

Jupyter Notebook Execution Report

Name: Jatin Narayan Bhoir
Project Title: Swiggy Data Analysis
Date: December 29, 2025

Cell 1: ■ Code

```
import pandas as pd\n\nimport re
```

Cell 2: ■ Code

```
df =pd.read_csv("swiggy_scrap_uncleaned.csv", encoding="utf-8")\n\ndf.head()
```

Output:

```
          hotel_name ... offer\n0      McDonald's ... 30% OFF UPTO ■75\n1            KFC ... 40% OFF UPTO ■80\n2  Domino's Pizza ... ■150 OFF ABOVE ■299\n3  Charcoal Eats - Biryani & Beyond ... 50% OFF UPTO ■100\n4        Sandwizzaa ... 60% OFF UPTO ■120\n\n[ 5 rows x 5 columns]
```

Cell 3: ■ Code

```
df.columns = [\n    "hotel_name",\n    "rating_food",\n    "food_type",\n    "location",\n    "offer"\n]
```

```
df.head()
```

Output:

```
          hotel_name ... offer
0      McDonald's ... 30% OFF UPTO ■75
1           KFC ... 40% OFF UPTO ■80
2  Domino's Pizza ... ■150 OFF ABOVE ■299
3 Charcoal Eats - Biryani & Beyond ... 50% OFF UPTO ■100
4      Sandwizzaa ... 60% OFF UPTO ■120

[5 rows x 5 columns]
```

Cell 4: ■ Code

```
df["rating"] = df["rating_food"].str.extract(r'(\d+\.\d+)').astype(float)

df.head()

#df.drop(columns=["rating_food"], inplace=True)
```

Cell 5: ■ Code

```
def extract_delivery_time(text):

    text = str(text)

    mins_part = text.split("•")[-1] # take part after bullet

    numbers = re.findall(r'\d+', mins_part)

    if len(numbers) == 1:
        return int(numbers[0])

    elif len(numbers) == 2:
        return (int(numbers[0]) + int(numbers[1])) / 2

    else:
        return None

df["delivery_time_mins"] = df["rating_food"].apply(extract_delivery_time)

df.head()
```

Output:

```
          hotel_name rating_food ... rating delivery_time_mins
0      McDonald's     4.5 • 27 mins ...     4.5                  27.0
```

```

1                 KFC  4.2 • 30 mins ... 4.2          30.0
2             Domino's Pizza 4.3 • 25 mins ... 4.3          25.0
3 Charcoal Eats - Biryani & Beyond 4.3 • 24 mins ... 4.3          24.0
4                 Sandwizzaa 4.6 • 22 mins ... 4.6          22.0

[5 rows x 7 columns]

```

Cell 6: ■ Code

```

#df["primary_food"] = df["food_type"].apply(lambda x: x[1])
df["primary_food"] = df["food_type"].apply(lambda x:
x.split(",")[0].strip().lower())
df.head()

```

Output:

```

            hotel_name ... primary_food
0           McDonald's ...     burgers
1                  KFC ...     burgers
2           Domino's Pizza ...     pizzas
3 Charcoal Eats - Biryani & Beyond ...     biryani
4                 Sandwizzaa ...     snacks

[5 rows x 8 columns]

```

Cell 7: ■ Code

```
print(df)
```

Output:

```

            hotel_name ... primary_food
0           McDonald's ...     burgers
1                  KFC ...     burgers
2           Domino's Pizza ...     pizzas
3 Charcoal Eats - Biryani & Beyond ...     biryani
4                 Sandwizzaa ...     snacks
...
1744           Allspice ... north indian
1745 Poddar Cuisine ...     chinese

```

```
1746                      JUG's Kitchen ...      indian
1747 Choco Magic Patisserie & Confectioners Studio ... desserts
1748                      Yogita Restaurant ...      indian
[1749 rows x 8 columns]
```

Cell 8: ■ Code

```
location_map = {
    "Malad Kan East": "Malad East",
    "Borivali": "Borivali West",
    "Thakur village Kandivali east": "Kandivali",
    "Thakur village": "Kandivali"
}

df["location"] = df["location"].replace(location_map)
df.head()
```

Output:

```
          hotel_name ... primary_food
0           McDonald's ...     burgers
1                 KFC ...     burgers
2           Domino's Pizza ...     pizzas
3 Charcoal Eats - Biryani & Beyond ...     biryani
4            Sandwizzaa ...     snacks
[5 rows x 8 columns]
```

Cell 9: ■ Code

```
df["offer"] = df["offer"].fillna("No Offer")
df.head()
```

Output:

```
          hotel_name ... primary_food
0           McDonald's ...     burgers
1                 KFC ...     burgers
2           Domino's Pizza ...     pizzas
3 Charcoal Eats - Biryani & Beyond ...     biryani
```

```
4          Sandwizzaa ...      snacks  
[ 5 rows x 8 columns]
```

Cell 10: ■ Code

```
df["discount_percent"] = df["offer"].str.extract(r'(\d+)%').astype(float)  
df["discount_percent"] = df["discount_percent"].fillna(0)  
df.head()
```

Output:

```
          hotel_name ... discount_percent  
0          McDonald's ...      30.0  
1                  KFC ...      40.0  
2          Domino's Pizza ...      0.0  
3 Charcoal Eats - Biryani & Beyond ...      50.0  
4          Sandwizzaa ...      60.0  
[ 5 rows x 9 columns]
```

