Internship Summary

Intern Name : Arindam Sharma

Internship Role : Data Science Intern

Organization : Cognifyz Technologies

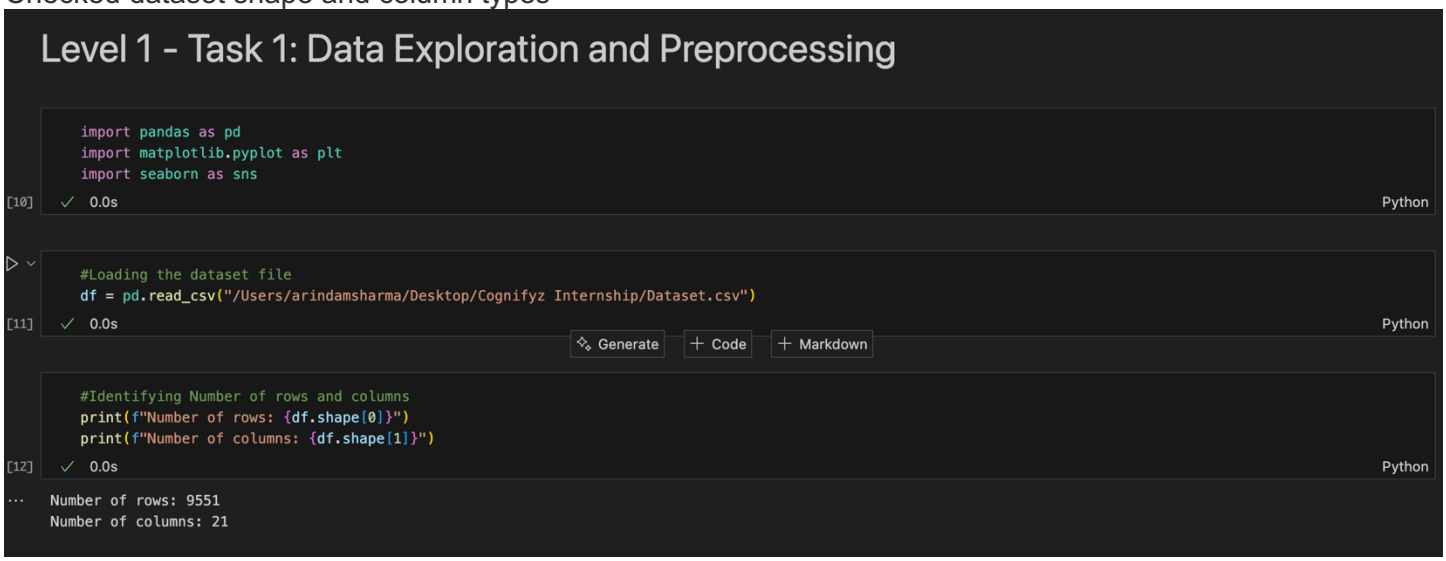
Completion Date : June 20, 2025

This report outlines the successful completion of tasks across Level 1 and Level 2 of the Data Science Internship.

Level 1: Tasks Overview

Task 1: Data Exploration and Preprocessing

- Checked dataset shape and column types



- Handled missing values in 'Cuisines'

```
# Checking for missing values
missing = df.isnull().sum()
print("Missing values:\n", missing[missing > 0])
```

[13] ✓ 0.0s Python

```
Missing values:
 Cuisines    9
dtype: int64
```

```
#Handling the mising values
df = df.dropna(subset=['Cuisines'])
print("Data types:\n", df.dtypes)
```

