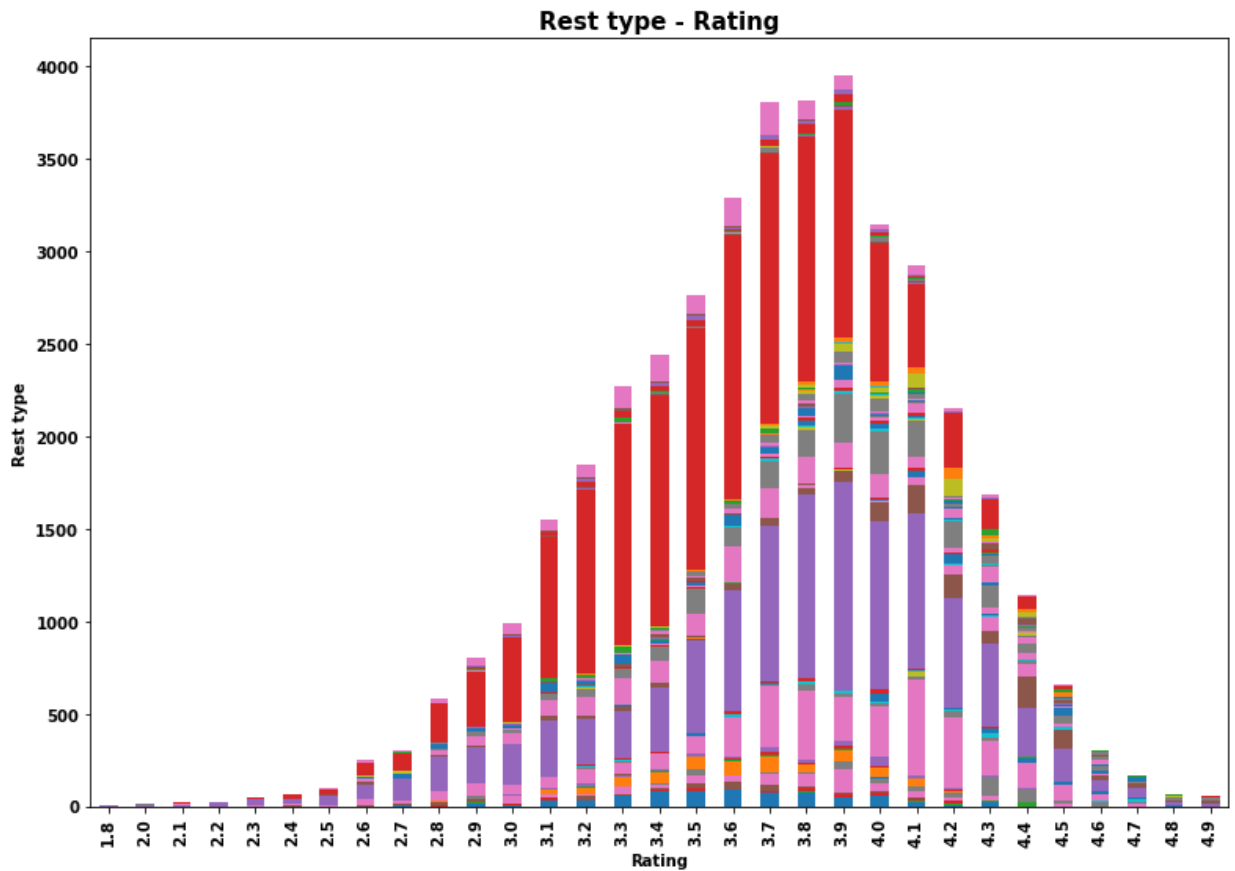
warnings.warn(

```
Out[26]: Text(0.5, 1.0, 'Restuarant Type')
```



