

## ✓ 1. Know Your Data

### ✓ Import Libraries

```
!pip uninstall -y numpy scipy gensim
```

```

⇒ Found existing installation: numpy 1.26.4
Uninstalling numpy-1.26.4:
  Successfully uninstalled numpy-1.26.4
Found existing installation: scipy 1.13.1
Uninstalling scipy-1.13.1:
  Successfully uninstalled scipy-1.13.1
Found existing installation: gensim 4.3.3
Uninstalling gensim-4.3.3:
  Successfully uninstalled gensim-4.3.3

```

```
!pip install numpy==1.26.4 scipy==1.13.1 gensim==4.3.3
```

```

⇒ Collecting numpy==1.26.4
  Using cached numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Collecting scipy==1.13.1
  Using cached scipy-1.13.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Collecting gensim==4.3.3
  Using cached gensim-4.3.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.11/dist-packages (from gensim==4.3.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages (from gensim==4.3.3)
Using cached numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Using cached scipy-1.13.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Using cached gensim-4.3.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Installing collected packages: numpy, scipy, gensim
Successfully installed gensim-4.3.3 numpy-1.26.4 scipy-1.13.1

```

```

# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import matplotlib.cm as cm
import seaborn as sns
import math
import time
from wordcloud import WordCloud

from scipy.stats import norm
from scipy import stats
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.linear_model import LogisticRegression

```

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import MinMaxScaler, StandardScaler

#importing kmeans
from sklearn.cluster import KMeans

#importing random forest and XgB
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

#Non-negative matrix Factorization
from sklearn.decomposition import NMF

from sklearn.naive_bayes import MultinomialNB

#principal component analysis
from sklearn.decomposition import PCA

#silhouette score
from sklearn.metrics import silhouette_score
from sklearn.model_selection import ParameterGrid

#importing stopwords
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.corpus import stopwords
#for tokenization
from nltk.tokenize import word_tokenize
# for POS tagging(Part of speech in NLP sentiment analysis)
nltk.download('averaged_perceptron_tagger')

#import stemmer
from nltk.stem.snowball import SnowballStemmer

#import tfidf
from sklearn.feature_extraction.text import TfidfVectorizer

#LDA

from sklearn.decomposition import LatentDirichletAllocation

#importing contraction
!pip install contractions
!pip install gensim
import gensim
from gensim import corpora

#importing shap for model explainability
```

```
!pip install shap
import shap
```

```
#download small spacy model
# !python -m spacy download en_core_web_sm
# import spacy
```

```
# The following lines adjust the granularity of reporting.
pd.options.display.float_format = "{:.2f}".format
```

```
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

```

[➡] [nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /root/nltk_data...
[nltk_data] Package averaged_perceptron_tagger is already up-to-
[nltk_data] date!
Requirement already satisfied: contractions in /usr/local/lib/python3.11/dist-
Requirement already satisfied: textsearch>=0.0.21 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: anyascii in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: pyahocorasick in /usr/local/lib/python3.11/dist-
Requirement already satisfied: gensim in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numpy<2.0,>=1.18.5 in /usr/local/lib/python3.11/di
Requirement already satisfied: scipy<1.14.0,>=1.7.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: shap in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.11/di
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.11/dis
Requirement already satisfied: numba>=0.54 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.11/dist-
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/p
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pytho
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/di
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pack

```

## ▼ Dataset Loading

```
# Load Dataset
hotel_df = pd.read_csv('/content/drive/MyDrive/Zomato data/Zomato Restaurant name
review_df = pd.read_csv('/content/drive/MyDrive/Zomato data/Zomato Restaurant rev
```

Dataset First View

```
# Dataset First Look restaurant
hotel_df.head()
```

	Name	Links	Cost	Collections	Cuisine
0	Beyond Flavours	https://www.zomato.com/hyderabad/beyond-flavou...	800	Food Hygiene Rated Restaurants in Hyderabad, C...	Chinese Continental Kebabs European South I.
1	Paradise	https://www.zomato.com/hyderabad/paradise-gach...	800	Hyderabad's Hottest	Biryani, North Indian Chinese

Next steps:

[Generate code with hotel\\_df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
# Dataset First Look review
review_df.head()
```

	Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures
0	Beyond Flavours	Rusha Chakraborty	The ambience was good, food was quite good . h...	5	1 Review , 2 Followers	5/25/2019 15:54	0
	Beyond	Rusha	Ambience is too good for		3 Reviews	5/25/2019	

Next steps:

[Generate code with review\\_df](#)

[View recommended plots](#)

[New interactive sheet](#)

Dataset Rows & Columns count

```
# Dataset Rows(Observation) & Columns count(Feature)
print(f'Total observation and feature for restaurant: {hotel_df.shape}')
print(f'Total observation and feature for review: {review_df.shape}')
```

```
➞ Total observation and feature for restaurant: (105, 6)
Total observation and feature for review: (10000, 7)
```

## ✓ Dataset Information

```
# Dataset Info
print('Restaurant Info')
print('\n')
hotel_df.info()
print('='*120)
print('\n')
print('Review Info')
print('\n')
review_df.info()
```

```
➞ Restaurant Info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Name             105 non-null   object
1   Links            105 non-null   object
2   Cost             105 non-null   object
3   Collections      51 non-null    object
4   Cuisines          105 non-null   object
5   Timings          104 non-null   object
```

```
dtypes: object(6)
```

```
memory usage: 5.1+ KB
```

```
Review Info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Restaurant      10000 non-null  object
1   Reviewer        9962 non-null   object
2   Review          9955 non-null   object
3   Rating          9962 non-null   object
4   Metadata        9962 non-null   object
5   Time            9962 non-null   object
6   Pictures        10000 non-null  int64
```

```
dtypes: int64(1), object(6)
```

```
memory usage: 547.0+ KB
```

## ▼ Duplicate Values

```
# Dataset Duplicate Value Count
print('For Restaurant')
print('\n')
print(f"Data is duplicated ? {hotel_df.duplicated().value_counts()},unique values")
print('\n')
print('='*120)
print('\n')
print('For Reviews')
print('\n')
print(f"Data is duplicated ? {review_df.duplicated().value_counts()},unique value")
```

↔ For Restaurant

```
Data is duplicated ? False      105
Name: count, dtype: int64,unique values with 0 duplication
```

=====

For Reviews

```
Data is duplicated ? False      9964
True                36
Name: count, dtype: int64,unique values with 36 duplication
```

```
#getting duplicate values
print(f' Duplicate data count = {review_df[review_df.duplicated()].shape[0]}')
review_df[review_df.duplicated()]
```



<b>8781</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8782</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8783</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8784</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8785</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8786</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8787</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8788</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8789</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8790</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8791</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8792</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8793</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8794</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8795</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8796</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0
<b>8797</b>	American Wild Wings	NaN	NaN	NaN	NaN	NaN	0

American Wild

```
8798 American Wild Wings NaN NaN NaN NaN NaN 0
8799 American Wild Wings NaN NaN NaN NaN NaN 0
9086 Arena Eleven NaN NaN NaN NaN NaN 0
9087 Arena Eleven NaN NaN NaN NaN NaN 0
9088 Arena Eleven NaN NaN NaN NaN NaN 0

#checking values for American Wild Things
review_df[(review_df['Restaurant'] == 'American Wild Wings')].shape
(100, 7)
9090 Arena Eleven NaN NaN NaN NaN NaN 0
9091 Arena Eleven NaN NaN NaN NaN NaN 0

#checking values for Arena Eleven
review_df[(review_df['Restaurant'] == 'Arena Eleven')].shape
(100, 7)
```

Missing Values/Null Values

```
# Missing Values/Null Values Count for hotel data
hotel_df.isnull().sum()

