End-to-End Restaurant Analytics with Zomato India Dataset

Introduction

The objective of this project is to analyze restaurant data across various cities in India to uncover meaningful insights related to dining patterns, cuisines, pricing, and overall restaurant performance. The dataset includes details such as restaurant names, locations, ratings, costs, service types, and more. To ensure accuracy, the data was first cleaned and preprocessed using Python, handling missing values and standardizing fields for better analysis.

After cleaning, an interactive dashboard was developed using Power BI to visualize key metrics and trends. The dashboard helps stakeholders explore city-wise restaurant distribution, identify top-rated cuisines and compare chain vs. single-outlet restaurants. This project demonstrates how data analytics can support data-driven decision-making in the competitive food service industry.

OVERVIEW

Source: Dataset sourced from Kaggle.
Total Rows: 0.2 Million Plus
Key Columns:

- Restaurant Name
- City
- Cuisines
- Average Cost for Two
- Rating
- Votes
- Establishment Type
- Online Delivery & Table Booking Availability
- Price Range
- Currencies
- Country Code

IMPORTING

It is performed in Python using Pandas.
Importing the data in Jupyter Notebook:

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Load dataset

```
df = pd.read_csv("C:/Users/Ankita Mall/Downloads/zomato_restaurants_in_India.csv", encoding='latin-1')
```

DATA CLEANING

Performed in Python using Pandas.

- Removing duplicates
- Dealing with missing values
- Omitting not useful features
- Gathering statistical Information
- Understand the structure

```
[7]: df.shape

[7]: (211944, 26)

[8]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   res_id                211944 non-null  int64
 1   name                  211944 non-null  object
 2   establishment         211944 non-null  object
 3   url                   211944 non-null  object
 4   address               211810 non-null  object
 5   city                  211944 non-null  object
 6   city_id               211944 non-null  int64
 7   locality              211944 non-null  object
 8   latitude              211944 non-null  float64
 9   longitude             211944 non-null  float64
 10  zipcode               48757 non-null   object
 11  country_id            211944 non-null  int64
 12  locality_verbose      211944 non-null  object
 13  cuisines              210553 non-null  object
 14  timings               208070 non-null  object
 15  average_cost_for_two  211944 non-null  int64
 16  price_range           211944 non-null  int64
```

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

[9]:

	res_id	city_id	latitude	longitude	country_id	average_cost_for_two	price_range	aggregate_rating	votes	photo_count	op
count	2.119440e+05	211944.000000	211944.000000	211944.000000	211944.0	211944.000000	211944.000000	211944.000000	211944.000000	211944.000000	
mean	1.349411e+07	4746.785434	21.499758	77.615276	1.0	595.812229	1.882535	3.395937	378.001864	256.971224	
std	7.883722e+06	5568.766386	22.781331	7.500104	0.0	606.239363	0.892989	1.283642	925.333370	867.668940	
min	5.000000e+01	1.000000	0.000000	0.000000	1.0	0.000000	1.000000	0.000000	-18.000000	0.000000	
25%	3.301027e+06	11.000000	15.496071	74.877961	1.0	250.000000	1.000000	3.300000	16.000000	3.000000	
50%	1.869573e+07	34.000000	22.514494	77.425971	1.0	400.000000	2.000000	3.800000	100.000000	18.000000	
75%	1.881297e+07	11306.000000	26.841667	80.219323	1.0	700.000000	2.000000	4.100000	362.000000	128.000000	
max	1.915979e+07	11354.000000	10000.000000	91.832769	1.0	30000.000000	4.000000	4.900000	42539.000000	17702.000000	