Gaussian Rest type and Rating

```
In [27]: #Gaussian Rest type and Rating
loc_plt=pd.crosstab(zomato['rate'],zomato['rest_type'])
loc_plt.plot(kind='bar',stacked=True);
plt.title('Rest type - Rating',fontsize=15,fontweight='bold')
plt.ylabel('Rest type',fontsize=10,fontweight='bold')
plt.xlabel('Rating',fontsize=10,fontweight='bold')
plt.xticks(fontsize=10,fontweight='bold')
plt.yticks(fontsize=10,fontweight='bold');
plt.legend().remove();
```



Types of Services

```
In [28]: #Types of Services
sns.countplot(zomato['type'])
sns.countplot(zomato['type']).set_xticklabels(sns.countplot(zomato['type']).get_xtickl
fig = plt.gcf()
fig.set_size_inches(15,15)
plt.title('Type of Service')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

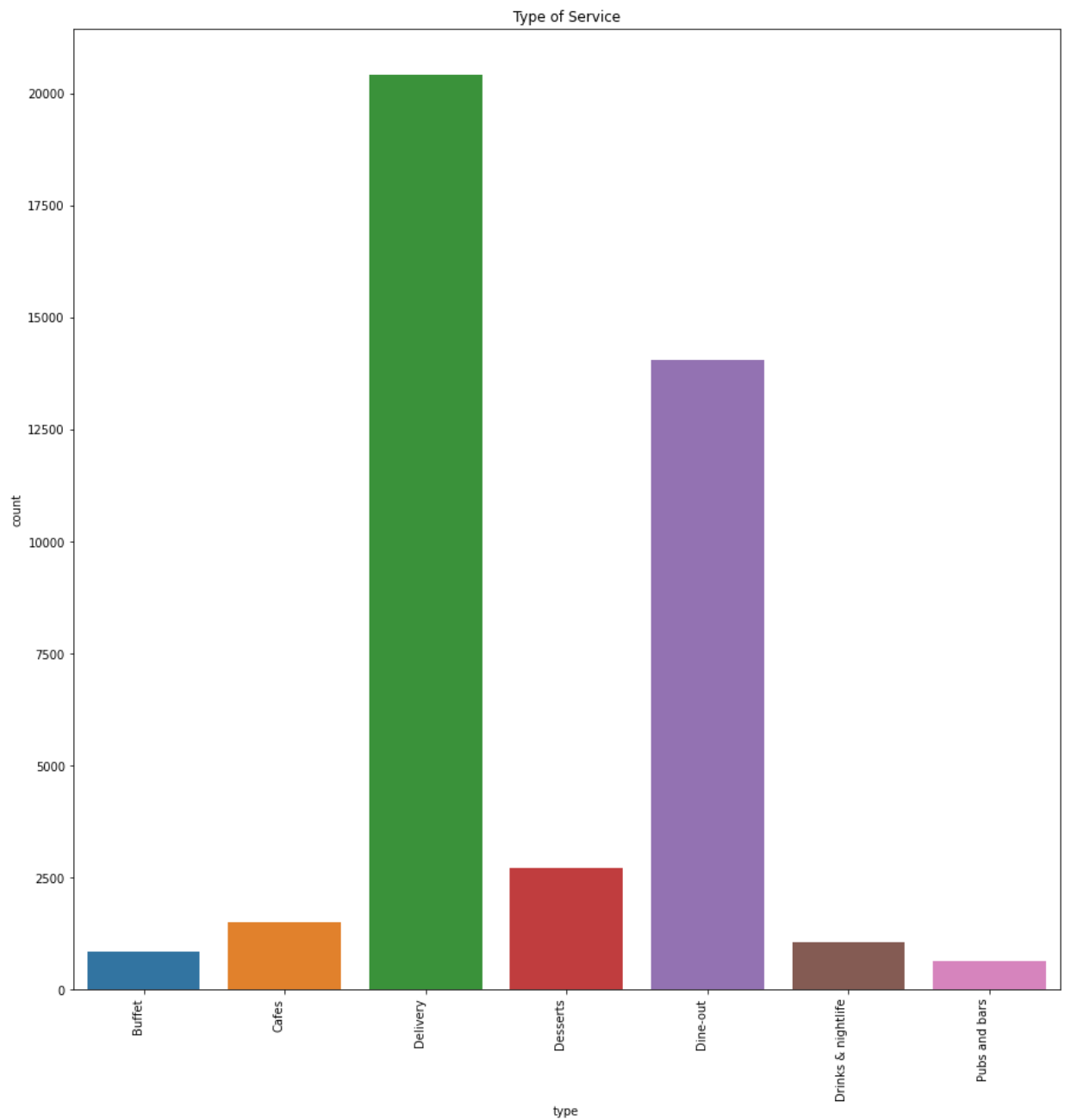
C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

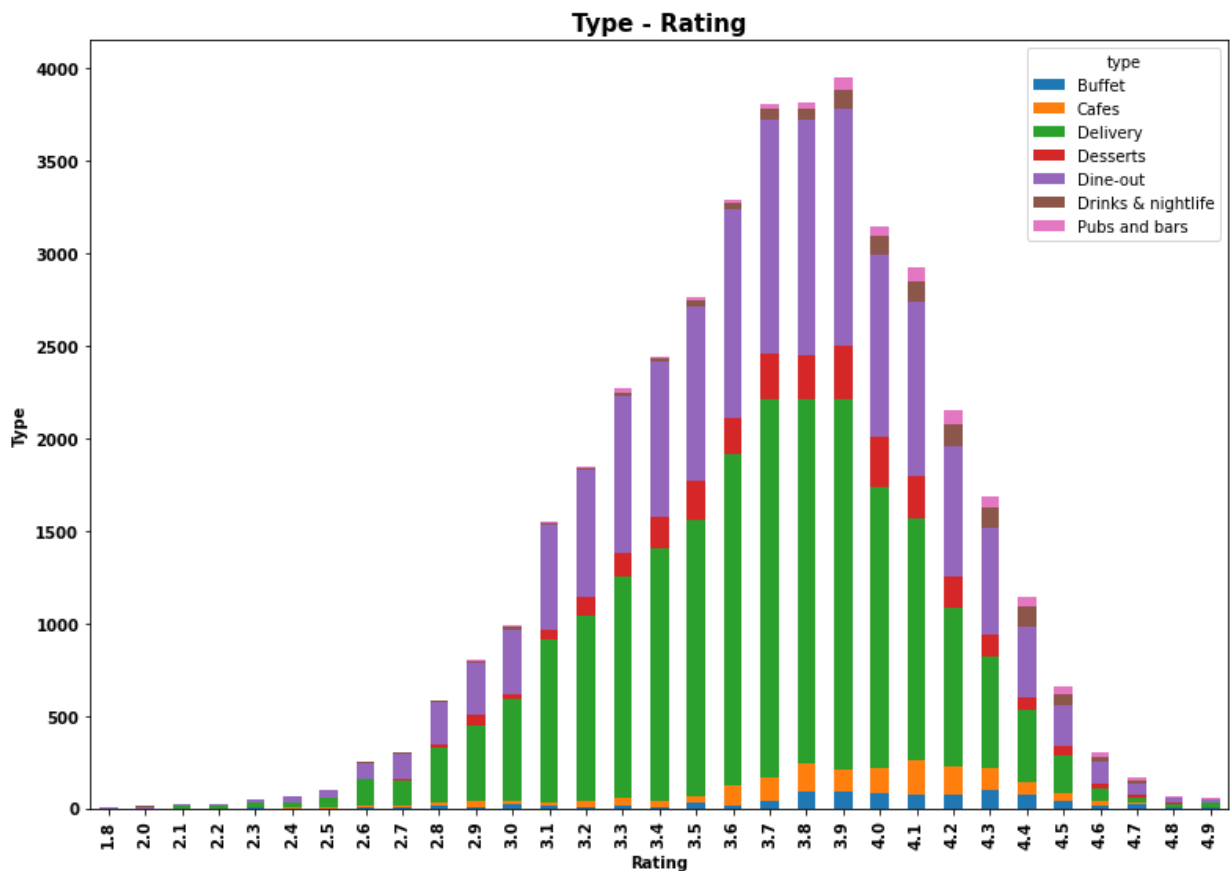
warnings.warn(

```
Out[28]: Text(0.5, 1.0, 'Type of Service')
```



Type and Rating

```
In [29]: #Type and Rating
type_plt=pd.crosstab(zomato['rate'],zomato['type'])
type_plt.plot(kind='bar',stacked=True);
plt.title('Type - Rating',fontsize=15,fontweight='bold')
plt.ylabel('Type',fontsize=10,fontweight='bold')
plt.xlabel('Rating',fontsize=10,fontweight='bold')
plt.xticks(fontsize=10,fontweight='bold')
plt.yticks(fontsize=10,fontweight='bold');
```



Cost of Restuarant

```
In [30]: #Cost of Restuarant
sns.countplot(zomato['cost'])
sns.countplot(zomato['cost']).set_xticklabels(sns.countplot(zomato['cost']).get_xtickl
fig = plt.gcf()
fig.set_size_inches(15,15)
plt.title('Cost of Restuarant')
```

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

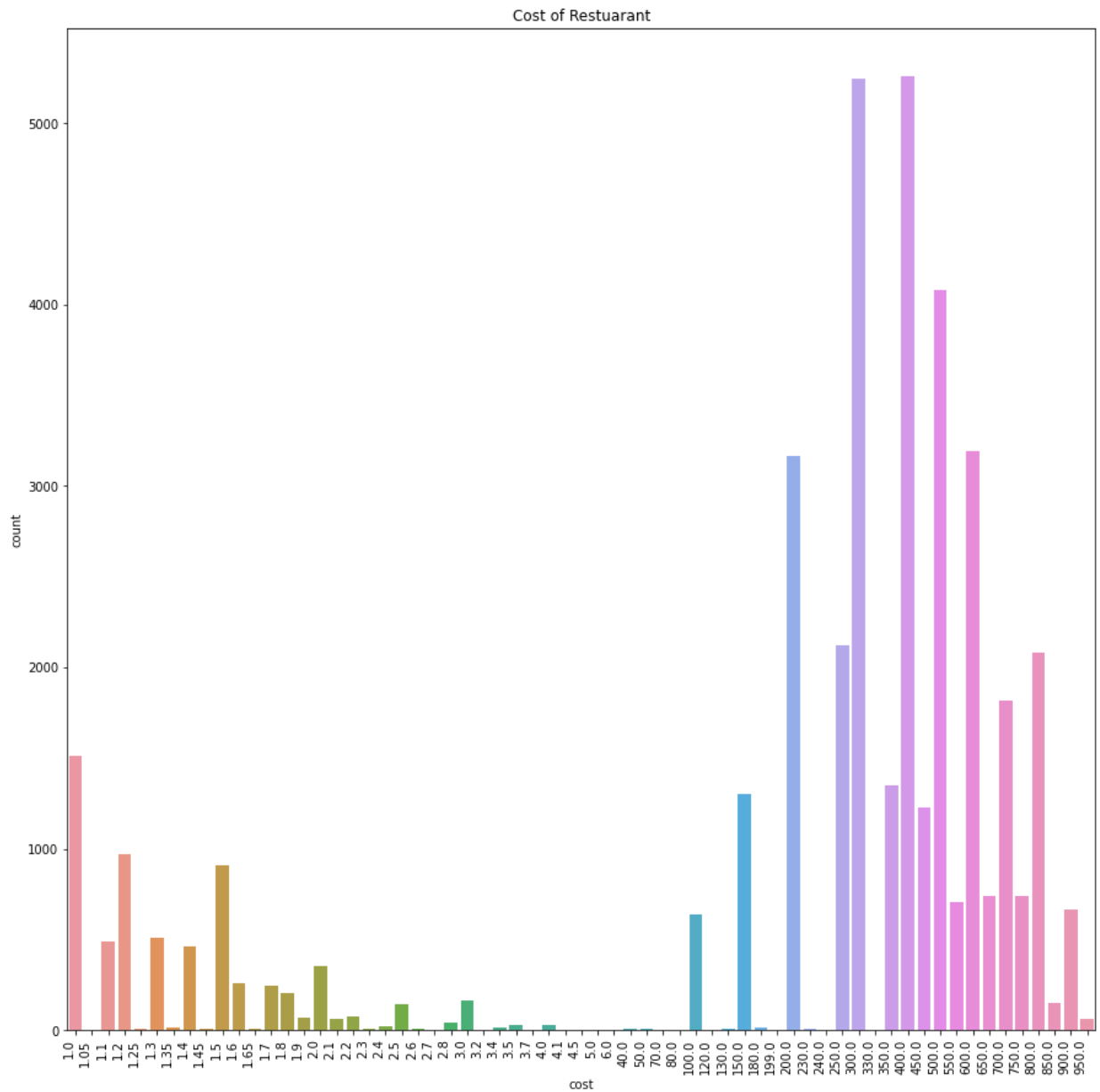
C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

C:\Users\LENOVO\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