0
Name 0
Links 0
Cost 0
Collections 54
Cuisines 0
Timings 1

dtype: int64

# Missing Values/Null Values Count for review data
review_df.isnull().sum()
```



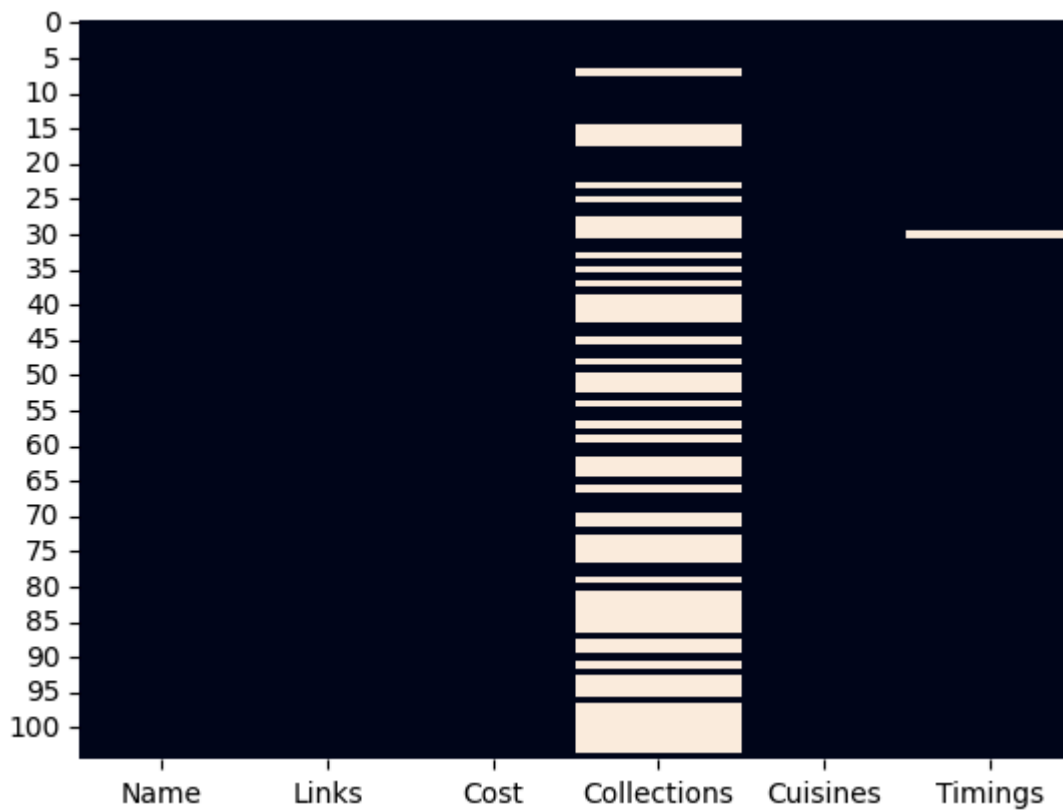


0

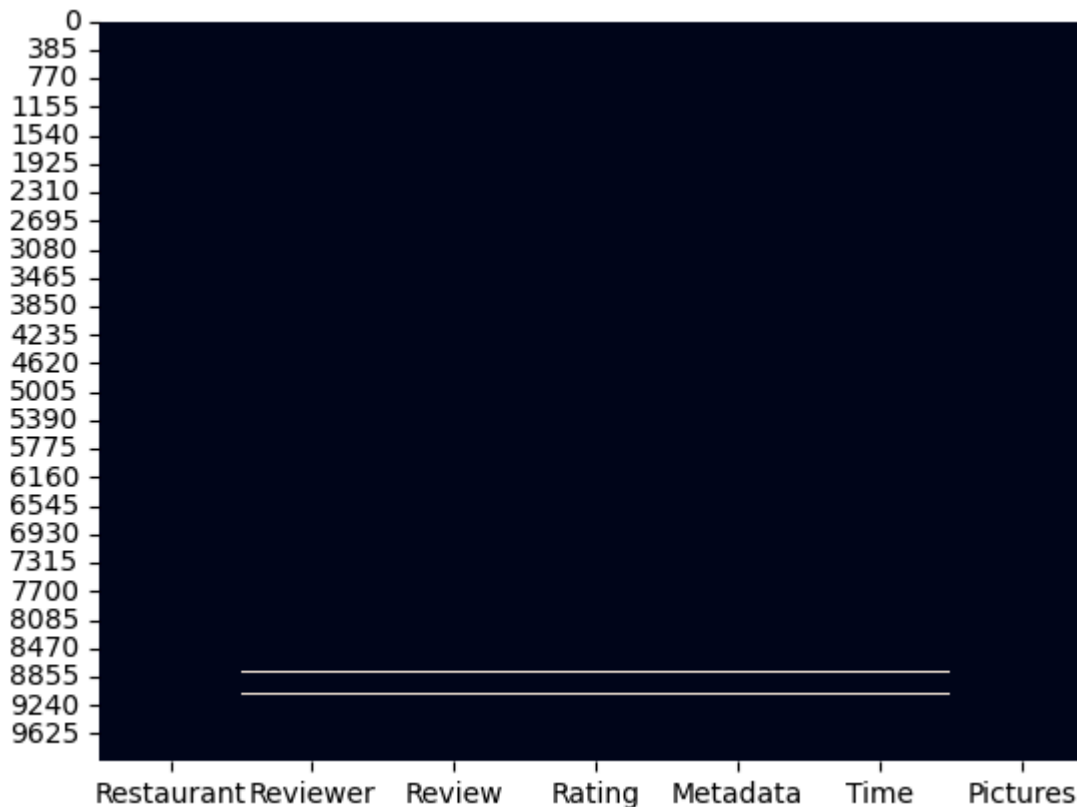
<b>Restaurant</b>	0
<b>Reviewer</b>	38
<b>Review</b>	45
<b>Rating</b>	38
<b>Metadata</b>	38
<b>Time</b>	38
<b>Pictures</b>	0

dtype: int64

```
# Visualizing the missing values for restaurant
# Checking Null Value by plotting Heatmap
sns.heatmap(hotel_df.isnull(), cbar=False);
```



```
# Visualizing the missing values for reviews
# Checking Null Value by plotting Heatmap
sns.heatmap(review_df.isnull(), cbar=False);
```



## ✓ What did you know about your dataset?

### Restaurant DataSet

- There are 105 total observation with 6 different features.
- Feature like collection and timing has null values.
- There is no duplicate values i.e., 105 unique data.
- Feature cost represent amount but has object data type because these values are separated by comma ','.
- Timing represent operational hour but as it is represented in the form of text has object data type.

### Review DataSet

- There are total 10000 observation and 7 features.
- Except picture and restaurant feature all others have null values.
- There are total of 36 duplicate values for two restaurant - American Wild Wings and Arena Eleven, where all these duplicate values generally have null values.
- Rating represent ordinal data, has object data type should be integer.
- Timing represent the time when review was posted but show object data time, it should be converted into date time.

## ✓ **2. Understanding Your Variables**

```
# Dataset Columns restaurant
print(f'Features : {hotel_df.columns.to_list()}')
```

```
Features : ['Name', 'Links', 'Cost', 'Collections', 'Cuisines', 'Timings']
```

```
# Dataset Columns review
print(f'Features : {review_df.columns.to_list()}')
```

```
Features : ['Restaurant', 'Reviewer', 'Review', 'Rating', 'Metadata', 'Time',
```

```
# Dataset Describe restaurant
hotel_df.describe().T
```

	count	unique	top	freq
Name	105	105	Beyond Flavours	1
Links	105	105	https://www.zomato.com/hyderabad/beyond-flavou...	1
Cost	105	29	500	13
Collections	51	42	Food Hygiene Rated Restaurants in Hyderabad	4
Cuisines	105	92	North Indian, Chinese	4
Timings	104	77	11 AM to 11 PM	6

```
# Dataset Describe review
review_df.describe(include='all').T
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Restaurant	10000	100	Beyond Flavours	100	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Reviewer	9962	7446	Parijat Ray	13	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Review	9955	9364	good	237	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Rating	9962	10	5	3832	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Metadata	9962	2477	1 Review	919	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Variables Description

Attributes ▶

Zomato Restaurant

- Name : Name of Restaurants
- Links : URL Links of Restaurants
- Cost : Per person estimated Cost of dining
- Collection : Tagging of Restaurants w.r.t. Zomato categories
- Cuisines : Cuisines served by Restaurants
- Timings : Restaurant Timings

### Zomato Restaurant Reviews

- Restaurant : Name of the Restaurant
- Reviewer : Name of the Reviewer
- Review : Review Text
- Rating : Rating Provided by Reviewer
- MetaData : Reviewer Metadata - No. of Reviews and followers
- Time: Date and Time of Review
- Pictures : No. of pictures posted with review

### ✓ Check Unique Values for each variable.

```
# Check Unique Values for each variable for restaurant
for i in hotel_df.columns.tolist():
    print("No. of unique values in ",i,"is",hotel_df[i].nunique(),".")
```

⇒ No. of unique values in Name is 105 .  
 No. of unique values in Links is 105 .  
 No. of unique values in Cost is 29 .  
 No. of unique values in Collections is 42 .  
 No. of unique values in Cuisines is 92 .  
 No. of unique values in Timings is 77 .

```
# Check Unique Values for each variable for reviews
for i in review_df.columns.tolist():
    print("No. of unique values in ",i,"is",review_df[i].nunique(),".")
```

⇒ No. of unique values in Restaurant is 100 .  
 No. of unique values in Reviewer is 7446 .  
 No. of unique values in Review is 9364 .  
 No. of unique values in Rating is 10 .  
 No. of unique values in Metadata is 2477 .  
 No. of unique values in Time is 9782 .  
 No. of unique values in Pictures is 36 .