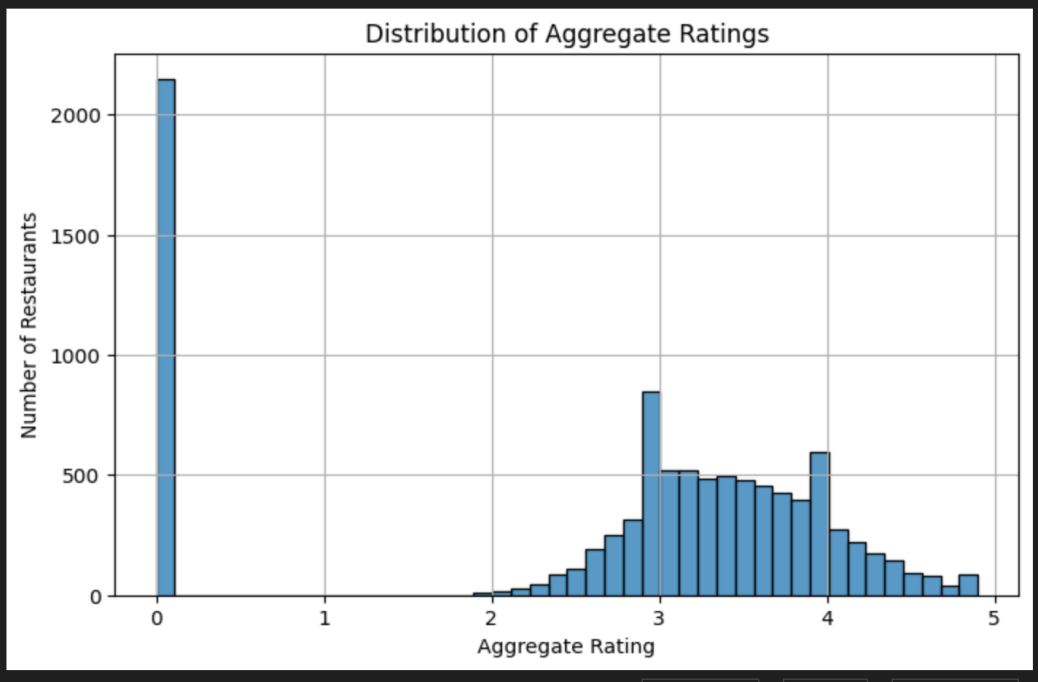
[14] ✓ 0.0s Python

```
Data types:
 Restaurant ID            int64
Restaurant Name          object
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 object
Average Cost for two      int64
Currency                 object
Has Table booking        object
Has Online delivery      object
Is delivering now        object
Switch to order menu     object
Price range               int64
Aggregate rating        float64
Rating color             object
Rating text              object
Votes                     int64
dtype: object
```

- Analyzed distribution of 'Aggregate rating'

```
#Distribution of Aggregate Rating
plt.figure(figsize=(8, 5))
sns.histplot(df['Aggregate rating'])
plt.title("Distribution of Aggregate Ratings")
plt.xlabel("Aggregate Rating")
plt.ylabel("Number of Restaurants")
plt.grid(True)
plt.show()
```

.8] ✓ 0.1s



Generate + Code + Markdown

- Identified class imbalance

```
# Checking for imbalances
print("Value counts for Aggregate rating:")
print(df['Aggregate rating'].value_counts().sort_index())
```

✓ 0.0s

```
Value counts for Aggregate rating:
Aggregate rating
0.0    2148
1.8       1
1.9       2
2.0       7
2.1      15
2.2      27
2.3      47
2.4      87
2.5     110
2.6     191
2.7     250
2.8     315
2.9     381
3.0     468
3.1     519
3.2     522
3.3     483
3.4     495
3.5     480
3.6     458
3.7     427
3.8     399
3.9     332
...
4.7      41
4.8      25
4.9      61
Name: count, dtype: int64
```

Task 2: Descriptive Analysis

- Generated statistical summaries for numerical features

Level 1 - Task 2: Descriptive Analysis

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("/Users/arindamsharma/Desktop/Cognifyz Internship/Dataset.csv")
```

[1] ✓ 1.6s

```
# Descriptive statistics
print("Descriptive statistics for numerical columns:")
print(df.describe())
```

[2] ✓ 0.0s