```
Out[30]: Text(0.5, 1.0, 'Cost of Restuarant')
```

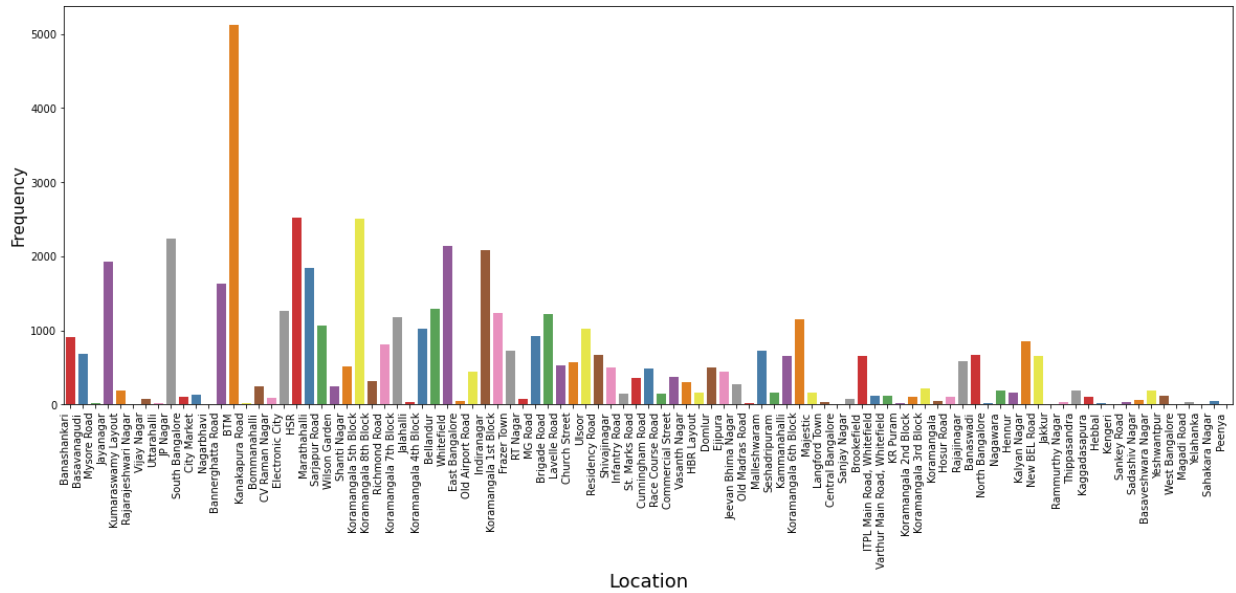


NO. of restaurants in a Location