### ✓ 3. *Data Wrangling*

#### ✓ Data Wrangling Code

```
#creating copy of both the data
hotel = hotel_df.copy()
review = review_df.copy()
```

#### ✓ Restaurant


```
#before changing data type for cost checking values
hotel['Cost'].unique()
```

```
⇒ array(['800', '1,300', '1,200', '1,500', '500', '300', '1,000', '350',
        '400', '1,600', '750', '550', '1,900', '450', '150', '1,400',
        '1,100', '600', '200', '900', '700', '1,700', '2,500', '850',
        '650', '1,800', '2,800', '1,750', '250'], dtype=object)
```

```
# Write your code to make your dataset analysis ready.
# changing the data type of the cost function
hotel['Cost'] = hotel['Cost'].str.replace(",","",).astype('int64')
```

```
#top 5 costlier restaurant
hotel.sort_values('Cost', ascending = False)[['Name','Cost'][:5]
```

```
⇒
```

	Name	Cost	
92	Collage - Hyatt Hyderabad Gachibowli	2800	
56	Feast - Sheraton Hyderabad Hotel	2500	
21	Jonathan's Kitchen - Holiday Inn Express & Suites	1900	
18	10 Downing Street	1900	
91	Cascade - Radisson Hyderabad Hitec City	1800	

```
#top 5 economy restaurant
hotel.sort_values('Cost', ascending = False)[['Name','Cost'][-5:]
```



	Name	Cost	
85	Momos Delight	200	
29	Hunger Maggi Point	200	
101	Sweet Basket	200	
89	Mohammedia Shawarma	150	
23	Amul	150	

```
#hotels that share same price
hotel_dict = {}
amount = hotel.Cost.values.tolist()

#adding hotel name based on the price by converting it into list
for price in amount:
    # Get all the rows that have the current price
    rows = hotel[hotel['Cost'] == price]
    hotel_dict[price] = rows["Name"].tolist()

#converting it into dataframe
same_price_hotel_df=pd.DataFrame.from_dict([hotel_dict]).transpose().reset_index(
    columns={'index':'Cost',0:'Name of Restaurants'})

#alternate methode to do the same
#same_price_hotel_df = hotel.groupby('Cost')['Name'].apply(lambda x: x.tolist()).

#getting hotel count
hotel_count = hotel.groupby('Cost')['Name'].count().reset_index().sort_values(
    'Cost', ascending = False)

#merging together
same_price_hotel_df = same_price_hotel_df.merge(hotel_count, how = 'inner',
    on = 'Cost').rename(columns = {'Name':'Total_Restaurant'})

#max hotels that share same price
same_price_hotel_df.sort_values('Total_Restaurant', ascending = False)[:5]
```



	Cost	Name of Restaurants	Total_Restaurant	
4	500	[eat.fit, KFC, Kritunga Restaurant, Karachi Ba...	13	
17	600	[Behrouz Biryani, Karachi Cafe, Hyderabad Chef...	10	
20	700	[Marsala Food Company, Green Bawarchi Restaura...	8	
2	1200	[Over The Moon Brew Company, The Glass Onion, ...	7	
8	400	[Sardarji's Chaats & More, Hotel Zara Hi-Fi, P...	6	

```
#hotels which has max price
same_price_hotel_df.sort_values('Cost', ascending = False)[:5]
```



	Cost	Name of Restaurants	Total_Restaurant	
26	2800	[Collage - Hyatt Hyderabad Gachibowli]	1	
22	2500	[Feast - Sheraton Hyderabad Hotel]	1	
12	1900	[10 Downing Street, Jonathan's Kitchen - Holid...	2	
25	1800	[Cascade - Radisson Hyderabad Hitec City]	1	
27	1750	[Zega - Sheraton Hyderabad Hotel]	1	

```
# splitting the cusines and storing in list
cuisine_value_list = hotel.Cuisines.str.split(',')
```

```
# storing all the cusines in a dict
cuisine_dict = {}
for cuisine_names in cuisine_value_list:
    for cuisine in cuisine_names:
        if (cuisine in cuisine_dict):
            cuisine_dict[cuisine]+=1
        else:
            cuisine_dict[cuisine]=1
```

```
# converting the dict to a data frame
cuisine_df=pd.DataFrame.from_dict([cuisine_dict]).transpose().reset_index().renam
    columns={'index':'Cuisine',0:'Number of Restaurants'})
```

```
#top 5 cuisine
cuisine_df.sort_values('Number of Restaurants', ascending =False)[:5]
```



	Cuisine	Number of Restaurants	
5	North Indian	61	
0	Chinese	43	
1	Continental	21	
6	Biryani	16	
18	Fast Food	15	

```
# splitting the cusines and storing in list
Collections_value_list = hotel.Collections.dropna().str.split(',')
```

```
# storing all the cusines in a dict
Collections_dict = {}
for collection in Collections_value_list:
    for col_name in collection:
        if (col_name in Collections_dict):
            Collections_dict[col_name]+=1
```

```
else:
    Collections_dict[col_name]=1
```

```
# converting the dict to a data frame
Collections_df=pd.DataFrame.from_dict([Collections_dict]).transpose().reset_index
    columns={'index':'Tags',0:'Number of Restaurants'})
```

```
#top 5 collection
Collections_df.sort_values('Number of Restaurants', ascending =False)[:5]
```



	Tags	Number of Restaurants	
2	Great Buffets	11	
0	Food Hygiene Rated Restaurants in Hyderabad	8	
5	Live Sports Screenings	7	
6	Hyderabad's Hottest	7	
1	Corporate Favorites	6	

## ✓ Reviews

```
#in order to change data type for rating checking values
review.Rating.value_counts()
```



	count
Rating	
5	3832
4	2373
1	1735
3	1193
2	684
4.5	69
3.5	47
2.5	19
1.5	9
Like	1

**dtype:** int64

```
#changing data type for each rating since had value as interger surrounded by inv
#since there is one rating as like converting it to 0 since no rating is 0 then t
```



```

review.loc[review['Rating'] == 'Like'] = 0
#changing data type for rating in review data
review['Rating'] = review['Rating'].astype('float')

#since there is one rating as like converting it to median
review.loc[review['Rating'] == 0] = review.Rating.median()

```

review


	Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures
0	Beyond Flavours	Rusha Chakraborty	The ambience was good, food was quite good . h...	5.00	1 Review , 2 Followers	5/25/2019 15:54	0
1	Beyond Flavours	Anusha Tirumalaneedi	Ambience is too good for a pleasant evening. S...	5.00	3 Reviews , 2 Followers	5/25/2019 14:20	0
2	Beyond Flavours	Ashok Shekhawat	A must try.. great food great ambience. Thnx f...	5.00	2 Reviews , 3 Followers	5/24/2019 22:54	0
3	Beyond	Swapnil	Soumen das and Arun was a great	5.00	1 Review ,	5/24/2019	0

Next steps:

[Generate code with review](#)
[View recommended plots](#)
[New interactive sheet](#)

changing date and extracting few feature for manipulation

review



	Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures
0	Beyond Flavours	Rusha Chakraborty	The ambience was good, food was quite good . h...	5.00	1 Review , 2 Followers	5/25/2019 15:54	0
1	Beyond Flavours	Anusha Tirumalaneedi	Ambience is too good for a pleasant evening. S...	5.00	3 Reviews , 2 Followers	5/25/2019 14:20	0
2	Beyond Flavours	Ashok Shekhawat	A must try.. great food great ambience. Thnx f...	5.00	2 Reviews , 3 Followers	5/24/2019 22:54	0
3	Beyond	Swapnil	Soumen das and Arun was a great	5.00	1 Review ,	5/24/2019	0

Next steps:

[Generate code with review](#)

 [View recommended plots](#)

[New interactive sheet](#)

```
review['Reviewer_Total_Review']=review['Metadata'].str.split(', ',expand=True)[0]
review['Reviewer_Followers']=review['Metadata'].str.split(', ', expand=True)[1]

review['Reviewer_Total_Review'] = pd.to_numeric(review['Reviewer_Total_Review'].s
review['Reviewer_Followers'] = pd.to_numeric(review['Reviewer_Followers'].str.spl
```

review



	Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures
0	Beyond Flavours	Rusha Chakraborty	The ambience was good, food was quite good . h...	5.00	1 Review , 2 Followers	5/25/2019 15:54	0
1	Beyond Flavours	Anusha Tirumalaneedi	Ambience is too good for a pleasant evening. S...	5.00	3 Reviews , 2 Followers	5/25/2019 14:20	0
2	Beyond Flavours	Ashok Shekhawat	A must try.. great food great ambience. Thnx f...	5.00	2 Reviews , 3 Followers	5/24/2019 22:54	0
3	Beyond Flavours	Swapnil Sarkar	Soumen das and Arun was a great guy. Only beca...	5.00	1 Review , 1 Follower	5/24/2019 22:11	0
4	Beyond Flavours	Dileep	Food is good.we ordered Kodi drumsticks and ba...	5.00	3 Reviews , 2 Followers	5/24/2019 21:37	0
...	...	...	...	...	...	...	...