[11]:
```
df.columns
```

[11]:
```
Index(['res_id', 'name', 'establishment', 'url', 'address', 'city', 'city_id',
       'locality', 'latitude', 'longitude', 'zipcode', 'country_id',
       'locality_verbose', 'cuisines', 'timings', 'average_cost_for_two',
       'price_range', 'currency', 'highlights', 'aggregate_rating',
       'rating_text', 'votes', 'photo_count', 'opentable_support', 'delivery',
       'takeaway'],
      dtype='object')
```

```
[13]: df.drop_duplicates(inplace=True)
```

```
[15]: df.isnull().sum()
```

```
[15]: res_id                   0
      name                     0
      establishment            0
      url                      0
      address                 18
      city                     0
      city_id                  0
      locality                 0
      latitude                 0
      longitude                0
      zipcode              47869
      country_id               0
      locality_verbose         0
      cuisines               470
      timings               1070
      average_cost_for_two     0
      price_range              0
      currency                 0
```

```
[17]: df.drop(['url', 'address', 'latitude', 'longitude', 'zipcode', 'country_id', 'opentable_support'], axis=1, inplace=True)
```

```
[19]: df["establishment"].unique()
```

```
[19]: array(["['Quick Bites']", "['Casual Dining']", "['Bakery']", "['CafÃ©']",
             "['Dhaba']", "['Bhojanalya']", "['Bar']", "['Sweet Shop']",
             "['Fine Dining']", "['Food Truck']", "['Dessert Parlour']",
             "['Lounge']", "['Pub']", "['Beverage Shop']", "['Kiosk']",
             "['Paan Shop']", "['Confectionery']", '[]', "['Shack']",
             "['Club']", "['Food Court']", "['Mess']", "['Butcher Shop']",
             "['Microbrewery']", "['Cocktail Bar']", "['Pop up']",
             "['Irani Cafe']"], dtype=object)
```

```
[22]: # Remove brackets and single quotes
      df["establishment"] = df["establishment"].str.strip("[]").str.replace("'", "")

      # Replace empty strings with 'NA'
      df["establishment"] = df["establishment"].apply(lambda x: "NA" if x.strip() == "" else x)

      # Check results
      print(df["establishment"].unique())
```

```
['Quick Bites' 'Casual Dining' 'Bakery' 'CafÃ©' 'Dhaba' 'Bhojanalya' 'Bar'
 'Sweet Shop' 'Fine Dining' 'Food Truck' 'Dessert Parlour' 'Lounge' 'Pub'
 'Beverage Shop' 'Kiosk' 'Paan Shop' 'Confectionery' 'NA' 'Shack' 'Club'
 'Food Court' 'Mess' 'Butcher Shop' 'Microbrewery' 'Cocktail Bar' 'Pop up'
 'Irani Cafe']
```

```
[24]: df["cuisines"].unique()
```

- Almost 75% of our data had duplicate rows.
- 5 variables with some kind of missing values.
- 7 unnecessary columns are omitted.

[29]:
```
print(df["highlights"].nunique())
```

```
31455
```

[30]:
```
print(df["highlights"].unique())
```

```
["['Lunch', 'Takeaway Available', 'Credit Card', 'Dinner', 'Cash', 'Air Conditioned', 'Indoor Seating', 'Pure Veg']"
 "['Delivery', 'No Alcohol Available', 'Dinner', 'Takeaway Available', 'Lunch', 'Cash', 'Indoor Seating']"
 "['No Alcohol Available', 'Dinner', 'Takeaway Available', 'Breakfast', 'Lunch', 'Cash', 'Delivery', 'Outdoor Seating', 'Air Conditioned', 'Self Servic
e', 'Indoor Seating', 'Digital Payments Accepted', 'Pure Veg', 'Desserts and Bakes']"
 ...
 "['Dinner', 'Delivery', 'Cash', 'Takeaway Available', 'Free Parking', 'Digital Payments Accepted', 'Pure Veg', 'Indoor Seating']"
 "['Dinner', 'Cash', 'Takeaway Available', 'Lunch', 'Delivery', 'Free Parking', 'Indoor Seating', 'Air Conditioned', 'Outdoor Seating', 'Digital Payments
Accepted', 'Catering Available', 'Pure Veg']"
 "['Dinner', 'Cash', 'Takeaway Available', 'Debit Card', 'Delivery', 'Credit Card', 'Free Parking', 'Outdoor Seating']"]
```

[31]:
```
hl = []
df["highlights"].apply(lambda x : hl.extend(x[2:-2].split("', '")))
hl = pd.Series(hl)
print("Total number of unique highlights = ", hl.nunique())
```

```
Total number of unique highlights =  104
```

[32]:
```
df.to_csv('zomato_cleaned.csv', index=False)
```

Performed string normalization in Establishment, Highlights and Cuisines column.