```
Descriptive statistics for numerical columns:
       Restaurant ID  Country Code    Longitude     Latitude  \
count   9.551000e+03   9551.000000  9551.000000  9551.000000
mean    9.051128e+06     18.365616    64.126574    25.854381
std     8.791521e+06     56.750546    41.467058    11.007935
min     5.300000e+01      1.000000  -157.948486   -41.330428
25%     3.019625e+05      1.000000    77.081343    28.478713
50%     6.004089e+06      1.000000    77.191964    28.570469
75%     1.835229e+07      1.000000    77.282006    28.642758
max     1.850065e+07    216.000000   174.832089    55.976980

       Average Cost for two  Price range  Aggregate rating         Votes
count           9551.000000  9551.000000       9551.000000   9551.000000
mean            1199.210763     1.804837          2.666370    156.909748
std            16121.183073     0.905609          1.516378    430.169145
min                0.000000     1.000000          0.000000      0.000000
25%              250.000000     1.000000          2.500000      5.000000
50%              400.000000     2.000000          3.200000     31.000000
75%              700.000000     2.000000          3.700000    131.000000
max           800000.000000     4.000000          4.900000  10934.000000
```

```
# Median for numerical columns
print("\nMedian values:")
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
print(df[numerical_cols].median())
```

[3] ✓ 0.0s

```
Median values:
Restaurant ID         6.004089e+06
Country Code          1.000000e+00
Longitude             7.719196e+01
Latitude              2.857047e+01
Average Cost for two  4.000000e+02
Price range           2.000000e+00
Aggregate rating      3.200000e+00
Votes                 3.100000e+01
dtype: float64
```

- Identified top cuisines and cities

```
# Top country codes
print("\nTop Country Codes:")
print(df['Country Code'].value_counts().head())

# Top cities
print("\nTop cities by Number of Restaurants:")
print(df['City'].value_counts().head(10))
```

```
Top Country Codes:
Country Code
1      8652
216     434
215      80
30       60
214      60
Name: count, dtype: int64

Top cities by Number of Restaurants:
City
New Delhi       5473
Gurgaon         1118
Noida           1080
Faridabad        251
Ghaziabad         25
Bhubaneshwar      21
Amritsar          21
Ahmedabad         21
Lucknow           21
Guwahati          21
Name: count, dtype: int64
```

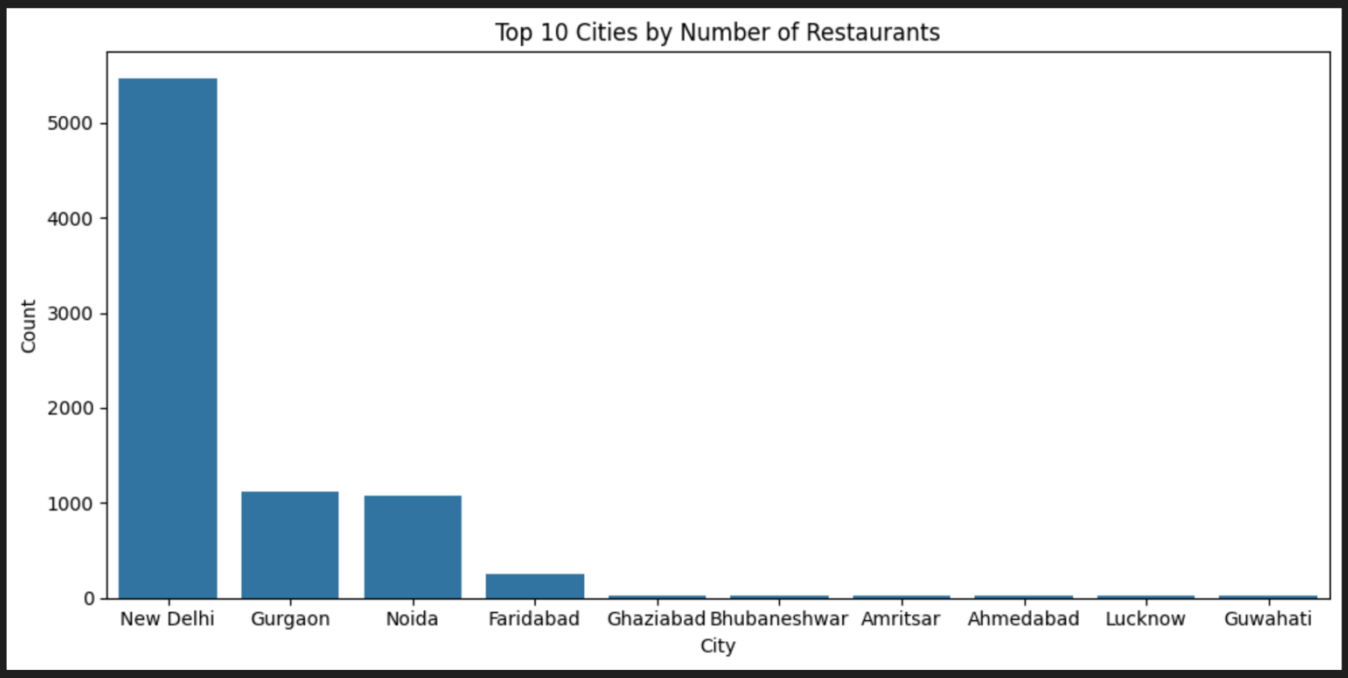
```
# Top cuisines
print("\nTop Most Common Cuisines:")
print(df['Cuisines'].value_counts().head(10))
```