Next steps:

[Generate code with review](#)[View recommended plots](#)[New interactive sheet](#)

```

review['Time']=pd.to_datetime(review['Time'],errors='coerce')
review['Review_Year'] = pd.DatetimeIndex(review['Time']).year
review['Review_Month'] = pd.DatetimeIndex(review['Time']).month
review['Review_Hour'] = pd.DatetimeIndex(review['Time']).hour

```

#Average engagement of restaurants

```

avg_hotel_rating = review.groupby('Restaurant').agg({'Rating':'mean',
            'Reviewer': 'count'}).reset_index().rename(columns = {'Reviewer': 'Total_
avg_hotel_rating

```



	Restaurant	Rating	Total_Review	
0		4.00	4.00	1
1	10 Downing Street	3.80	100	
2	13 Dhaba	3.48	100	
3	3B's - Buddies, Bar & Barbecue	4.76	100	
4	AB's - Absolute Barbecues	4.88	100	
...	...	...	...	
96	Urban Asia - Kitchen & Bar	3.65	100	
97	Yum Yum Tree - The Arabian Food Court	3.56	100	
98	Zega - Sheraton Hyderabad Hotel	4.45	100	
99	Zing's Northeast Kitchen	3.65	100	
100	eat.fit	3.20	100	

101 rows × 3 columns

Next steps:

[Generate code with avg\\_hotel\\_rating](#)[View recommended plots](#)[New interactive sh](#)

```
#unless data
review[review['Restaurant'] == 4.0]
```



	Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures	Reviewer_1
7601	4.00	4.00	4.00	4.00	4.00	NaT	4	

```
#checking hotel count as total hotel in restaurant data was 105
review.Restaurant.nunique()
```



101

```
#finding hotel without review
hotel_without_review = [name for name in hotel.Name.unique().tolist()
                        if name not in review.Restaurant.unique().tolist()]
hotel_without_review
```



```
['IndiBlaze',
 'Sweet Basket',
 'Angaara Counts 3',
 'Wich Please',
 'Republic Of Noodles - Lemon Tree Hotel']
```

```
#top 5 most engaging or rated restaurant
avg_hotel_rating.sort_values('Rating', ascending = False)[:5]
```



	Restaurant	Rating	Total_Review
4	AB's - Absolute Barbecues	4.88	100
12	B-Dubs	4.81	100
3	3B's - Buddies, Bar & Barbecue	4.76	100
68	Paradise	4.70	100
36	Flechazo	4.66	100



```
#top 5 lowest rated restaurant
avg_hotel_rating.sort_values('Rating', ascending = True)[:5]
```



	Restaurant	Rating	Total_Review
42	Hotel Zara Hi-Fi	2.40	100
11	Asian Meal Box	2.58	100
67	Pakwaan Grand	2.71	100
58	Mathura Vilas	2.82	100
15	Behrouz Biryani	2.83	100



```
#Finding the most followed critic
most_followed_reviewer = review.groupby('Reviewer').agg({'Reviewer_Total_Review':
    'Reviewer_Followers':'max', 'Rating':'mean'}).reset_index().rename(columns
    'Rating':'Average_Rating_Given').sort_values('Reviewer_Followers', asc
most_followed_reviewer[:5]
```



	Reviewer	Reviewer_Total_Review	Reviewer_Followers	Average_Rating_Given
5464	Satwinder Singh	186.00	13410.00	3.0
1702	Eat_vth_me	60.00	13320.00	5.0
5236	Samar Sardar	8.00	11329.00	3.0

```
#finding which year show maximum engagement
hotel_year = review.groupby('Review_Year')['Restaurant'].apply(lambda x: x.tolist)
hotel_year['Count']= hotel_year['Restaurant'].apply(lambda x: len(x))
hotel_year
```

	Review_Year	Restaurant	Count	
0	2016.00	[Labonel, Labonel, Labonel, Labonel, Labonel, ...	43	
1	2017.00	[KS Bakers, KS Bakers, KS Bakers, KS Bakers, K...	213	
2	2018.00	[Shah Ghouse Spl Shawarma, Shah Ghouse Spl Sha...	4903	
3	2019.00	[Beyond Flavours, Beyond Flavours, Beyond Flav...	4802	

Next  
steps:

[Generate code with hotel\\_year](#)
[View recommended plots](#)
[New interactive sheet](#)

```
#merging both data frame
hotel = hotel.rename(columns = {'Name':'Restaurant'})
merged = hotel.merge(review, on = 'Restaurant')
merged.shape
```

(9999, 17)

```
#Price point of restaurants
price_point = merged.groupby('Restaurant').agg({'Rating':'mean',
        'Cost': 'mean'}).reset_index().rename(columns = {'Cost': 'Price_Point'})
```

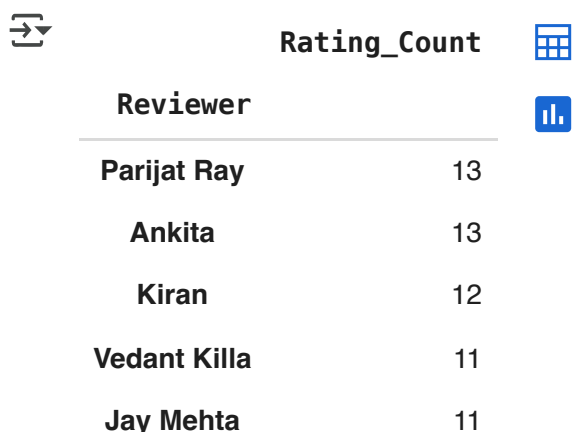
```
#price point for high rated restaurants
price_point.sort_values('Rating',ascending = False)[:5]
```

	Restaurant	Rating	Price_Point	
3	AB's - Absolute Barbecues	4.88	1500.00	
11	B-Dubs	4.81	1600.00	
2	3B's - Buddies, Bar & Barbecue	4.76	1100.00	
67	Paradise	4.70	800.00	
35	Flechazo	4.66	1300.00	

```
#price point for lowest rated restaurants
price_point.sort_values('Rating',ascending = True)[:5]
```

	Restaurant	Rating	Price_Point	
41	Hotel Zara Hi-Fi	2.40	400.00	
10	Asian Meal Box	2.58	200.00	
66	Pakwaan Grand	2.71	400.00	
57	Mathura Vilas	2.82	500.00	
14	Behrouz Biryani	2.83	600.00	

```
#rating count by reviewer
rating_count_df = pd.DataFrame(review.groupby('Reviewer').size(), columns=[
                                                                    "Rating_Count"])
rating_count_df.sort_values('Rating_Count', ascending = False)[:5]
```



Reviewer	Rating_Count
Parijat Ray	13
Ankita	13
Kiran	12
Vedant Killa	11
Jay Mehta	11

## ✓ What all manipulations have you done and insights you found?

Firstly, I started with changing data types for cost and rating. In rating there was only one rating which was string or has value of like so I change it into median of the rating. This was done to make data consistent.

Restaurant data : In this dataset I first figured out 5 costlier restaurant in which Collage - Hyatt Hyderabad Gachibowli has maximum price of 2800 and then found the lowest which is Amul with price of 150. Then I found how many hotel share same price i.e., 13 hotel share 500 price. North indian cuisine with great buffet tags is mostly used in hotels.

Review data : In this dataset I found famous or restaurant that show maximum engagement. Followed by that I found most followed critic which was Satwinder Singh who posted total of 186 reviews and had followers of 13410 who gives and average of 3.67 rating for each order he makes. Lastly I also found in year 2018 4903 hotels got reviews.

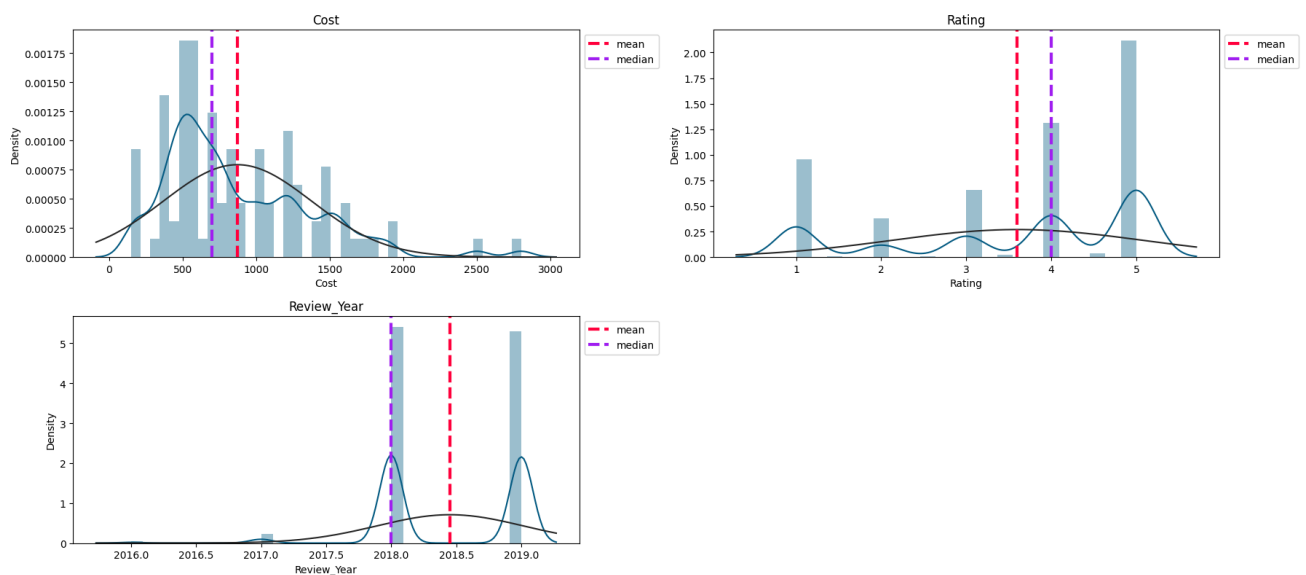
Then I merged the two dataset together to figure out the price point for the restaurant, top rated restaurant AB's - Absolute Barbecues has a price point of 1500 and the low rated Hotel Zara Hi-Fi has price point of 400.

In order to exactly understand why even with price point of 1500 these hotel has maximum number of rating and sentiment of those rating need to extract words from the text and do further analysis of the review and then followed by forming clusters so that one can get recommendation about top quality restaurants.