Power BI Report, Modeling & Visualization

After completing the data cleaning and preprocessing in Python, the cleaned dataset was imported into Power BI to uncover deeper insights through interactive visualizations. Several supporting tables were created to facilitate analysis and improve performance:

- **Chains_Outlet_Count**: Categorizes restaurants as single or chain outlets based on the count of their occurrences.

```
Chains_Outlet_Count =
SUMMARIZE(
  'zomato_cleaned',
  'zomato_cleaned'[name],
  "Total Outlets", CALCULATE(COUNTROWS('zomato_cleaned'))
)
```

- **ChainTable**: Computes the average rating for each restaurant chain.

```
ChainsTable =
SUMMARIZE(
  'zomato_cleaned',
  'zomato_cleaned'[name],
  "Outlets", COUNT('zomato_cleaned'[name]),
   "Avg Rating", AVERAGE('zomato_cleaned'[aggregate_rating])
)
```

- **CuisineTable and HighlightTable:** Created for string normalization and to analyze cuisines and highlights more effectively.

```
CuisineTable =
SELECTCOLUMNS(
    GENERATE(
        FILTER(
            'zomato_cleaned',
            NOT ISBLANK('zomato_cleaned'[cuisines])
        ),
        VAR CuisineText = SUBSTITUTE('zomato_cleaned'[cuisines], ", ", "|")
        RETURN
        SELECTCOLUMNS(
            ADDCOLUMNS(
                GENERATESERIES(1, 5, 1),
                "SplitCuisine", TRIM(PATHITEM(CuisineText, [Value], TEXT))
            ),
            "Cuisine", [SplitCuisine]
        )
    ),
    "resi_id", 'zomato_cleaned'[res_id],
    "Restaurant Name", 'zomato_cleaned'[name],
    "Cuisine", [Cuisine]
)
```

```
HighlightTable =
SELECTCOLUMNS (
    FILTER (
        GENERATE (
            FILTER (
                'zomato_cleaned',
                NOT ISBLANK('zomato_cleaned'[highlights])
            ),
            VAR CleanText = SUBSTITUTE(SUBSTITUTE(SUBSTITUTE('zomato_cleaned'[highlights], "[", ""), "]", ""), "'", "")
            VAR SplitText = SUBSTITUTE(CleanText, ", ", "|")
            RETURN
                SELECTCOLUMNS (
                    ADDCOLUMNS (
                        GENERATESERIES(1, 10, 1),
                        "SingleHighlight", TRIM(PATHITEM(SplitText, [Value], TEXT))
                    ),
                    "resi_id", 'zomato_cleaned'[res_id],
                    "Restaurant Name", 'zomato_cleaned'[name],
                    "Highlight", [SingleHighlight]
                )
        ),
        NOT ISBLANK([Highlight])
    ),
    "resi_id", [resi_id],
    "Restaurant Name", [Restaurant Name],
    "Highlight", [Highlight]
)
```

Data Modeling :

The data model was designed using a **star schema** approach to ensure smooth filtering and accurate cross-table analysis. Relationships were established between the zomato_cleaned table and the supporting tables such as Chains_Outlet_Count, ChainTable, etc., using common keys like restaurant name or ID.

However, due to the many-to-many nature of fields like cuisines and highlights, CuisineTable and HighlightTable couldn't be directly related to the main table, and thus slicers do not impact their visuals. These were handled separately to preserve report integrity.

KPIs (Displayed on Cards)

To give stakeholders a quick overview, the following key performance indicators were shown using Power BI cards:

- Total Restaurants
- Average Cost for Two
- Unique Cuisines
- Average Rating
- Cities Covered

Visualizations :

The Power BI report includes various visuals to represent critical insights, such as:

- Number of restaurants by establishment type
- Number of restaurants by cuisine
- Number of restaurants by highlights
- Number of restaurants by city
- Number of restaurants by outlet type (Single vs Chain)
- Top 10 restaurant chains in India
- Top 7 restaurants by rating
- Most expensive restaurants

Filters and Slicers :

To make the dashboard interactive, the following slicers were implemented:

- City
- Restaurant Name
- Outlet Type

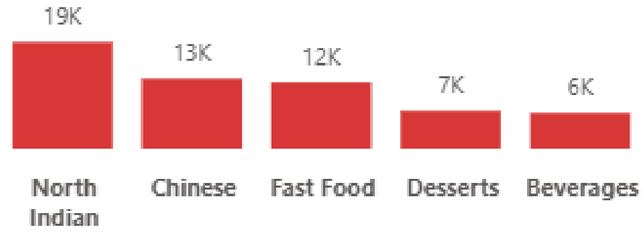
These slicers work across all connected visuals, thanks to the data model. Due to the limitations mentioned above, the slicers do not filter the cuisine and highlight visuals.

zomato India Restaurant Insights Dashboard

Outlet_Type
- chain
- Single Ou...

Total Restaurant	Avg cost of two	Unique cuisines	Cities covered	Average rating
56K	538	9K	99	3