```
Top Most Common Cuisines:
Cuisines
North Indian                          936
North Indian, Chinese                 511
Chinese                               354
Fast Food                             354
North Indian, Mughlai                 334
Cafe                                  299
Bakery                                218
North Indian, Mughlai, Chinese        197
Bakery, Desserts                      170
Street Food                           149
Name: count, dtype: int64
```

- Visualized city-wise restaurant counts

```
# Visual representation
top_cities = df['City'].value_counts().head(10)
plt.figure(figsize=(10, 5))
sns.barplot(x=top_cities.index, y=top_cities.values)
plt.title("Top 10 Cities by Number of Restaurants")
plt.xlabel("City")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```

✓ 0.1s



Task 3: Geospatial Analysis

- Visualized restaurant locations using latitude/longitude on a map

Level 1 - Task 3: Geospatial Analysis

```
import pandas as pd
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("/Users/arindamsharma/Desktop/Cognifyz Internship/Dataset.csv")
```

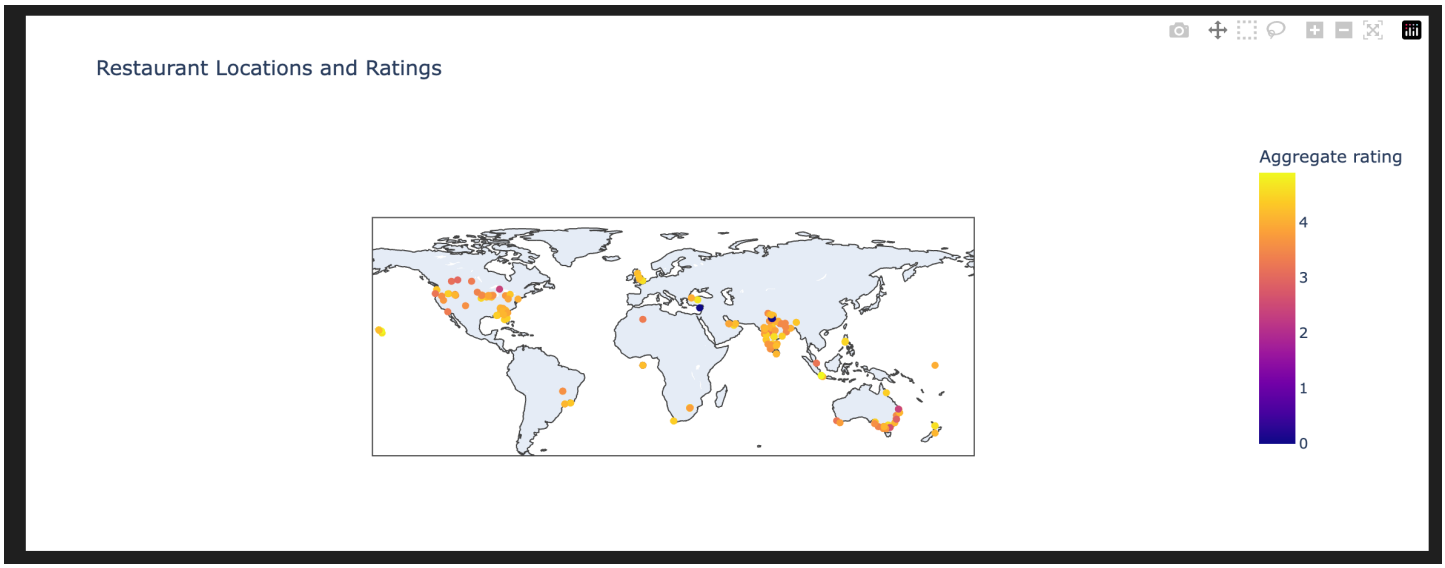
[1] ✓ 1.3s

```
# Dropping missing coordinates (if any)
df = df.dropna(subset=["Latitude", "Longitude"])
```

[2] ✓ 0.0s

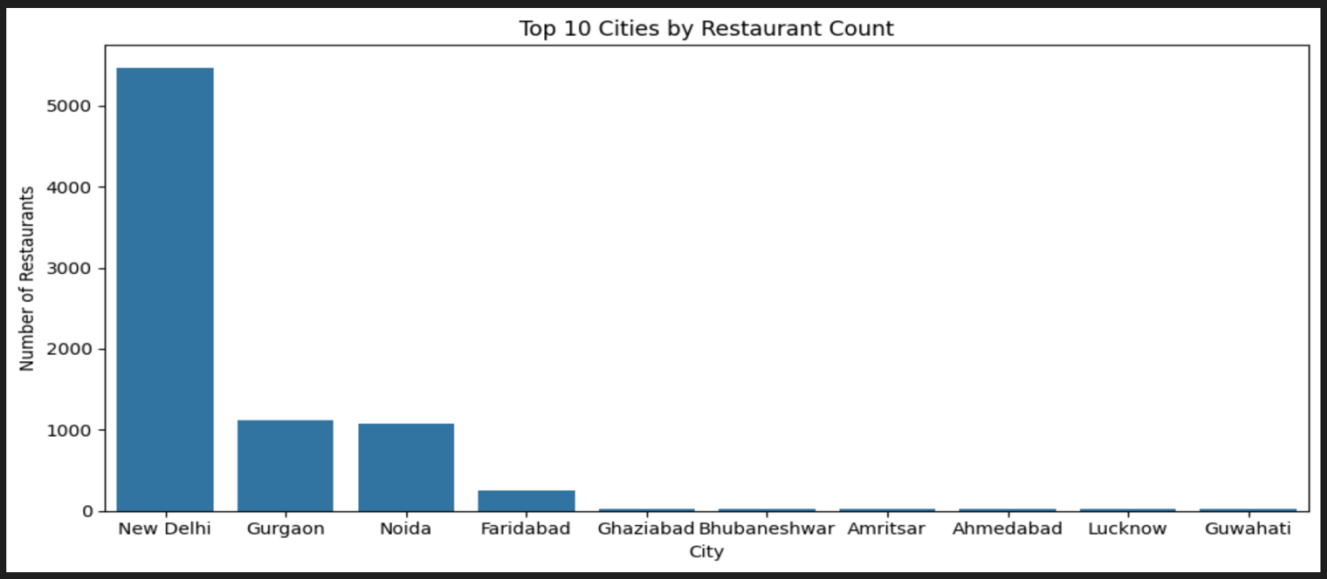
```
# Plotting restaurant locations using plotly
fig = px.scatter_geo(df, lat='Latitude', lon='Longitude',
                     color='Aggregate rating',
                     hover_name='Restaurant Name',
                     title='Restaurant Locations and Ratings')
fig.show()
```

[3] ✓ 0.6s



- Analyzed city-wise distributions

```python
# Restaurnat distribution across cities
top_cities = df['City'].value_counts().head(10)
plt.figure(figsize=(10, 5))
sns.barplot(x=top_cities.index, y=top_cities.values)
plt.title("Top 10 Cities by Restaurant Count")
plt.xlabel("City")
plt.ylabel("Number of Restaurants")
plt.tight_layout()
plt.show()
```



- Calculated correlation between location and ratings

```python
# Checking correlation between restaurant's location and rating
print("Correlation between Latitude, Longitude and Aggregate Rating:")
print(df[['Latitude', 'Longitude', 'Aggregate rating']].corr())
```

```
Correlation between Latitude, Longitude and Aggregate Rating:
                  Latitude  Longitude  Aggregate rating
Latitude          1.000000   0.043207          0.000516
Longitude         0.043207   1.000000         -0.116818
Aggregate rating  0.000516  -0.116818          1.000000
```

Level 2: Tasks Overview

Task 1: Table Booking and Online Delivery

- Calculated availability percentages for table booking and delivery

Level 2 - Task 1: Table Booking and Online Delivery

```
import pandas as pd

df = pd.read_csv("/Users/arindamsharma/Desktop/Cognifyz Internship/Dataset.csv")
```

✓ 0.0s

```
# Percentage of Restaurant's that offer Table Booking
table_booking_pct = df['Has Table booking'].value_counts(normalize=True) * 100
print("Table Booking Availability (%):\n", table_booking_pct)

# Percentage of Restaurant's that offer Online delivery
online_delivery_pct = df['Has Online delivery'].value_counts(normalize=True) * 100
print("\nOnline Delivery Availability (%):\n", online_delivery_pct)
```

✓ 0.0s

```
Table Booking Availability (%):
 Has Table booking
No     87.875615
Yes    12.124385
Name: proportion, dtype: float64

Online Delivery Availability (%):
 Has Online delivery
No     74.337766
Yes    25.662234
Name: proportion, dtype: float64
```

- Compared ratings for restaurants with/without table booking

```
avg_rating_booking = df.groupby('Has Table booking')['Aggregate rating'].mean()
print("Average Ratings with and without Table Booking:\n", avg_rating_booking)
```

✓ 0.0s

```
Average Ratings with and without Table Booking:
 Has Table booking
No     2.559359
Yes    3.441969
Name: Aggregate rating, dtype: float64
```

- Analyzed delivery availability by price range

```
delivery_by_price = pd.crosstab(df['Price range'], df['Has Online delivery'], normalize='index') * 100
print("Online Delivery Availability by Price Range (%):\n")
print(delivery_by_price)
```

✓ 0.0s

```
Online Delivery Availability by Price Range (%):

Has Online delivery         No        Yes
Price range
1                   84.225923  15.774077
2                   58.689367  41.310633
3                   70.809659  29.190341
4                   90.955631   9.044369
```

Task 2: Price Range Analysis

- Found most common price ranges

Level 2 - Task 2: Price Range Analysis

```
import pandas as pd

df = pd.read_csv("/Users/arindamsharma/Desktop/Cognifyz Internship/Dataset.csv")
```

✓ 0.0s

```
# Determining most common price range
price_counts = df['Price range'].value_counts()
print("Most common price ranges:\n", price_counts)
```