## ✓ **4. Data Vizualization, Storytelling & Experimenting with charts :** **Understand the relationships between variables**

## ✓ Chart - 1 Distplot for Distribution

```
# Chart - 1 visualization code
plt.figure(figsize = (18,8));
for i,col in enumerate(['Cost','Rating','Review_Year']) :
    # plt.figure(figsize = (8,5));
    plt.subplot(2,2,i+1);
    sns.distplot(merged[col], color = '#055E85', fit = norm);
    feature = merged[col]
    plt.axvline(feature.mean(), color='#ff033e', linestyle='dashed', linewidth=3,
    plt.axvline(feature.median(), color='#A020F0', linestyle='dashed', linewidth=3,
    plt.legend(bbox_to_anchor = (1.0, 1), loc = 'upper left')
    plt.title(f'{col.title()}');
    plt.tight_layout();
```



## ✓ 1. Why did you pick the specific chart?

Distplot is helpful in understanding the distribution of the feature.

## ✓ 2. What is/are the insight(s) found from the chart?



- All three are show skewness.
- Maximum restaurant show price range for 500.
- In 2018 number of reviews are more.

### ✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

Price always place important role in any business alongwith rating which show how much engagement are made for the product.

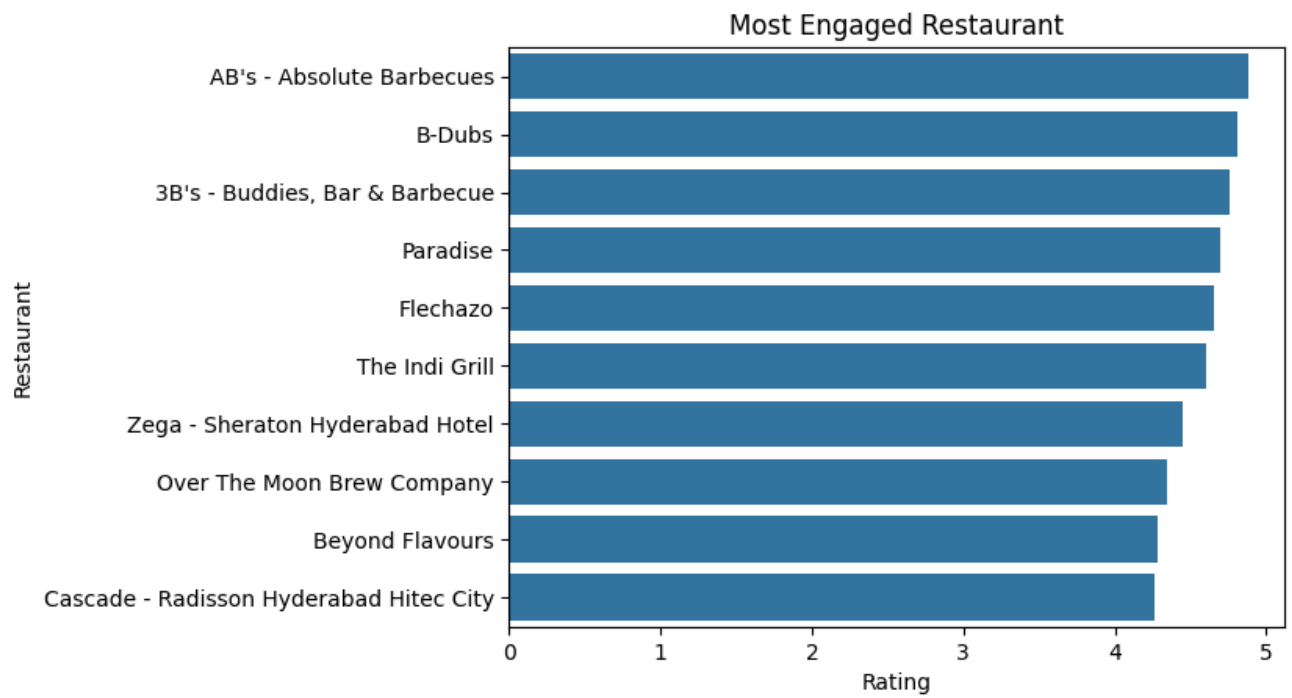
But in this chart it is unable to figure any impact on business when plotted all alone.

### ✓ Price Point and Maximum Engagement

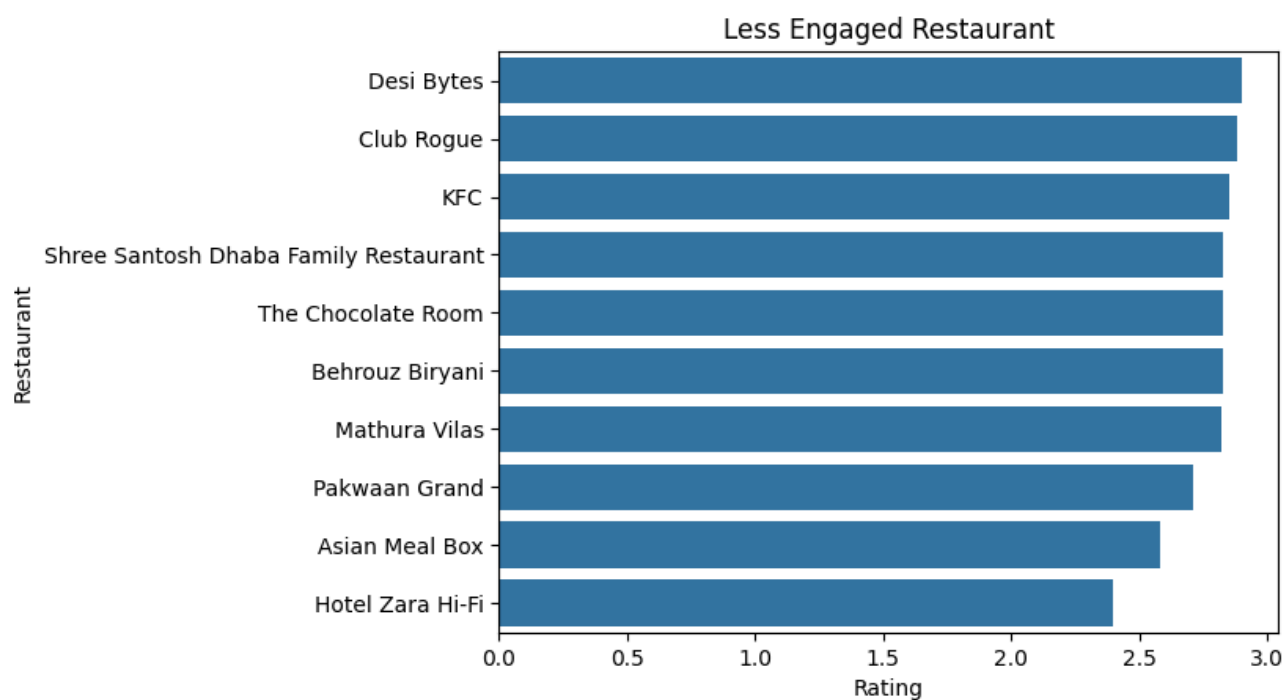
### ✓ Chart - 2 Maximum Engagement and Lowest Engagement

```
#geting the top 10 hotel that show maximum engagement
most_engaged_hotel = price_point.sort_values('Rating', ascending = False)

# Chart - 2 visualization code for most liked
sns.barplot(data = most_engaged_hotel[:10], x = 'Rating', y = 'Restaurant')
plt.title('Most Engaged Restaurant')
plt.show()
```



```
#chart for less liked hotels  
sns.barplot(data = most_engaged_hotel[-10:], x = 'Rating', y = 'Restaurant')  
plt.title('Less Engaged Restaurant')  
plt.show()
```



✓ 1. Why did you pick the specific chart?

I picked barplot for the above graph because it show frequency level for different category.

✓ 2. What is/are the insight(s) found from the chart?

AB's - Absolute Barbecues, show maximum engagement and retention as it has maximum number of rating on average and Hotel Zara Hi-Fi show lowest engagement as has lowest average rating.

✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

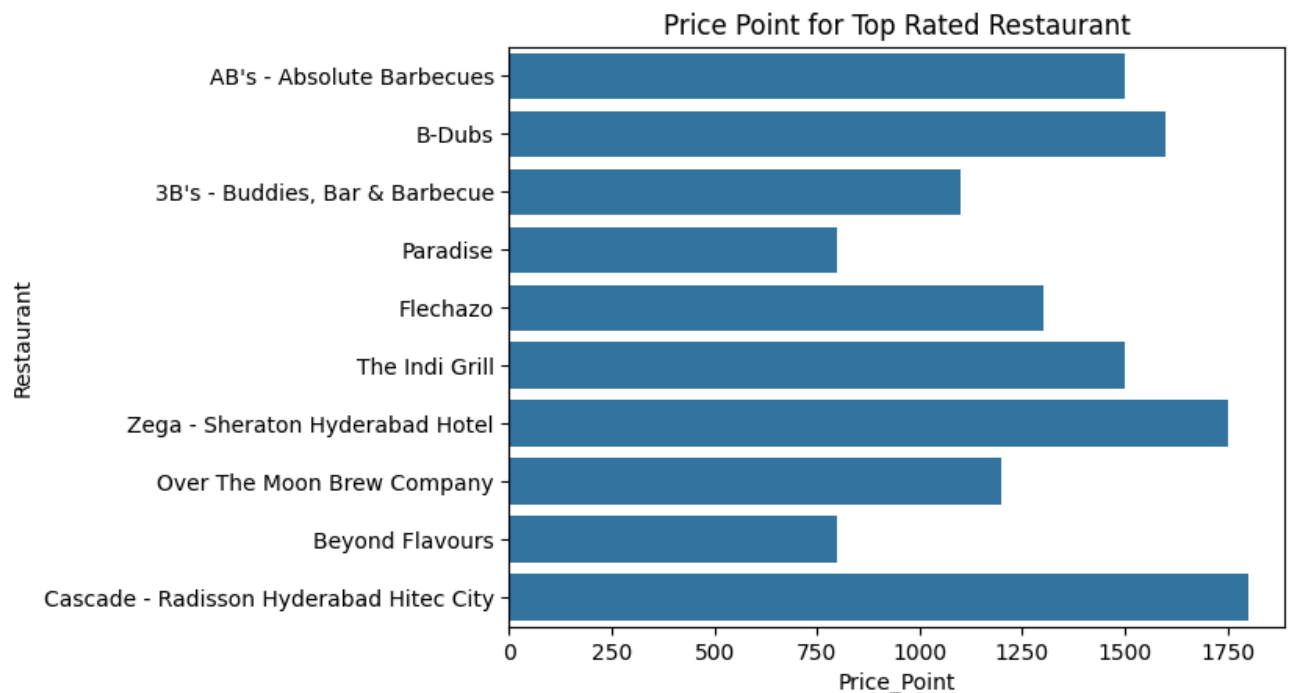
Engagement and retention for any business is very much important as profit and scalability for any business depend upon retention of customers. Maximum retention means people prefer to use the same brand over others.

Some restaurant show less rating which can show negative growth if not monitored why they receive less order for example KFC is listed in low rated it is sure they have different outlet and

their own outsourcing and listed here because of the popularity of the app and to increase their sale and demand but are not giving 100% dedication to the platform to generate revenue.

### ✓ Chart - 3 Price Point for High Rated and Low Rated Hotels

```
# Chart - 3 visualization code for price point of high rated restaurant
sns.barplot(data = most_engaged_hotel[:10], x = 'Price_Point', y = 'Restaurant')
plt.title('Price Point for Top Rated Restaurant')
plt.show()
```



```
#visualization code for price point of low rated restaurant
sns.barplot(data = most_engaged_hotel[-10:], x = 'Price_Point',
            y = 'Restaurant', palette = 'hsv')
plt.title('Price Point for Low Rated Restaurant', size = 15)
# Setting the background color of the plot
# using set_facecolor() method
ax = plt.axes()
ax.set_facecolor("black")
plt.show()
```



✓ 1. Why did you pick the specific chart?

Here I choose barplot because bar plot is a good choice for plotting hotel name and price point as it is a simple and effective way to display the comparison of different categories (hotel names) and their corresponding values (price points) on the same chart. Also, it allow to have a sense of the price range of each hotel and how they compare to each other.

✓ 2. What is/are the insight(s) found from the chart?

Price point for high rated hotel AB's= Absolute Barbecues is 1500 and price point for low rated restaurant Hotel Zara Hi-Fi is 400.

✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

Since it is customer centered business i.e., direct to consumer it is important to understand price point which makes this business more affordable for evryone, therefore it is important for business to crack the price point.

Here most liked restaurant has a price point of 1500 which is even though a little high than average but as this business is all about food quality and taste it show maximum engagement which means it serve best quality of food, however deep dive on analysing review text can exactly give why this price point is preferred most.

Some restaurant with lowest rating even with low price point is not making engagement, this may create a negative impact on business.

However it can not be finalized that this hotel should unlisted as there may be chance of different cuisine they both serve and it also depend upon the locality they both serve, therefore based on that small promotional offers can also be given for low rated restaurant to increase sales.

## ✓ Commoditized Cuisine

## ✓ Chart - 4 Proportion of Cuisine Sold by Most Restaurant

```
#list of all cuisine
cuisine_list = cuisine_df.sort_values('Number of Restaurants', ascending = False)

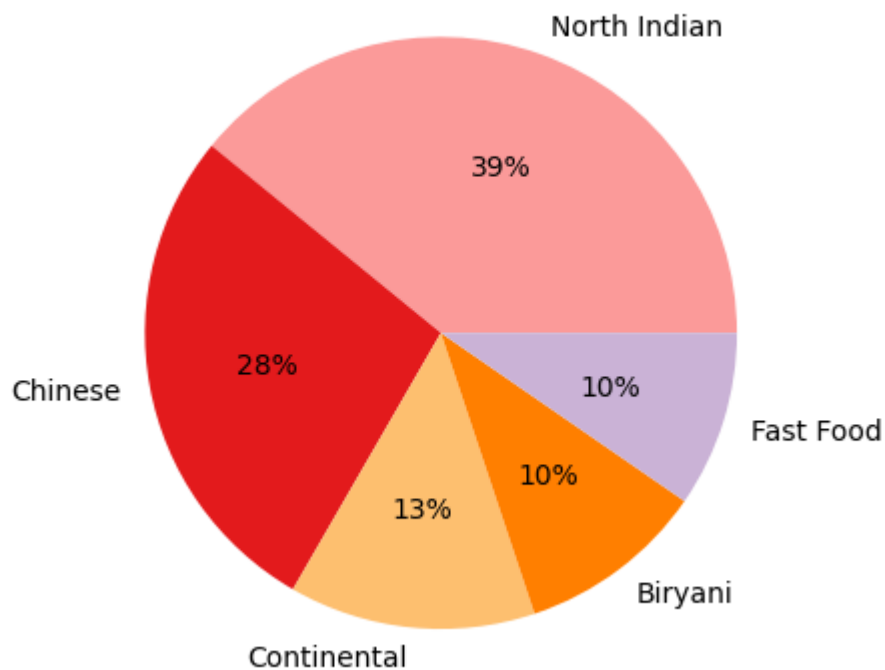
# Chart - 4 visualization code pie chart for top 5 mpst selling cuisine
data = cuisine_df.sort_values('Number of Restaurants', ascending = False)[
    'Number of Restaurants'].tolist()[:5]
labels = cuisine_list

#define Seaborn color palette to use
colors = sns.color_palette('Paired')[4:9]

#create pie chart
plt.pie(data, labels = labels, colors = colors, autopct='%0f%%')
plt.title('Top 5 Most Selling Cuisine', size =22, color= 'blue')
plt.show()
```



## Top 5 Most Selling Cuisine



```
#wordcloud for Cuisine
# storind all cuisine in form of text
plt.figure(figsize=(15,8))
text = " ".join(name for name in cuisine_df.Cuisine )

# Creating word_cloud with text as argument in .generate() method

word_cloud = WordCloud(width = 2000, height = 2000,collocations = False,
                        colormap='rainbow',background_color = 'black').generate(text)

# Display the generated Word Cloud

plt.imshow(word_cloud, interpolation='bilinear');

plt.axis("off");
```



```
# #creating variable to store restaurant and cuisine from hotel dataset
# cproduct = hotel[['Restaurant','Cost',      'Cuisines']].copy()
# #splitting cuisines
# cproduct['Cuisines'] = cproduct['Cuisines'].str.split(',')
# #exploding the cuisine list from above to separate row
# cproduct = cproduct.explode('Cuisines')
# #removing trailing spaces
# cproduct['Cuisines'] = cproduct['Cuisines'].apply(lambda x: x.strip())
# #grouping cuisines and then making list of restaurants
# cprod = cproduct.groupby('Cuisines')['Restaurant'].apply(lambda x: x.tolist()).
# # cproduct['Cuisines'].unique()
# cprod['Restaurant_Count'] = cprod['Restaurant'].apply(lambda x: len(x))
# cprod[cprod['Restaurant_Count']==1].sort_values('Restaurant_Count', ascending =
```



### ✓ 1. Why did you pick the specific chart?

Here I choose to use pie chart because it show proportion of each quantity and used wordcloud because it show all text and highlight the most frequent words.

### ✓ 2. What is/are the insight(s) found from the chart?

Based on the above chart it is clear that most of the hotel sold North Indian food followed by chinese.

### ✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

Identifying the Commoditized Cuisine plays an important role as it helps in identifying the challenge or Competitive Advantage i.e., Knowing which cuisines are commoditized allows a restaurant or food business to differentiate themselves from their competitors by offering unique and non-commoditized options.

If a cuisine is commoditized, the prices for ingredients and labor for that cuisine may be higher than for non-commoditized cuisines. Identifying these commoditized cuisines can help a business to control costs by focusing on non-commoditized options or finding ways to lower the cost of commoditized items.

Identifying commoditized cuisines can also provide insight into consumer preferences, which can be used to make informed decisions about menu offerings, pricing, and promotions.

Plotting a pie chart of cuisine types can help to identify the most popular cuisine types among its customers. This information can be used to make strategic decisions about which cuisines to focus on promoting and expanding. For example, as the significant portion of customers are searching for north indian restaurants, Zomato could focus on adding more north indian restaurants to its platform and promoting them to customers.

Similarly, a word cloud of cuisine can help Zomato identify the most frequently mentioned cuisine types in customer reviews. This can provide insight into which cuisines are most popular and well-regarded among customers, and which cuisines may need improvement.

However, these types of charts do not provide all the information about the business, and can not be the only decision making factor. For example, a pie chart showing that a certain cuisine is popular does not tell us about the profitability of that cuisine or the competition in that category. The same goes for word cloud, it only shows us the frequency of the cuisine mentioned, it can not tell us if the mentions are positive or negative.

Additionally, these charts do not provide information about the other factors that can impact the business such as market trends, consumer preferences, and economic conditions. Therefore, it's important for Zomato to consider other data and information when making strategic decisions.

### ✓ Chart - 5 Most used Tags

```
#list of all collection
collection_list = Collections_df.sort_values('Number of Restaurants',
                                             ascending = False)['Tags'].tolist()[:5]

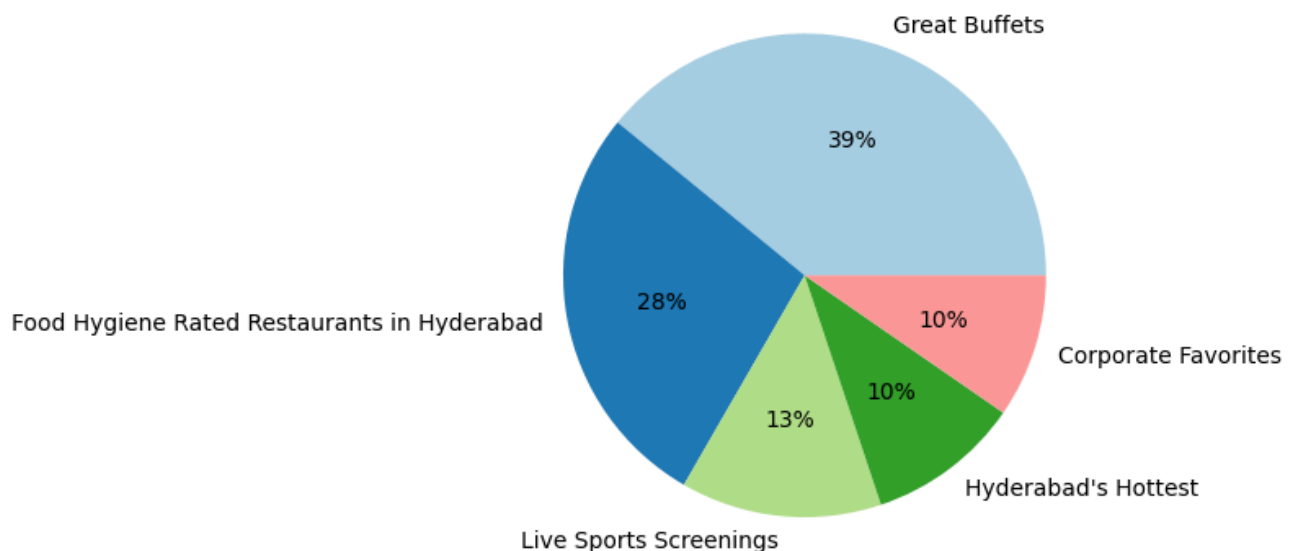
# Chart - 5 visualization code pie chart for top 5 mpst selling cuisine
data = cuisine_df.sort_values('Number of Restaurants', ascending = False)[
    'Number of Restaurants'].tolist()[:5]
labels = collection_list

#define Seaborn color palette to use
colors = sns.color_palette('Paired')[:5]

#create pie chart
plt.pie(data, labels = labels, colors = colors, autopct='%0f%%')
plt.title('Top 5 Most Selling Cuisine', size =22, color= 'red')
plt.show()
```



## Top 5 Most Selling Cuisine



```
#wordcloud for Cuisine
# storind all cuisine in form of text
plt.figure(figsize=(15,8))
text = " ".join(name for name in Collections_df.Tags )
```

```
# Creating word_cloud with text as argument in .generate() method

word_cloud = WordCloud(width = 1400, height = 1400,collocations = False,
                        colormap='rainbow', background_color = 'black').generate(te

# Display the generated Word Cloud

plt.imshow(word_cloud, interpolation='bilinear');

plt.axis("off");
```



#### ✓ 1. Why did you pick the specific chart?

The pie chart provides a clear and simple way to see the proportion of different food attributes, making it easy to identify the most popular attributes and compare them to one another. It also

allows for a quick comparison of the popularity of different attributes, and can be useful in identifying patterns or trends in the data.

On the other hand, a word cloud displays the most frequently mentioned attributes in a way that is visually striking and easy to understand. It is useful for identifying the most frequently mentioned attributes and can be used to quickly identify patterns and trends in customer reviews.

Both charts, when used together, can provide a comprehensive understanding of customer reviews and can be used to identify customer preferences, which can help Zomato to make strategic decisions to improve their business.

✓ 2. What is/are the insight(s) found from the chart?

Great Buffets is the most frequently used tags and other tags like great, best, north, Hyderabad is also used in large quantity.

✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

Plotting a pie chart of tags used to describe food can help a restaurant review and food delivery platform Zomato to identify the most popular adjectives used to describe the food. This information can be used to make strategic decisions about which food attributes to focus on promoting and expanding. For example, if a significant portion of customers are describing the food as "delicious" or "fresh", Zomato could focus on adding more restaurants that are known for their delicious and fresh food and promoting them to customers.

Similarly, a word cloud of tags used to describe food can help Zomato identify the most frequently mentioned food attributes in customer reviews. This can provide insight into which attributes are most popular and well-regarded among customers, and which attributes may need improvement.

However, it's important to note that these types of charts do not provide all the information about the business, and can not be the only decision making factor. For example, a pie chart showing that a certain adjective is popular does not tell us about the profitability of that adjective or the competition in that category. The same goes for word cloud, it only shows us the frequency of the adjective mentioned, it can not tell us if the mentions are positive or negative.

Additionally, these charts do not provide information about the other factors that can impact the business such as market trends, consumer preferences, and economic conditions. Therefore, it's important for Zomato to consider other data and information when making strategic

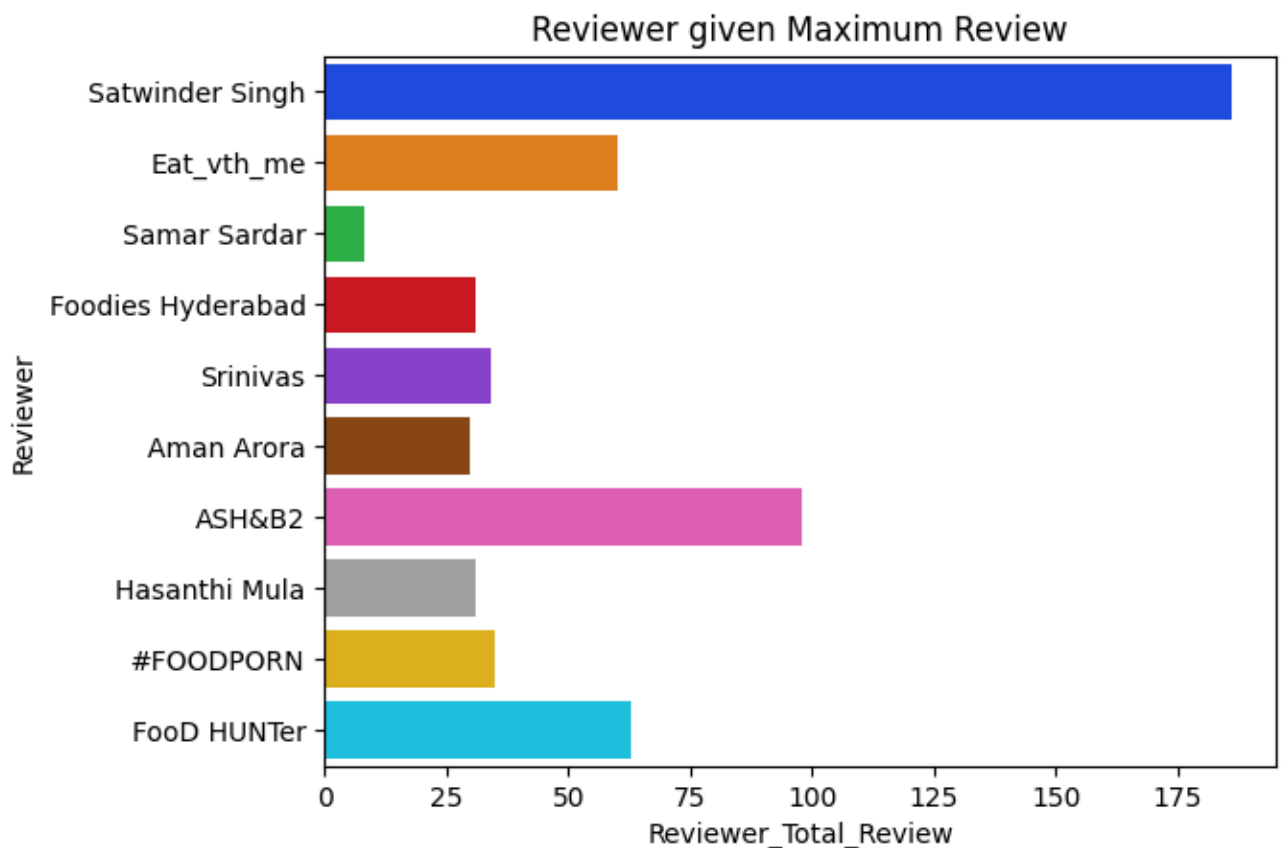
decisions. Also, it's important to note that the data used for creating these charts should be cleaned and validated, as the results may be biased if the data is not accurate or complete.

## ✓ Most Popular Critics

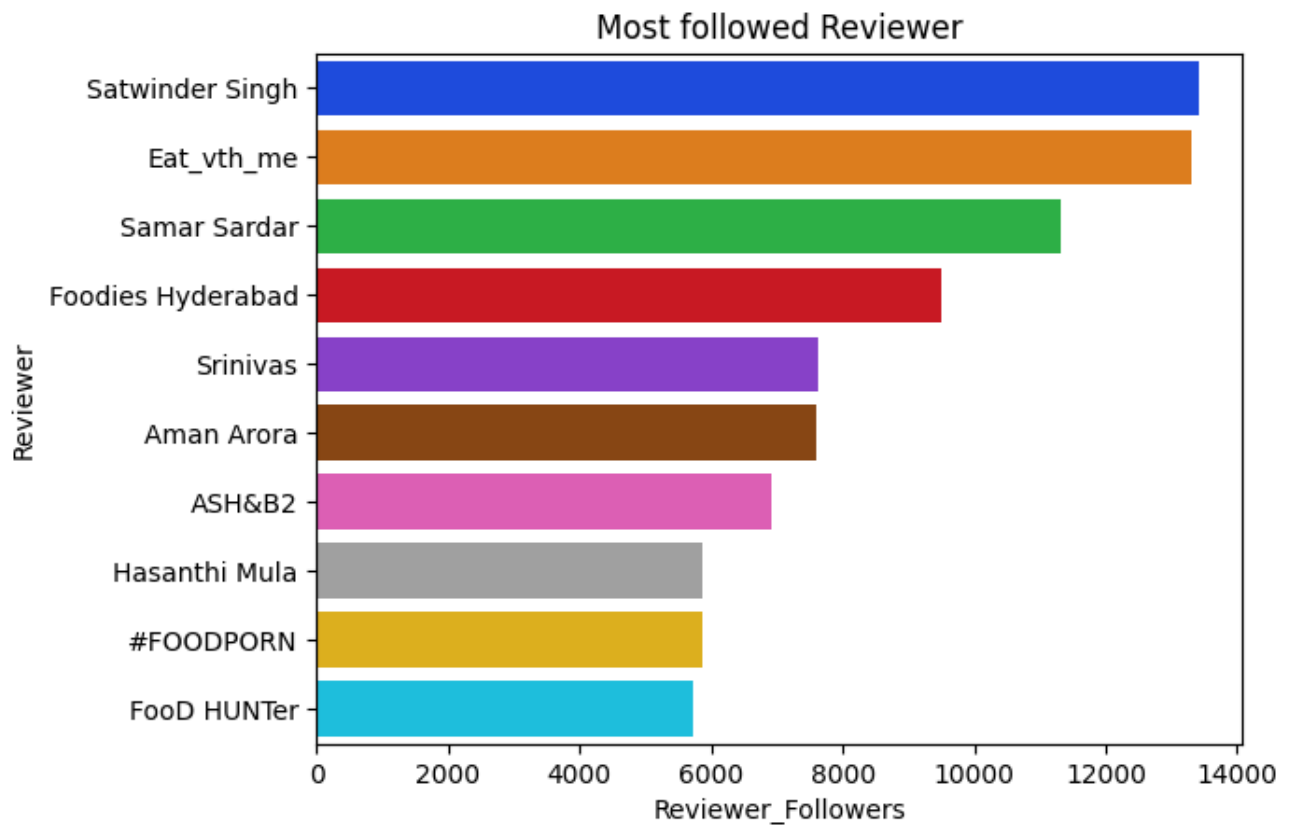
Chart - 6 Learn about Reviewers

## ✓ Chart - 6 visualization code for most review

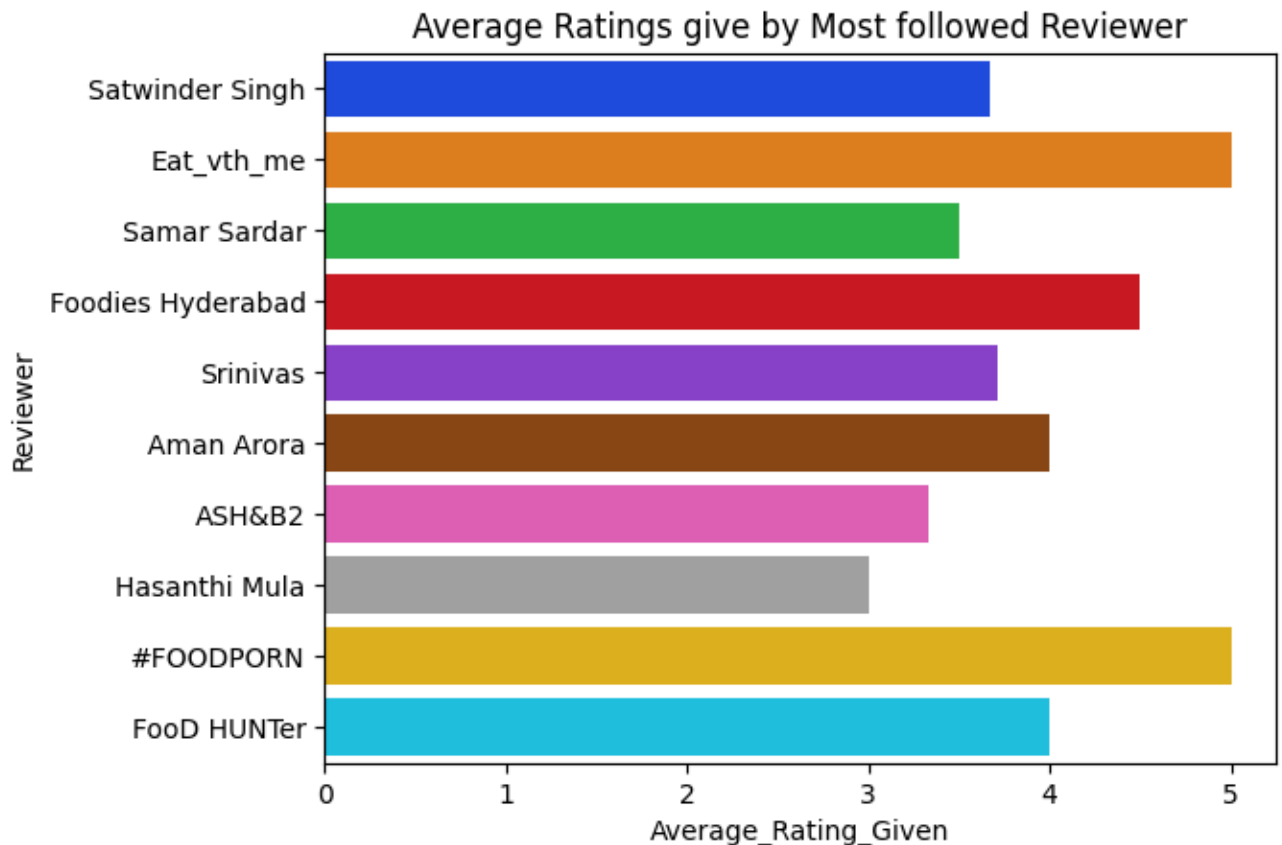
```
# Chart - 6 visualization code for most review
sns.barplot(data = most_followed_reviewer[:10], x = 'Reviewer_Total_Review',
            y = 'Reviewer', palette='bright')
plt.title('Reviewer given Maximum Review')
plt.show()
```



```
# visualization code for most review follower
sns.barplot(data = most_followed_reviewer[:10], x = 'Reviewer_Followers',
            y = 'Reviewer', palette='bright')
plt.title('Most followed Reviewer')
plt.show()
```



```
# visualization code for average rating given by most followed reviewer
sns.barplot(data = most_followed_reviewer[:10], x = 'Average_Rating_Given',
            y = 'Reviewer', palette='bright')
plt.title('Average Ratings give by Most followed Reviewer')
plt.show()
```



✓ 1. Why did you pick the specific chart?

Barplot helps in understanding the frequency of rating, follower and total reviews with respect to reviewer. Plotting total review, average reviewer rating, and total follower allows to see the correlation between these variables and how they relate to one another for each reviewer. It can also give insight on how reviewers with more followers tend to get more reviews, how their ratings tend to be, etc.

✓ 2. What is/are the insight(s) found from the chart?

Satwinder singh is the most popular critic who has maximum number of follower and on an average he give 3.5 rating.

✓ 3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

This information can be used to make strategic decisions about which reviewers to focus on promoting and expanding. For example, if a certain reviewer has a high average rating and a large number of followers, Zomato could focus on promoting their reviews to customers.

It's important to note that this chart does not provide all the information about the business, and can not be the only decision making factor. However it can help on promotions food based on reviews.

## ✓ Most Expensive Restaurant

## ✓ Chart - 7 Hotel with Highest Price and Lowest Price

```
#extracting name and price
```

```
price_of_hotel = hotel.sort_values('Cost', ascending = False)[['Restaurant', 'Cost
```

```
# Chart - 7 visualization code for howtel with maximum price
```

```
sns.barplot(data = price of hotel[:10], x = "Cost", y='Restaurant', palette = 'br
```