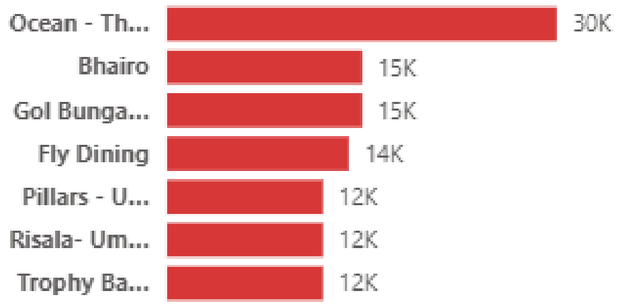
No. of Restaurant by Cuisine



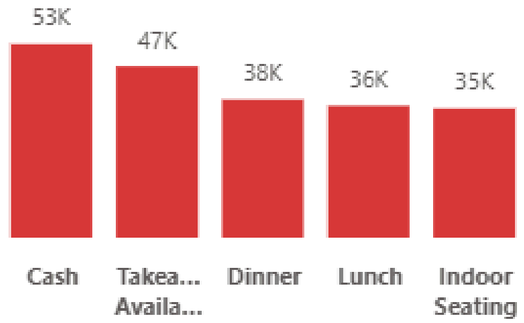
name: All

city: All

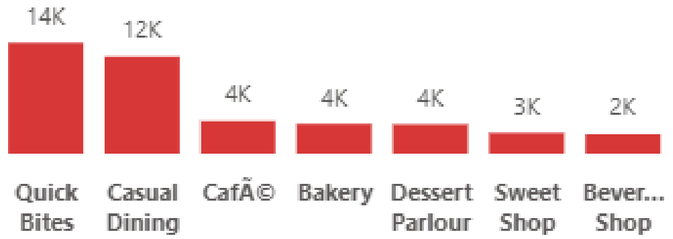
Expensive Restaurant



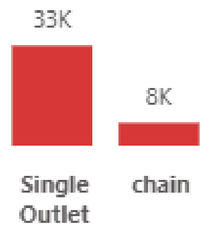
No. of Restaurant by Highlight



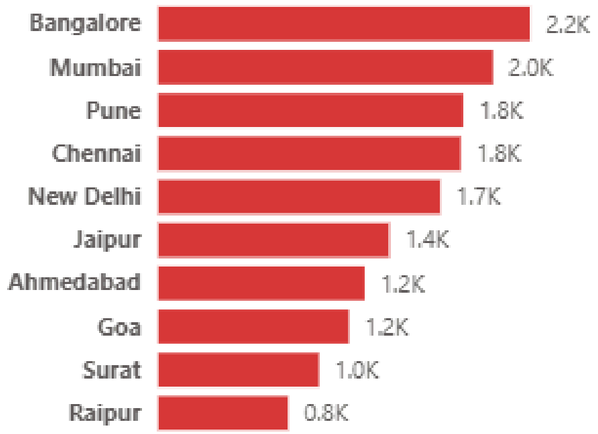
No. of Restaurant by establishment



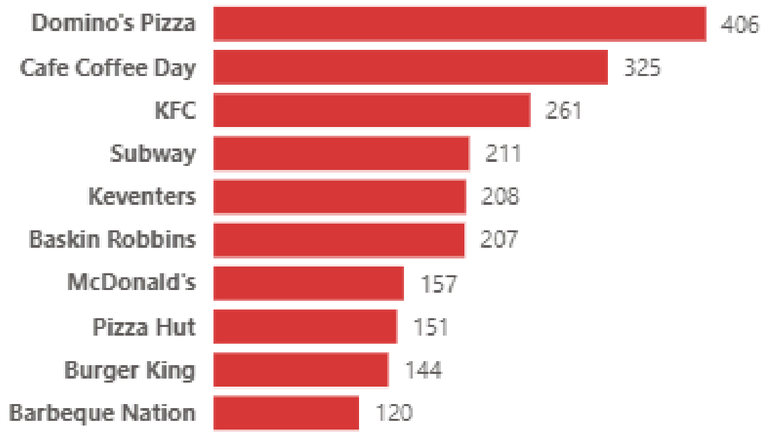
No. of Restaurant by Outlet_Type



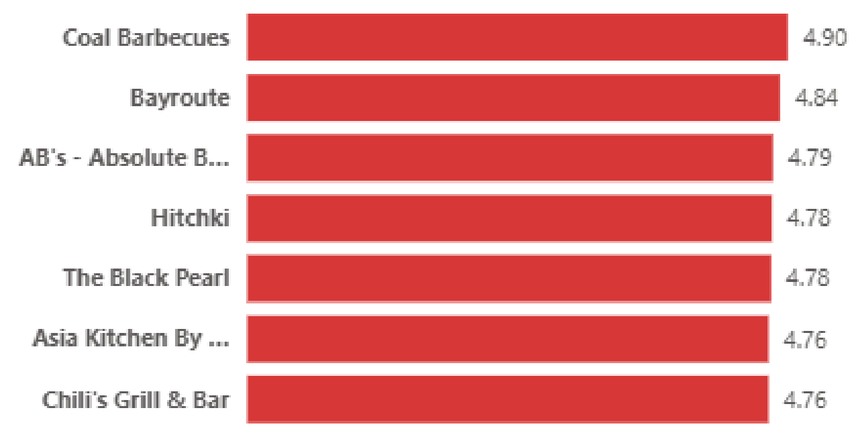
No. of Restaurant by city



Top 10 Restaurant chain in India



Top 7 Restaurant chain by rating



INSIGHTS

- "Quick Bites" and "Casual Dining" dominate the market with over 26k combined restaurants, showing strong preference for fast and casual food formats.
- North Indian, Chinese, and Fast Food are the most served cuisines, making up the majority of offerings.
- Cash payments, Takeaway, and Dinner options are the most common, with over 35k+ restaurants offering each.
- Bangalore, Mumbai, and Pune lead in restaurant counts, indicating high food demand and consumer density.
- Most restaurants (33k) are single outlets, but chains like Domino’s and KFC maintain strong brand presence and consistent ratings.
- Highly rated restaurants often have unique offerings and are part of premium or well-managed chains.
- Premium dining spots are limited but charge significantly high amounts, mostly located in heritage or luxury locations.

Recommendations

- **Target Growing Cities:** Focus expansion in Bangalore, Mumbai, and Pune due to dense food markets and customer interest.
- **Promote Chain Growth:** Encourage successful single outlets to scale into chains for higher brand value and ratings.
- **Menu Strategy:** Include top cuisines like North Indian and Chinese, but explore niche ones to stand out.
- **Enhance Highlights:** Offering multiple services (like lunch, dinner, takeaway) increases restaurant visibility and appeal.
- **Balance Pricing:** Stay close to the average cost for two (₹538) to stay competitive while offering value.

THANK YOU

Source Link