```
In [31]: #NO. of restaurants in a Location
fig = plt.figure(figsize=(20,7))
loc = sns.countplot(x="location",data=zomato_orgnl, palette = "Set1")
loc.set_xticklabels(loc.get_xticklabels(), rotation=90, ha="right")
plt.ylabel("Frequency",size=15)
plt.xlabel("Location",size=18)
loc
plt.title('NO. of restaurants in a Location',size = 20,pad=20)
```

```
Out[31]: Text(0.5, 1.0, 'NO. of restaurants in a Location')
```


NO. of restaurants in a Location

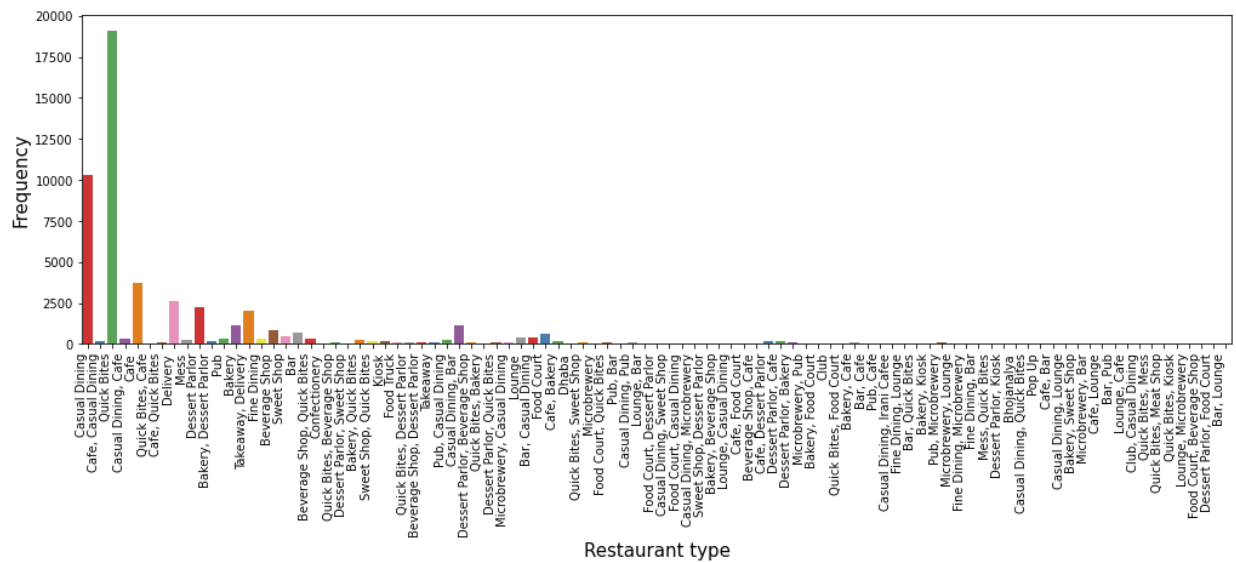


Restaurant type

```
In [32]: #Restaurant type
fig = plt.figure(figsize=(17,5))
rest = sns.countplot(x="rest_type",data=zomato_orgn1, palette = "Set1")
rest.set_xticklabels(rest.get_xticklabels(), rotation=90, ha="right")
plt.ylabel("Frequency",size=15)
plt.xlabel("Restaurant type",size=15)
rest
plt.title('Restaurant types',fontsize = 20 ,pad=20)
```

```
Out[32]: Text(0.5, 1.0, 'Restaurant types')
```

Restaurant types



Most famous restaurant chains in Bengaluru

```
In [33]: #Most famous restaurant chains in Bengaluru
plt.figure(figsize=(15,7))
chains=zomato_orgn1['name'].value_counts()[ :20]
```

```
sns.barplot(x=chains,y=chains.index,palette='Set1')
plt.title("Most famous restaurant chains in Bangaluru",size=20,pad=20)
plt.xlabel("Number of outlets",size=15)
```

Out[33]: Text(0.5, 0, 'Number of outlets')

