Capstone Project

Zomato Sentiment Analysis

1. Lalit Ahirrao
2. Aniket Gajmal
3. Rushikesh Pawar
4. Prasad Ghegade
5. Samarth Gangurde

Table of Content

Reason Behind the Project

Dataset Information

Data Summary

Features Analysis

Data Preprocessing

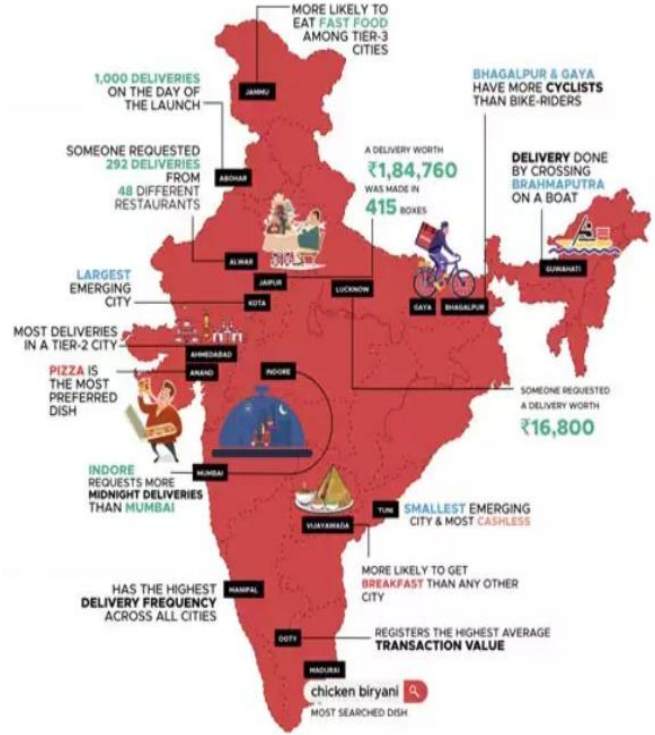
Exploratory Data Analysis

Implementing Algorithms

Challenges

Conclusion

Reason behind the project


MORE LIKELY TO EAT FAST FOOD AMONG TIER-3 CITIES
JAMMU
1,000 DELIVERIES ON THE DAY OF THE LAUNCH
ABOHAR
SOMEONE REQUESTED 292 DELIVERIES FROM 48 DIFFERENT RESTAURANTS
ALWAR
LARGEST EMERGING CITY
JAIPUR
KOTA
MOST DELIVERIES IN A TIER-2 CITY
AHMEDABAD
PIZZA IS THE MOST PREFERRED DISH
ANAND
INDORE
INDORE REQUESTS MORE MIDNIGHT DELIVERIES THAN MUMBAI
MUMBAI
A DELIVERY WORTH ₹1,84,760 WAS MADE IN 415 BOXES
LUCKNOW
BHAGALPUR & GAYA HAVE MORE CYCLISTS THAN BIKE-RIDERS
GAYA
BHAGALPUR
DELIVERY DONE BY CROSSING BRAHMAPUTRA ON A BOAT
GUWAHATI
SOMEONE REQUESTED A DELIVERY WORTH ₹16,800
TUNI
SMALLEST EMERGING CITY & MOST CASHLESS
VIJAYAWADA
MORE LIKELY TO GET BREAKFAST THAN ANY OTHER CITY
HAS THE HIGHEST DELIVERY FREQUENCY ACROSS ALL CITIES
MANIPAL
OOTY
REGISTERS THE HIGHEST AVERAGE TRANSACTION VALUE
MADURAI
chicken biryani
MOST SEARCHED DISH

India is quite famous for its diverse multi cuisine available in a large number of restaurants and hotel resorts, which is reminiscent of unity in diversity. Restaurant business in India is always evolving. More Indians are warming up to the idea of eating restaurant food whether by dining outside or getting food delivered. The growing number of restaurants in every state of India has been a motivation to inspect the data to get some insights, interesting facts and figures about the Indian food industry in each city. So, this project focuses on analysing the Zomato restaurant data for each city in India.

Problem Statement:-

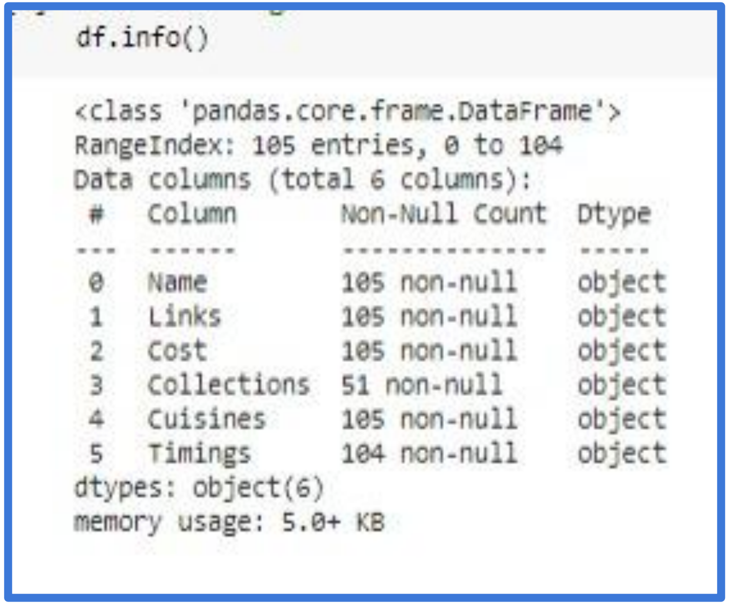
The Project focuses on Customers and Company, you have to analyze the sentiments of the reviews given by the customer in the data and made some useful conclusion in the form of Visualizations. Also, cluster the zomato restaurants into different segments. The data is visualized as it becomes easy to analyse data at instant. The Analysis also solve some of the business cases that can directly help the customers finding the Best restaurant in their locality and for the company to grow up and work on the fields they are currently lagging in.

Dataset Information

This dataset contains information on Names, Links, cost, collection, cuisines, Timings, Restaurants, Reviewer, Review, Rating, MetaData, Time and pictures. To cluster a data points and for sentiment analysis

Dataset 1 :

- 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

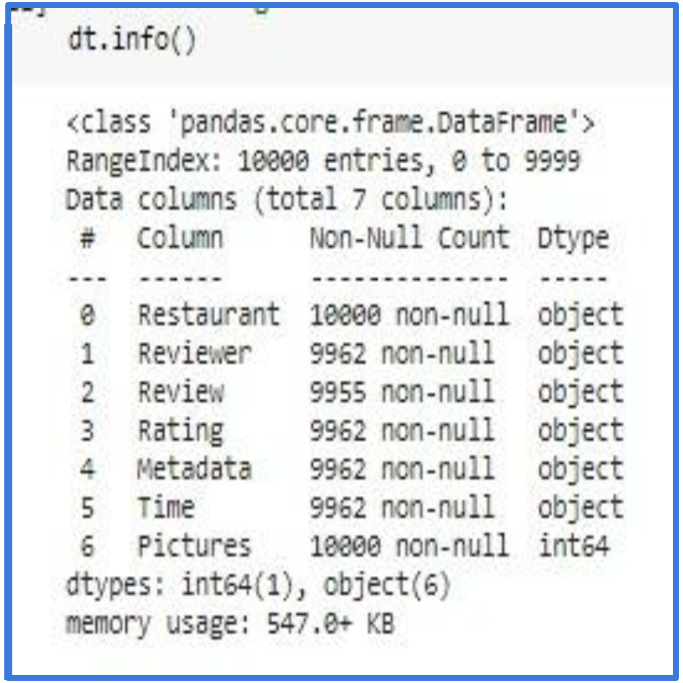


```
df.info()

<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: 5.0+ KB
```

Dataset 2 :

- 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



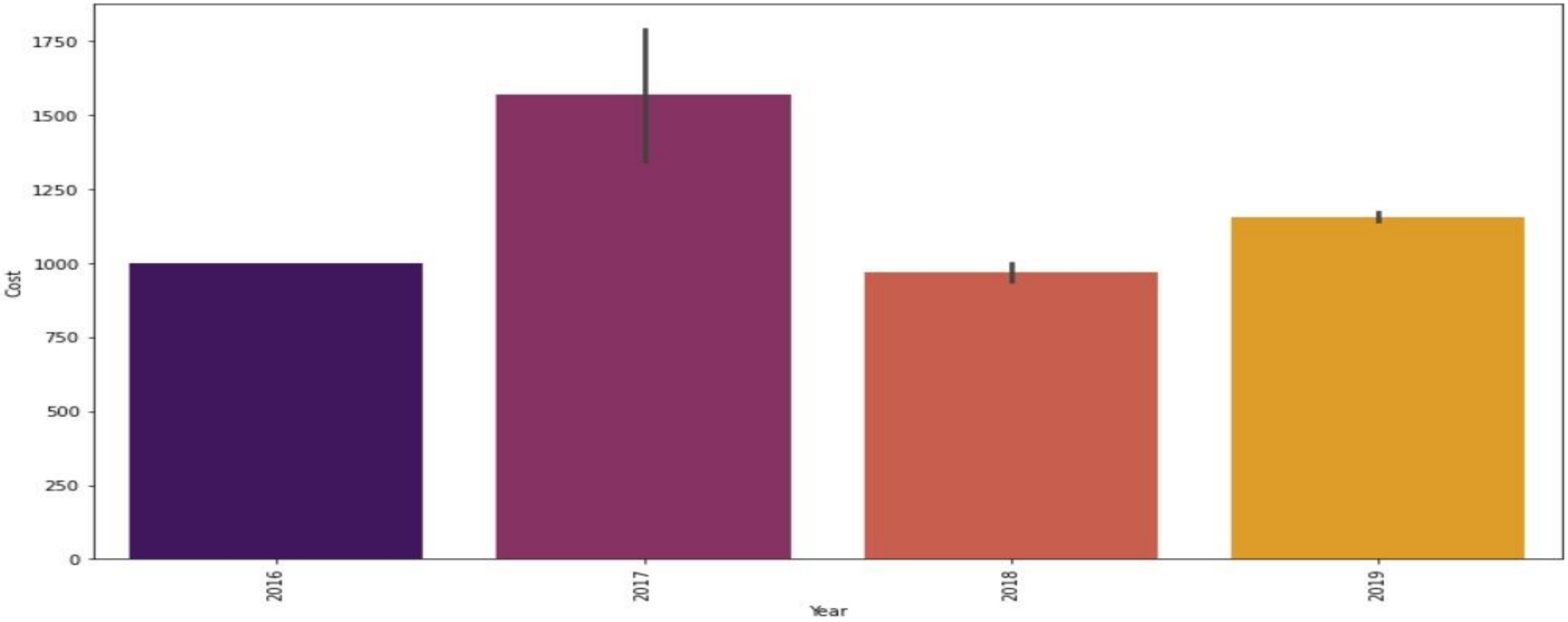
```
dt.info()

<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
```

Preprocessing of dataset

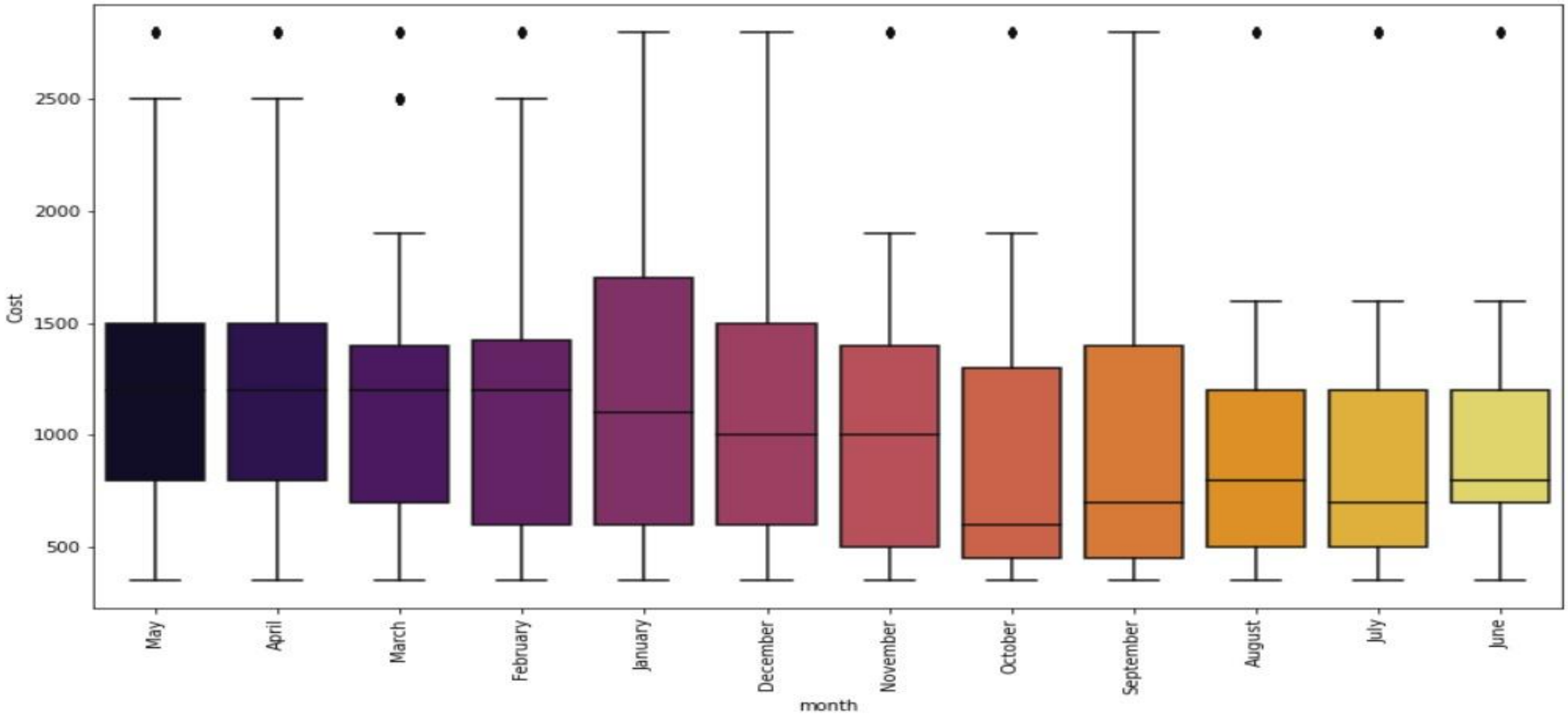
In preprocessing we merged two datasets by using left joint, dropped unnecessary columns like links, dropped missing values which were negligible and replace considerable missing values with the others, Renamed the columns for better understanding, extracted Year, Month, Day from the Date column.Converted Time variable into categorical variable by replacing time to Morning,Afternoon and Evening

Exploratory Data Analysis



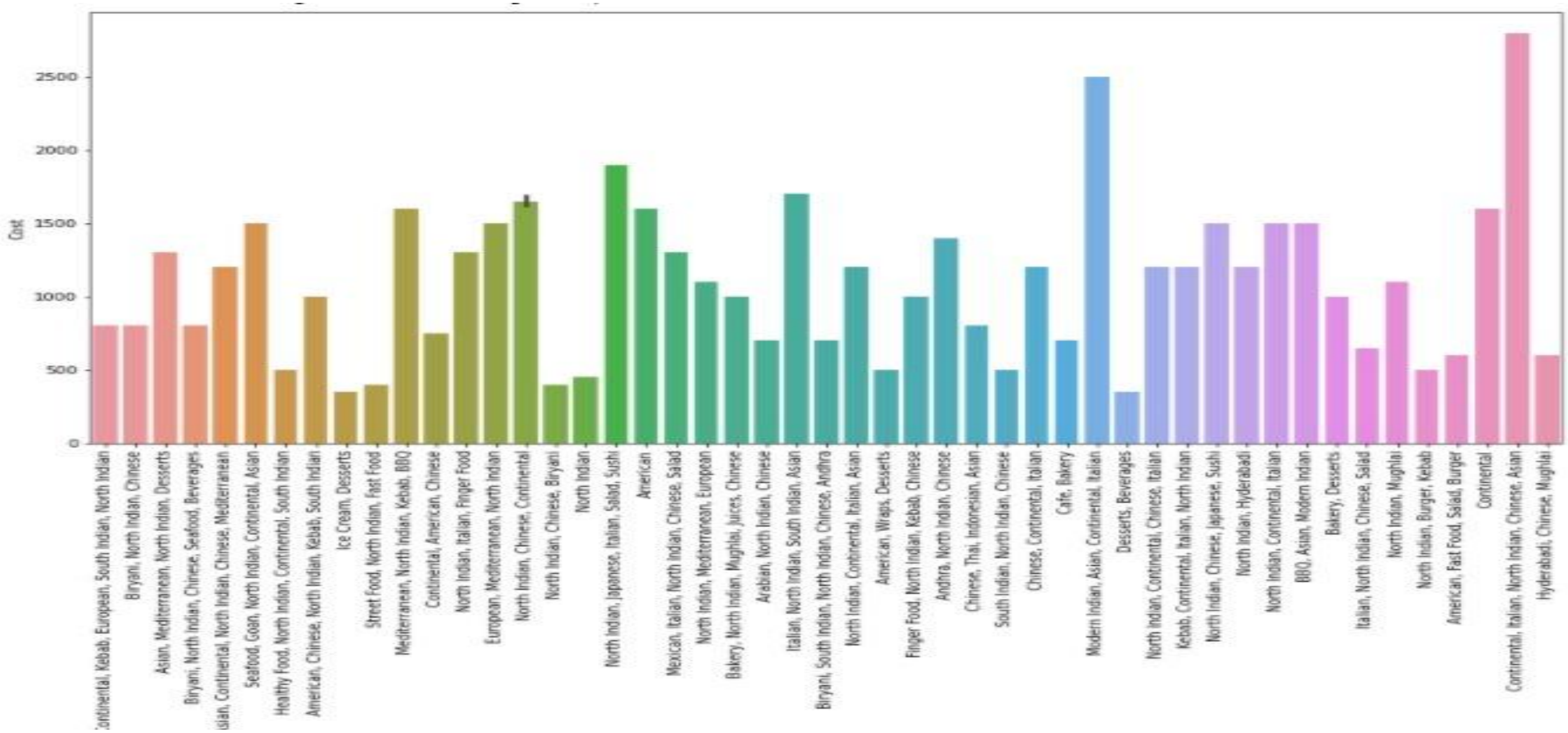
2017 was the expensive year according to the above tables

ANALYSIS OF MONTH VARIABLE



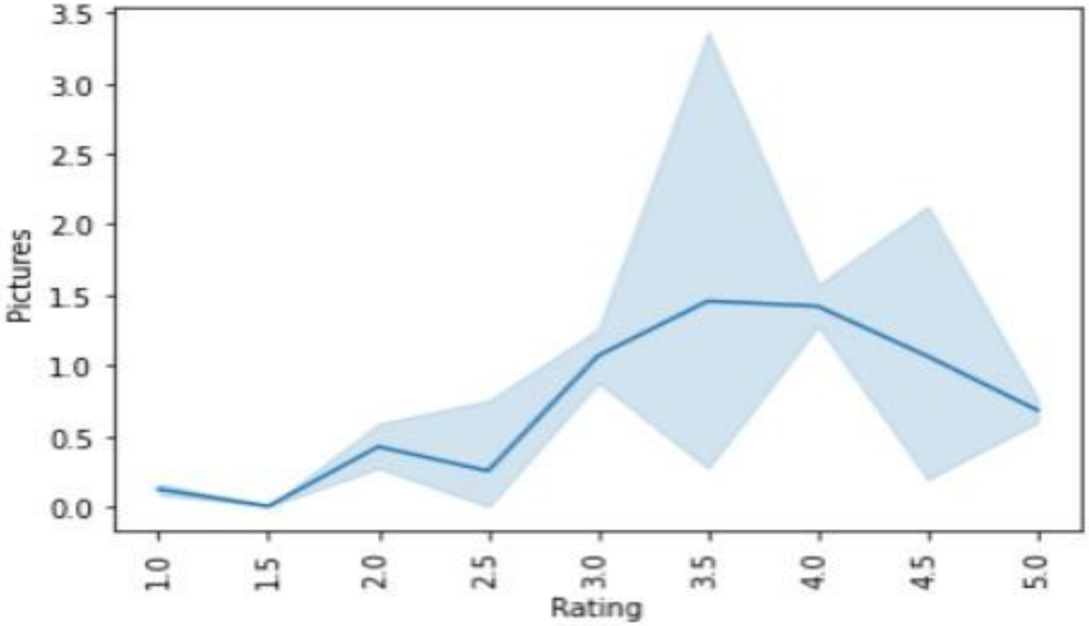
Avg cost in January, December, September is too high where as its low in July and June

ANALYSIS OF CUISINES VARIABLE



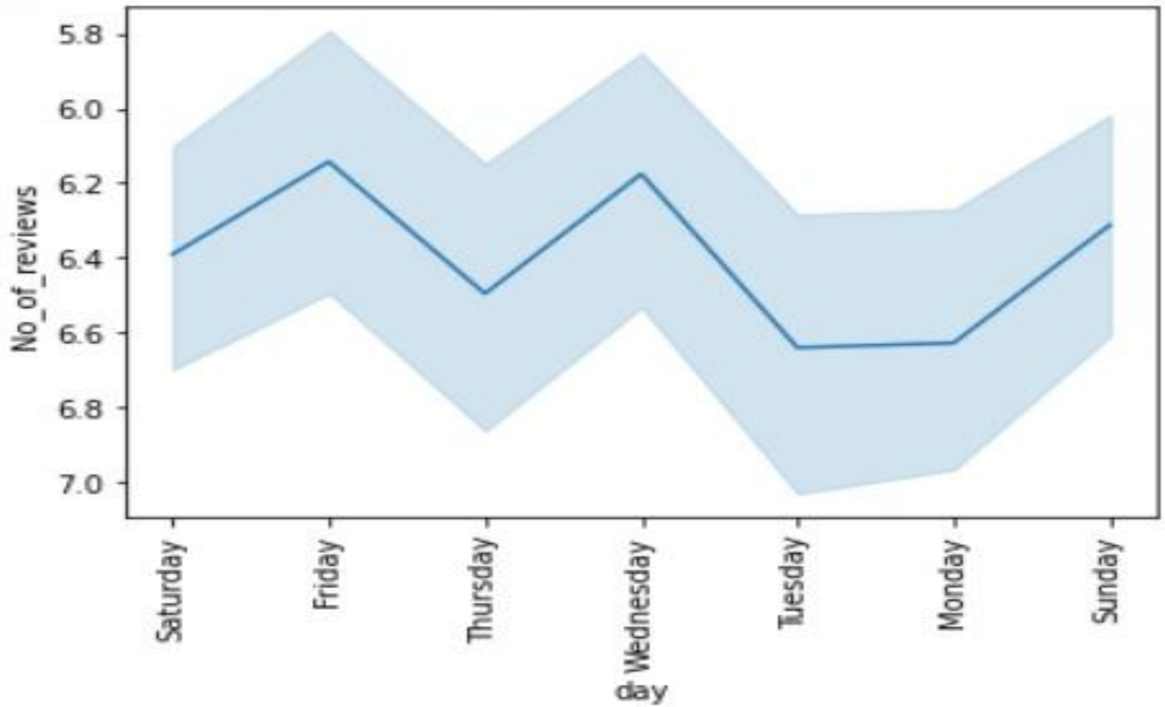
Italian,North Indian, