**Zomato Restaurant Project Article**

Project Description:

Zomato Data Analysis is one of the most useful analysis for foodies who want to taste the best

cuisines of every part of the world which lies in their budget. This analysis is also for those who

want to find the value for money restaurants in various parts of the country for the cuisines.

Additionally, this analysis caters the needs of people who are striving to get the best cuisine of

the country and which locality of that country serves that cuisines with maximum number of

restaurants.

**Problem Definition:**

The primary objective of this Zomato data analysis project is to help food enthusiasts identify the best value-for-money restaurants offering a variety of cuisines within their budget. This analysis aims to cater to individuals seeking to explore top-rated cuisines from different parts of the world, as well as those looking for the most popular and highly-rated restaurants in various localities. By leveraging machine learning techniques, this project strives to uncover patterns and insights that can guide users in making informed dining choices, ensuring they experience the best culinary delights without exceeding their budget.

**Why This Problem is Important:**

In today’s fast-paced world, dining out has become a significant part of people’s lifestyles. However, finding the best restaurants that offer high-quality food within a budget can be challenging. This problem is particularly relevant for:

1. **Food Enthusiasts:** Individuals who love exploring diverse cuisines and want to experience the best dishes from different parts of the world without overspending.
2. **Budget-Conscious Diners:** People who are mindful of their spending but still want to enjoy good food and dining experiences.
3. **Travelers:** Tourists and travelers looking to discover local culinary delights in various regions without breaking the bank.
4. **Local Residents:** Residents who want to find the best value-for-money restaurants in their locality.

**Impact of Solving This Problem:**

1. **Enhanced Dining Experiences:** By identifying the best value-for-money restaurants, food enthusiasts can enjoy high-quality meals without exceeding their budget, leading to more satisfying dining experiences.
2. **Informed Decision-Making:** Users can make informed decisions about where to dine based on data-driven insights, such as restaurant ratings, reviews, and cost-effectiveness.
3. **Increased Restaurant Visibility:** Restaurants that offer great value for money can gain more visibility and attract more customers, leading to increased business and customer satisfaction.
4. **Culinary Exploration:** People can explore a wider variety of cuisines and discover hidden gems in different localities, enriching their culinary experiences.
5. **Economic Benefits:** By promoting value-for-money dining options, the analysis can contribute to the local economy by driving more traffic to deserving restaurants.

**In summary, the Zomato data analysis project aims to bridge the gap between quality and affordability, helping diners make informed choices and enjoy the best culinary experiences within their budget.**

**Data Analysis:**

1) **Data Storage**:

This problem statement contains two datasets- \*\*Zomato.csv\*\* and \*\*country\_code.csv.\*\*

\*\*Country\_code.csv\*\* contains two variables:

• Country code

• Country name

2) **Dataset Link:**

• https://github.com/FlipRoboTechnologies/ML\_-Datasets/blob/main/Z\_Restaurant/Country-Code.xlsx

• https://raw.githubusercontent.com/FlipRoboTechnologies/ML\_-Datasets/main/Z\_Restaurant/zomato.csv

3) **Merge the dataframes:**

• Merge the dataframes(Zomato.csv and country\_code.csv) based on 'Country Code'

**4) Dimension of the Dataset:**

• Data set contains 9551 rows and 22 Columns

**5) Columns of the Dataset:**

Among these, the variables 'Average Cost for Two' and 'Price Range' are designated as targets for distinct analytical approaches. The remaining 20 variables serve as independent variables

Given the dataset's dual perspectives, we aim to analyze 'Average Cost for Two' as a regression problem and 'Price Range' as a classification problem during the Exploratory Data Analysis (EDA) phase

**6) Datatype of the Columns:**

This dataset contains

🡺14 object(string) value columns

🡺3 float64 value columns

🡺5 int64(integers) value columns

**7) Missing values:**

🡺After my initial Inspection , it seems the dataset is mostly complete, except for the ‘Cuisines’ feature, which has 9 instances of missing data.

8) **Dropping the columns:**

**🡺**Removing the columns 'Restaurant ID', 'Address', 'Switch to order menu' from the Dataframe.

🡺Reason for removing the column:

==>The 'Restaurant ID' column contains unique values identical to the dataset's total rows. This column serves solely for record identification and does not influence the target variable. Therefore, it can be safely dropped.

==>The 'Address' column displays high cardinality with 8,918 unique values out of a total of 9,551 rows, constituting approximately 93.4% of the data. Considering that 'Country', 'City', 'Longitude', and 'Latitude' already provide location information, retaining the 'Address' column may not contribute significantly to the model. Consequently, it is advisable to consider dropping this feature to streamline the dataset.

==>The column 'Switch to order menu' consists of only a single unique value. As it does not contribute any additional valuable information for the predictive model, it is advisable to drop this column.

**Now the dataset contains 9951 rows and 19 columns. The target variables and independent variables.**

**9) Summary statistics of the numerical and non-numerical variables:**

**==>**There is **no negative value present.**

**==>**However, The counts of 'Cuisines' have a difference from the total of rows whichmeans **there are missing values in this column .**

**==>**The mean value is larger than the median(50%) in columns 'Average Cost for two' and 'Votes'columns. Therefore, the **data could be skewed to the right**. 'Country Code' is categorical despite of being numeric. In the columns 'Longitude' and 'Latitude' the median(50%) is larger than the mean, then the **data could be skewed to the left**.

**==>**Since the max value is greater than two standard deviations plus the mean in the features 'Average Cost for two' and 'Votes'columns listed above, **there could be outliers in the data.**

**There are no duplicate records in the dataset**.

**Exploratory Data Analysis (EDA):**

**Data Visualization:**

**1)Univariate Analysis:**

**🡺Average Cost for two feature:**

The distribution of 'Average Cost for two' is skewed to the right and presents outliers in the Box plot.

**🡺Price range feature:**

Out of the total dataset, 4,444 entries (46.5%) fall within price range 1, followed by 3,113 entries (32.6%) in range 2, 1,408 entries (14.7%) in range 3, and 586 entries (6.1%) in range 4.

🡺**City :**

The dataset comprises observations for 141 cities, with the majority being in New Delhi (57.3%). Following New Delhi, the distribution includes Gurgaon (11.7%), Noida (11.3%), Faridabad (2.6%), and other cities with smaller proportions. For Guwahati, Ludknow, Ahmedabad, Amritsar, and Bhubaneshwar, there are additional observations, each representing 0.2% or less in the aforementioned list.

**🡺Currency :**

The dominant currency is the Indian Rupee (Rs.), accounting for 90.6% of occurrences, followed by the US Dollar ($) at 5.0%, the British Pound at 0.8%, and so forth. The least represented currencies are the Botswana Pula (P), Indonesian Rupiah (IDR), Qatari Rial (QR), and Sri Lankan Rupee (LKR) at 0.2%.

**🡺Has Table booking :**

Of the total observations, 87.9% indicate the absence of table bookings, while 12.1% indicate the presence of a booking.

🡺**Has Online delivery:**

Out of the total observations, 74.3% indicate the absence of online delivery, while 25.7% indicate the presence of online delivery.

🡺**Is delivering now:**

Out of the total observations, 99.6% indicate No for is delivering now, while 0.4% indicate Yes for is delivering now.

🡺**Rating color:**

In terms of rating color distribution, the dataset shows that 39.1% of the entries are labeled as Orange, 22.5% as White, 22% as Yellow, 11.3% as Green, 3.2% as Dark Green, and 1.9% as Red.

🡺**Rating text:**

In terms of rating text distribution, the dataset shows that 39.1% of the entries are labeled as Average, 22.5% as Not rated, 22% as Good, 11.3% as Very Good, 3.2% as Excellent, and 1.9% as Poor.

🡺**Country :**

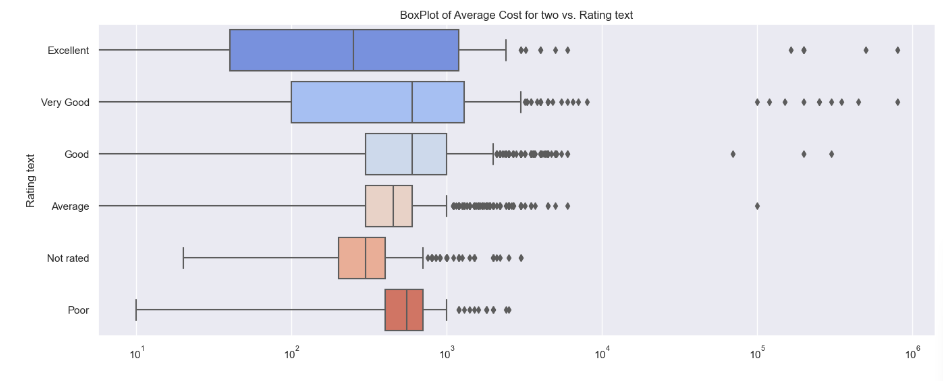
India, accounting for 90.6% of occurrences, followed by the United States at 4.5%, the United Kingdom at 0.8%, and so forth. The least represented currencies are the Phillipines, Indonesia, Singapure, Qatar, and Sri Lanka at 0.2%, and finally, Canada at 0%

🡺**Aggregate rating :**

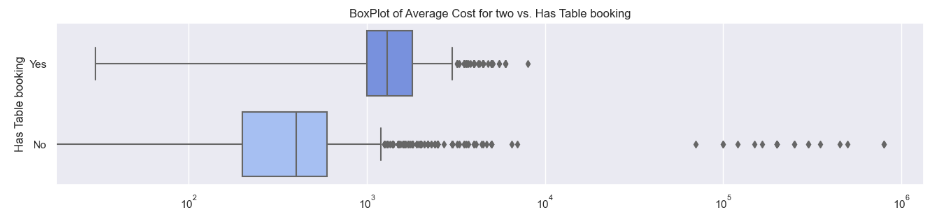
The distribution of the 'Aggregate rating' variable, representing the average rating out of 5, exhibits a nearly normal distribution with 33 unique values. Notably, there is a substantial number of zero ratings, constituting approximately 22.5% of the sample.

**2) Bivariate Analysis:**

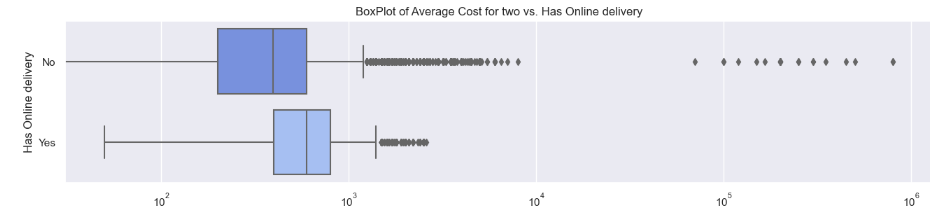
**🡺Average Cost for two distribution according to Rating:**

****

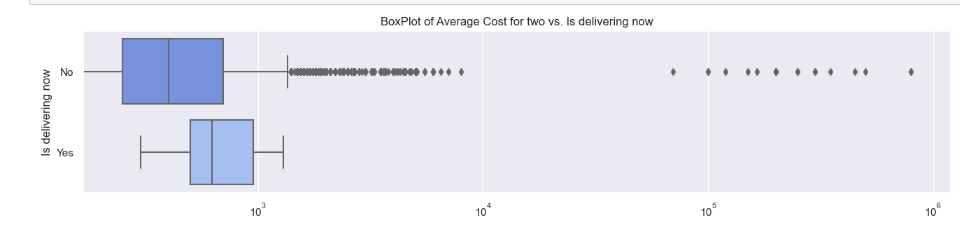
**🡺Average Cost for two distribution according to Has Table booking:**

****

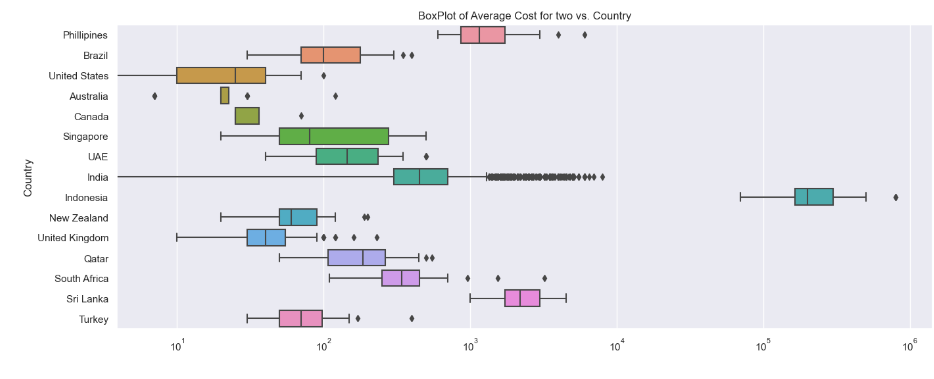
**🡺Average Cost for two distribution according to Has Online delivery:**

****

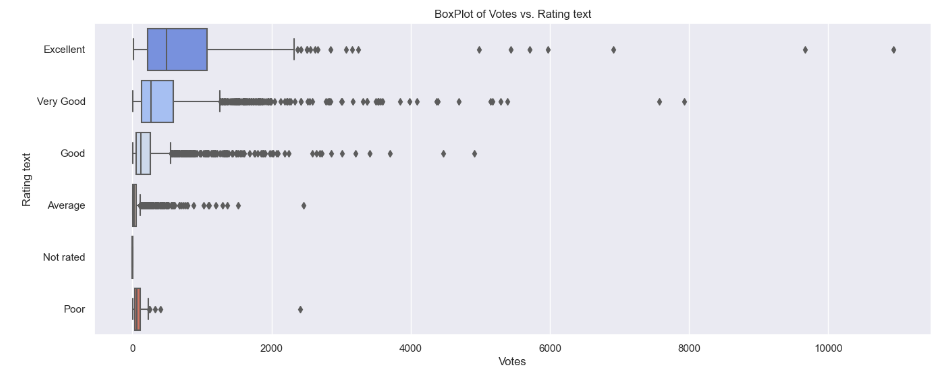
**🡺Average Cost for two distribution according to Is delivering now:**

****

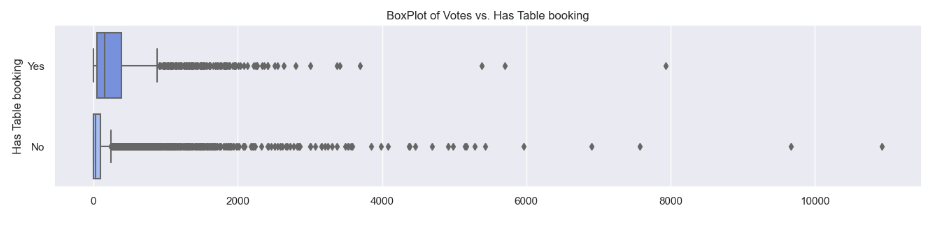
**🡺Average Cost for two distribution according to Country:**

****

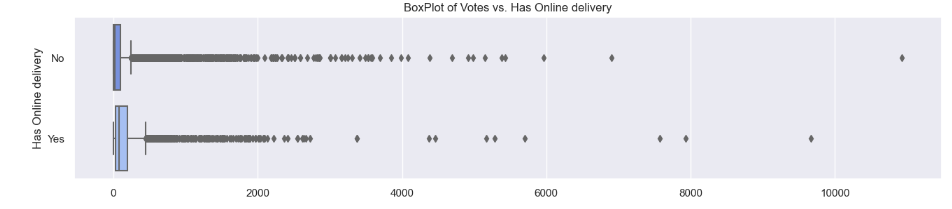
**🡺Votes distribution according to Rating:**

****

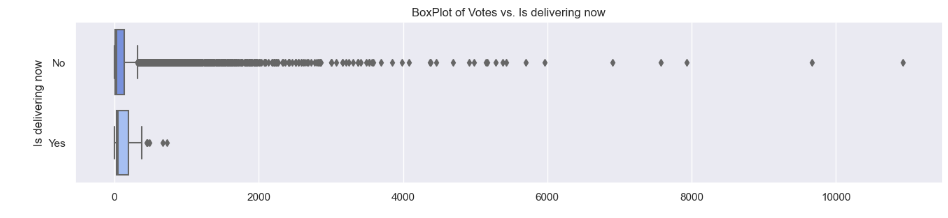
**🡺Votes distribution according to Has Table booking:**

****

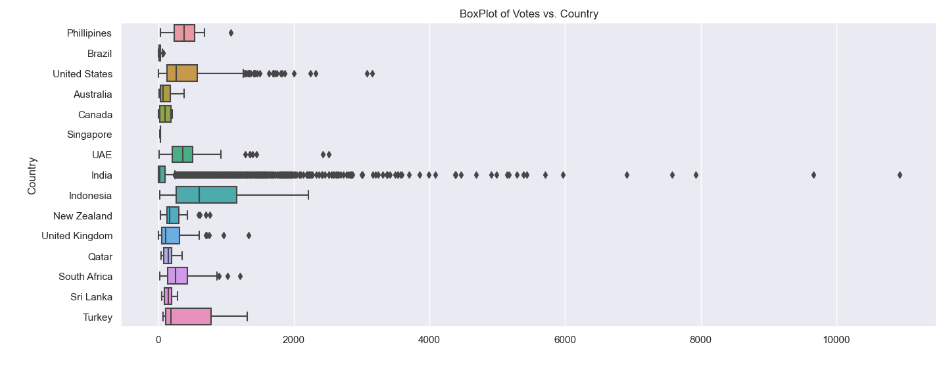
**🡺Votes distribution according to Has Online delivery:**

****

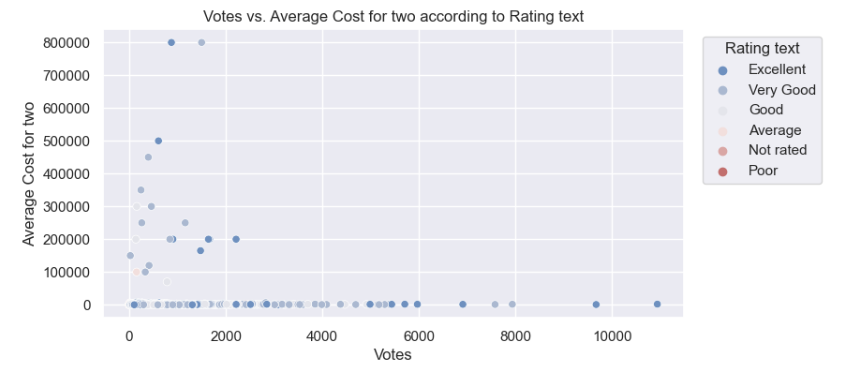
**🡺Votes distribution according to Is delivering now:**

****

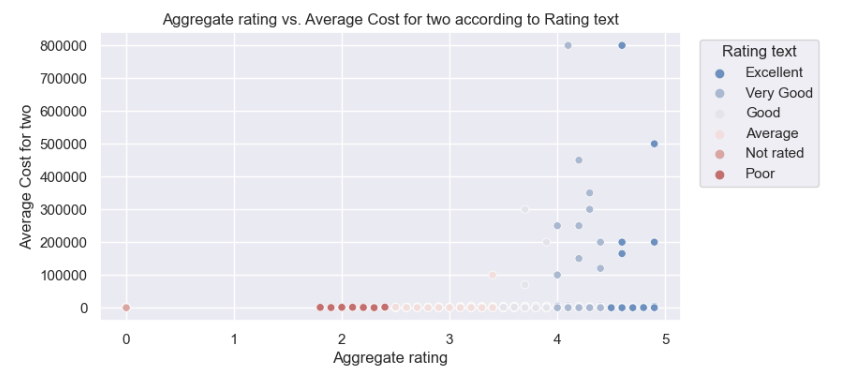
**🡺Votes distribution according to Country:**

****

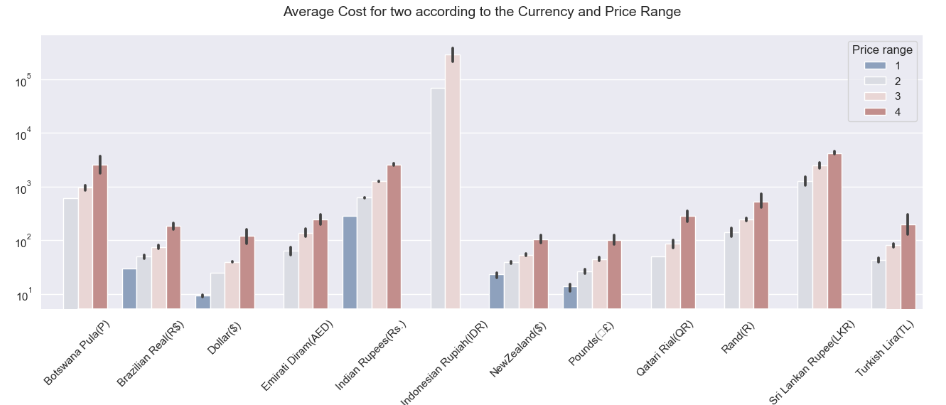
**🡺Comparing Average Cost with Votes:**

****

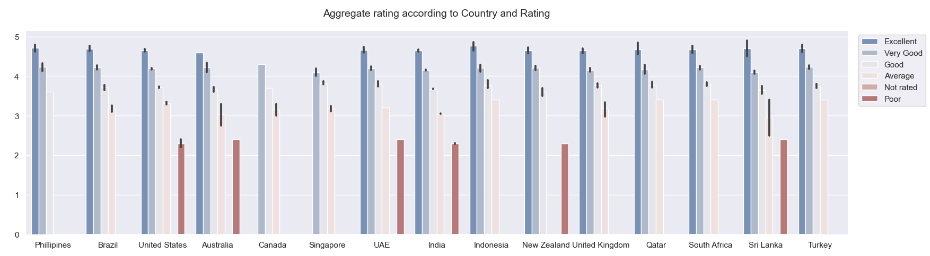
**🡺Comparing Average Cost with Aggregate Rating:**

****

**🡺Average Cost for two according to the Country and Price Range:**

****

**🡺 Aggregate rating according to Country and Rating:**

****

**🡺Restaurant location categorized by Price range:**

****

**3) Multivariate Analysis:**

**🡺Pairplot:**

A moderate positive correlation between the 'Average Cost for Two' variable and the 'Votes,' 'Price Range,'Ággregate rating' and 'Country Code' variables. Conversely, the 'Price Range' variable shows a weak positive correlation with the 'Votes,' 'Average Cost for Two,' 'Ággregate rating' and 'Country Code' variables**.**

**🡺Correlation between 'Average Cost for Two' and 'Price range' with independent variables:**

**1)**The variable 'Average Cost for Two' exhibits a weak correlation with all other variables.

2)The 'Price Range' variable shows a moderate positive correlation with 'Aggregate Rating' (+0.44), 'Votes' (+0.31), and 'Country Code' (+0.24).

3)Additionally, 'Country Code' demonstrates a moderate positive correlation with 'Longitude'(-0.70).

**Preprocessing Data:**

**1)Missing values:**

**🡺**It was identified that the dataset contains missing values in the 'Cuisines' column.

🡺Statistically, it is generally acceptable to omit observations with missing values if they constitute less than 5% of the overall sample. Therefore, we will establish a threshold of 2% for removal of missing observations.

🡺Dropping the missing values with % of missing less than 2%.

🡺After Dropping the missing values,

• 9 rows with missing values were deleted.

• This represent 0.09% of the data.

• In the new dataset there are 9542 rows and 19 columns.

2) **Feature Engineering:**

**why it was necessary?**

To address high cardinality issues in the categorical variables listed below, we will implement feature engineering and other strategies as outlined:

🡺'Restaurant Name' Column: Although 'Restaurant Name' exhibits high cardinality, it potentially influences both target variables. For instance, it might indicate whether the restaurant is associated with a well-known brand. Frequency-Based Encoding will be applied to this column.

🡺'Locality' and 'Locality Verbose' Columns: From the previous analysis, it is evident that both features provide very similar information. Therefore, 'Locality Verbose' will be dropped, and Frequency-Based Encoding will be applied to 'Locality' to handle its high cardinality.

🡺'Cuisines' Variable: This variable has a unique characteristic where each row can contain multiple cuisine categories. To address this, binary columns will be created for each cuisine category, and the categories will be aggregated to reduce the number of unique values. This approach aims to make the variable more manageable.

**Encoding binary columns:**

🡺Unique values for Has Table booking: ['Yes' 'No']

🡺Encoded values for Has Table booking: [1 0]

🡺Unique values for Has Online delivery: ['No' 'Yes']

🡺Encoded values for Has Online delivery: [0 1]

🡺Unique values for Is delivering now: ['No' 'Yes']

🡺Encoded values for Is delivering now: [0 1]

**Encoding categorical columns:**

**🡺**To encode the categorical features we will use LabelEncoder.

**Clustering Longitude and Latitude:**

🡺To reduce dimensionality and capture location patterns, we plan to employ unsupervised learning clustering algorithms, specifically KMeans. However, before applying KMeans, we need to determine the optimal value for K, the number of clusters. We will use two techniques for this purpose: the Elbow Method and the Silhouette Score.

**Silhouette Score**

🡺The silhouette score measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). Choose the K that maximizes the silhouette score, corresponding to the peak in the silhouette score plot.

**Elbow Method:**

🡺After fitting the KMeans algorithm with a range of K values, for each K, compute the sum of squared distances from each point to its assigned center. Plot the sum of squared distances against K, and look for the "elbow" point in the plot, where the rate of decrease sharply changes.

The "elbow" is where you might see a bend or a significant change in the slope of the curve.

**Create the clusters:**

🡺We applied the KMeans algorithm with a range of K=6 values to create the 'LocationCluster' based on the 'Latitude' and 'Longitude' columns.

**Grouping and Aggregating Categories based on 'Cuisines' feature:**

**🡺**You can group or aggregate categories that share similar characteristics. This can reduce the number of unique values and make the variable more manageable.

**Drop unnecessary columns:**

**🡺**After feature Engineering we have to remove non-needed columns.

**3) Removing outliers:**

**🡺** We selected the features with outliers from the Univariate Analysis and considered only the continuous ones. The outliers in the target variable would not be removed.

🡺 Detect outliers with zscore method.

🡺 threshold = 3

🡺 Since we can not afford to lose more than 10% of the data, we have selectively retained the continuous variables with outliers based on the Univariate Analysis. Adopting a threshold of 3, the data culling process resulted in a loss of only 1.98%. As a consequence, our current dataset comprises 9,353 rows and 159 columns.

4) **Skewness Correction:**

**🡺** Checking the skewness in the data - we do not considere the created columns based on Cuisine since are categorical.

🡺 We do not considere the created columns based on Cuisine since are categorical

🡺Skewness acceptable range -0.5 to +0.5.

🡺 We will focus on the numerical columns 'Aggregate rating','Average Cost for Two' and 'Votes.' It's important to note that 'Average Cost for Two' serves as the target variable for the initial prediction task. Therefore, any transformations applied to it during modeling should be reversed after making predictions.

🡺 We are going to use the cube-root method, square-root method, and log method to transform the columns with a skewness of more than 0.5. Later, we will compare the resulting skewness in order to select the best transform method.

🡺>cube-root transform method is better for columns 'Average Cost for two' and 'Votes'.

🡺>Do no apply any transform method is better for columns 'Aggregate rating'.

5) **Feature Scaling using Standard Scalarization:**

**Separating features and label for regression model to predict Average Cost for two:**

**🡺**Feature Dimension = (9353, 158)

🡺Label Dimension = (9353,)

**Separating features and label for classification model to predict Price range:**

🡺Feature Dimension = (9353, 158)

🡺Label Dimension = (9353,)

**Scaling Data:**

**🡺** from sklearn.preprocessing import StandardScaler

🡺Fit and Tranform the x and x2 for regression and classification model.

**Muticollinearity Analysis and Feature Selection:**

**Checking VIF for regression model 1 to predict Average Cost for two:**

🡺Upon examining the Variance Inflation Factor (VIF) values, we identified certain features that exhibit multicollinearity:

==>Malaysian (VIF: 32.260764)

==>Malay (VIF: 31.944276)

==>Indian (VIF: 14.534695)

==>North Indian (VIF: 12.918888)

🡺VIF values greater than 10 suggest a high correlation among these features. To address this issue, we will begin by dropping one of the columns. If the multicollinearity persists, we will then consider removing the column with the highest VIF.

**🡺**We drop the 'Malay' and 'Indian' columns to solve the multicollinearity problem since the first one was the second with higher VIF value, then the feature remaining highest VIF value.

🡺Dropping the 'Malay' and 'Indian' columns solved the multicollinearity issue. We can now move ahead with model building.

**Checking VIF for classification model 2 to predict Price range:**

**🡺** Upon examining the Variance Inflation Factor (VIF) values, we identified certain features that exhibit multicollinearity:

==>Malaysian (VIF: 32.262503)

==>Malay (VIF: 31.947120)

==>Indian (VIF: 14.546492)

==>North Indian (VIF: 12.941748)

🡺VIF values greater than 10 suggest a high correlation among these features. To address this issue, we will begin by dropping one of the columns. If the multicollinearity persists, we will then consider removing the column with the highest VIF.

🡺 We drop the 'Malay' and 'Indian' columns to solve the multicollinearity problem since the first one was the second with higher VIF value, then the feature remaining highest VIF value.

🡺 Dropping the 'Malay' and 'Indian' columns solved the multicollinearity issue. We can now move ahead with model building.

**6) OverSampling:**

**🡺**As we see previously in the Section . The dataset was imbalance. So, we are going to apply SMOTE for oversampling the data**.**

**🡺** After appling SMOTE ,The data is balanced, enabling the construction of machine learning classification models.

**Modeling:**

**Modeling for predicting Average Cost for two:**

**🡺** Import Regression Algorithms.

🡺Finding the best random state.(Maximum r2 score is 0.8383 at random\_state 154).

🡺Creating train test split

🡺Creating a Function with R2\_score,Mean\_Absolute\_Error,Mean\_Squared\_Error,Random\_mean\_squared\_error, difference between R2 score and cross validation.

**LinearRegression:**

**🡺**R2\_Score:83.83%(Test Result)

🡺Difference between R2\_score and cross validation is: 9.47937

**RandomForestRegressor:**

**🡺** R2\_Score:90.45%

🡺 Difference between R2\_score and cross validation is: 0.39761.

**SVR:**

**🡺** R2\_Score:72.86%

🡺 Difference between R2\_score and cross validation is: 0.15026

**KNeighborsRegressor:**

**🡺** R2\_Score:71.27%

🡺 Difference between R2\_score and cross validation is: 0.26427

**Lasso:**

**🡺** R2\_Score:25.79%

🡺 Difference between R2\_score and cross validation is: 0.05346

**Ridge:**

**🡺** R2\_Score:83.83%.

🡺 Difference between R2\_score and cross validation is: 0.36736

**GradientBoostingRegressor:**

**🡺** R2\_Score:89.83%

🡺 Difference between R2\_score and cross validation is: 0.18241

**DecisionTreeRegressor:**

**🡺** R2\_Score:84.02%

🡺 Difference between R2\_score and cross validation is: 0.38437

From the summary of the models results and comparing the cross-validation scores (CV\_Mean) and R2 score (test), we conclude Random Forest Regressor is our best-performing model since the two metrics are very close , this indicates that the model is performing consistently on both the test set and across different folds in cross-validation.

This consistency is a positive sign, suggesting that the model generalizes well to new data.

In order to apply Hyper Parameter tunning we are going to select **Random Forest Regressor as our final model.**

**Hyper Parameter Tuning:**

**🡺** Finding the best parameters for Random Forest Regressor.

🡺 Best Parameters for RandomForestRegressor model:

{'n\_estimators': 30,

'min\_samples\_split': 9,

'min\_samples\_leaf': 1,

'max\_depth': 70,

'bootstrap': True}

🡺 Create the model with the best parameters

**Saving the model:**

**🡺** Saving the model using .pkl.( models/averagecostfor2\_regressor\_model.pkl)

🡺 Predicting the saved model.

🡺 The predicted values in a dataset to compared the prediction with the test data with R2 score 90.548%.

**Modeling for predicting Price range:**

**🡺** Import ClassificationAlgorithms.

🡺Finding the best random state.( Best accuracy is 98.01% at random\_state 42).

🡺Creating train test split

🡺Creating a Function with accuracy score,confusion matrix,classification report and difference between accuracy score and cross validation

.

**LogisticRegression:**

**🡺** Accuracy Score:90.00%(Test Result)

🡺Difference between accuracy score and cross validation is: 0.01167

**RandomForestClassifier:**

**🡺**Accuracy Score:97.74%

🡺Difference between accuracy score and cross validation is: 0.02240

**SVC(Support Vector Machine Classifier):**

**🡺**Accuracy Score:88.81%

🡺Difference between accuracy score and cross validation is: 0.01066

**GradientBoostingClassifier:**

**🡺**Accuracy Score:95.98%

🡺Difference between accuracy score and cross validation is: 0.02357

**AdaBoostClassifier:**

**🡺**Accuracy Score:77.45%

🡺Difference between accuracy score and cross validation is: 0.06273

**BaggingClassifier:**

**🡺**Accuracy Score:97.41%

🡺Difference between accuracy score and cross validation is: 0.04141

**ExtraTreesClassifier:**

**🡺** Accuracy Score:96.28%

🡺 Difference between accuracy score and cross validation is: 0.03182

**DecisionTreeClassifier:**

**🡺**Accuracy Score:96.52%

🡺Difference between accuracy score and cross validation is: 0.04679

**The model with the best accuracy is RandomForestClassifier with 97.74% of accuracy.**

After applying cross-validation technique we observe the better cross-validation score is for RandomForestClassifier and the difference with accuracy score is 0.0224011 confirming as follows:

**RandomForestClassifier is our best Model**

**Hyper Parameter Tuning:**

**🡺** Finding the best parameters for Random Forest Classifier

🡺 Best Parameters for RandomForestClassifiermodel:

{'n\_estimators': 295,

'max\_leaf\_nodes': 30,

'max\_features': None,

'max\_depth': 100,

'min\_samples\_split': 4,

}

🡺 Create the model with the best parameters

**Saving the model:**

**🡺** Saving the model using .pkl.( models/price\_range\_cl\_model.pkl)

🡺 Predicting the saved model.

🡺 These values represent the predictions on the fraction of the dataset reserved for testing, enabling a comparison with an accuracy of 97.687%.

**Final Conclusions:**

🡺The given dataset comprises 9551 rows and 22 columns; however, our univariate/bivariate analysis was conducted on only 19 features. Columns such as 'Restaurant ID', 'Address', and 'Switch to order menu' were dropped as they serve only for identification purposes and do not contribute value to the prediction model.

🡺We have two target variables: 'Average Cost for Two', which is continuous, prompting the development of a regression model, and 'Price Range,' a categorical variable with four possible values, requiring the creation of a classification model."

🡺The chosen regression model is the RandomForestRegressor, achieving a R2 score of 90**.**548**%**for predicting 'Average Cost for Two'.

🡺For the classification task, the RandomForestClassifier achieved an accuracy score of 97.687%in predicting 'Price Range'.

🡺The dataset do not present duplicates.

🡺Handling missing values resulted in the removal of 9 rows, constituting a minimal 0.09% loss of data.

🡺Outlier treatment involved eliminating 1.98% of the total records (189 rows). Treshold was set as 3.

🡺To mitigate potential multicollinearity issues, two columns were dropped from both the regression and classification model datasets.