Cell 11: ■ Code

```
df["offer_amount_rs"] = df["offer"].str.extract(r'\s*(\d+)').astype(float)  
df["offer_amount_rs"] = df["offer_amount_rs"].fillna(0)  
df.head()
```

Output:

```
          hotel_name ... offer_amount_rs  
0          McDonald's ...      75.0  
1                  KFC ...      80.0  
2          Domino's Pizza ...    150.0  
3 Charcoal Eats - Biryani & Beyond ...    100.0  
4          Sandwizzaa ...      120.0  
[ 5 rows x 10 columns]
```

Cell 12: ■ Code

```
def get_offer_type(text):
```

```

if "UPTO" in text:
    return "UPTO"
elif "ABOVE" in text:
    return "ABOVE"
elif "ITEM" in text:
    return "ITEM_BASED"
else:
    return "NONE"

df[ "offer_type" ] = df[ "offer" ].apply(get_offer_type)
df.head()

```

Output:

	hotel_name	rating_food	...	offer_amount_rs	offer_type
0	McDonald's	4.5 • 27 mins	...	75.0	UPTO
1	KFC	4.2 • 30 mins	...	80.0	UPTO
2	Domino's Pizza	4.3 • 25 mins	...	150.0	ABOVE
3	Charcoal Eats - Biryani & Beyond	4.3 • 24 mins	...	100.0	UPTO
4	Sandwizzaa	4.6 • 22 mins	...	120.0	UPTO

[5 rows x 11 columns]

Cell 13: ■ Code

```

df[ "has_offer" ] = df[ "offer" ].apply(lambda x: "No" if x == "No Offer" else "Yes")
df.head()

```

Output:

	hotel_name	rating_food	...	offer_type	has_offer
0	McDonald's	4.5 • 27 mins	...	UPTO	Yes
1	KFC	4.2 • 30 mins	...	UPTO	Yes
2	Domino's Pizza	4.3 • 25 mins	...	ABOVE	Yes
3	Charcoal Eats - Biryani & Beyond	4.3 • 24 mins	...	UPTO	Yes
4	Sandwizzaa	4.6 • 22 mins	...	UPTO	Yes

[5 rows x 12 columns]

Cell 14: ■ Code

```

df = df.drop_duplicates(subset=[ "hotel_name", "location"], keep="first")

df.info()

df.head()

```

Output:

```

<class 'pandas.core.frame.DataFrame'>

Index: 1730 entries, 0 to 1748

Data columns (total 12 columns):

 #   Column           Non-Null Count Dtype  
--- 
 0   hotel_name      1730 non-null   object  
 1   rating_food     1730 non-null   object  
 2   food_type       1730 non-null   object  
 3   location        1730 non-null   object  
 4   offer            1730 non-null   object  
 5   rating           1267 non-null   float64 
 6   delivery_time_mins 1730 non-null   float64 
 7   primary_food    1730 non-null   object  
 8   discount_percent 1730 non-null   float64 
 9   offer_amount_rs 1730 non-null   float64 
 10  offer_type      1730 non-null   object  
 11  has_offer       1730 non-null   object  

dtypes: float64(4), object(8)

memory usage: 175.7+ KB

          hotel_name  rating_food ... offer_type has_offer
0          McDonald's    4.5 • 27 mins ...      UPTO      Yes
1                  KFC    4.2 • 30 mins ...      UPTO      Yes
2          Domino's Pizza    4.3 • 25 mins ...     ABOVE      Yes
3 Charcoal Eats - Biryani & Beyond    4.3 • 24 mins ...      UPTO      Yes
4          Sandwizzaa    4.6 • 22 mins ...      UPTO      Yes

[5 rows x 12 columns]

```

Cell 15: ■ Code

```

df.to_csv( "SWIGGY_cleaned_final.csv" , index=False)

```

Cell 16: ■ Code

```
df['location'].value_counts().head(10)
```

Output:

```
location
Borivali West      435
Malad East        242
Malad Kan West    191
Malad West         162
Kandivali East     133
Kandivali West      86
Dahisar             76
Jog Gor East       74
Goregaon            39
Goregaon East       31
Name: count, dtype: int64
```

Cell 17: ■ Code

```
df['food_type'].value_counts().head(15)
```

Output:

```
food_type
Chinese            79
Indian              74
Bakery              41
Indian, Chinese     39
Bakery, Desserts    32
Desserts            31
Beverages           25
North Indian        18
Fast Food            17
Ice Cream, Desserts  15
Pizzas              15
```

```
Chinese, Indian      15  
South Indian        14  
North Indian, Chinese 14  
Ice Cream           13  
  
Name: count, dtype: int64
```

Cell 18: ■ Code

```
df.groupby('location')['rating'].mean().sort_values(ascending=False)
```

Output:

```
location  
Gor East            4.9  
Avenue Hotel         4.7  
Goregaon West\t     4.7  
Malad kan east       4.7  
Kandivali borivali East 4.7  
...  
Opp. Chincholi Bunder Road  NaN  
Pizza Story          NaN  
Raja Dalvi Marg       NaN  
Thakur Village, Kandivali (East)  NaN  
kandivali             NaN  
  
Name: rating, Length: 123, dtype: float64
```

Cell 19: ■ Code