✓ 0.0s

```
Most common price ranges:
 Price range
1    4444
2    3113
3    1408
4     586
Name: count, dtype: int64
```

- Calculated average ratings per price range

```
#Average rating for each price range
avg_rating_by_price = df.groupby('Price range')['Aggregate rating'].mean()
print("Average rating for each price range:\n", avg_rating_by_price)
```

✓ 0.0s

```
Average rating for each price range:
 Price range
1    1.999887
2    2.941054
3    3.683381
4    3.817918
Name: Aggregate rating, dtype: float64
```

- Identified rating color for highest-rated price category

```python
# Color that represents highest average rating for each price range
best_price_range = avg_rating_by_price.idxmax()

best_range_df = df[df['Price range'] == best_price_range]

top_color = best_range_df['Rating color'].value_counts().idxmax()

print(f"The price range with the highest average rating is: {best_price_range}")
print(f"The most common rating color in that range is: {top_color}")
```

✓ 0.0s

```
The price range with the highest average rating is: 4
The most common rating color in that range is: Yellow
```

```python
#Visualisation
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.barplot(x=avg_rating_by_price.index, y=avg_rating_by_price.values)
plt.title("Average Rating by Price Range")
plt.xlabel("Price Range")
plt.ylabel("Average Rating")
plt.show()
```

✓ 0.1s

Task 3: Feature Engineering

- Created new features: name length and address length

Level 2 - Task 3: Feature Engineering

```
import pandas as pd

df = pd.read_csv("/Users/arindamsharma/Desktop/Cognifyz Internship/Dataset.csv")
```

1] ✓ 0.6s

```
#Additional features of the restaurant
df['Name Length'] = df['Restaurant Name'].apply(lambda x: len(str(x)))
df['Address Length'] = df['Address'].apply(lambda x: len(str(x)))

df[['Restaurant Name', 'Name Length', 'Address', 'Address Length']].head()
```

2] ✓ 0.0s

	Restaurant Name	Name Length	Address	Address Length
0	Le Petit Souffle	16	Third Floor, Century City Mall, Kalayaan Avenu...	71
1	Izakaya Kikufuji	16	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	67
2	Heat - Edsa Shangri-La	22	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	56
3	Ooma	4	Third Floor, Mega Fashion Hall, SM Megamall, O...	70
4	Sambo Kojin	11	Third Floor, Mega Atrium, SM Megamall, Ortigas...	64

- Encoded binary features for table booking and delivery

```
#Encoding Categorical variables:
df['Table_Booking_Encoded'] = df['Has Table booking'].map({'Yes': 1, 'No': 0})
df['Online_Delivery_Encoded'] = df['Has Online delivery'].map({'Yes': 1, 'No': 0})

df[['Has Table booking', 'Table_Booking_Encoded', 'Has Online delivery', 'Online_Delivery_Encoded']].head()
```

✓ 0.0s

	Has Table booking	Table_Booking_Encoded	Has Online delivery	Online_Delivery_Encoded
0	Yes	1	No	0
1	Yes	1	No	0
2	Yes	1	No	0
3	No	0	No	0
4	Yes	1	No	0

```
# Final check
print(df[['Name Length', 'Address Length', 'Table_Booking_Encoded', 'Online_Delivery_Encoded']].info())
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Name Length             9551 non-null   int64
 1   Address Length          9551 non-null   int64
 2   Table_Booking_Encoded   9551 non-null   int64
 3   Online_Delivery_Encoded 9551 non-null   int64
dtypes: int64(4)
memory usage: 298.6 KB
None
```

Tools and Technologies Used

- Python
- Pandas, NumPy
- Seaborn, Matplotlib, Plotly
- Jupyter Notebook, VS Code

Conclusion

This internship provided valuable hands-on experience with real-world datasets, data analysis, and feature engineering. I successfully applied data science techniques to explore restaurant trends and prepare data for modeling. I am grateful to Cognifyz Technologies for the opportunity to grow my skills in a practical environment.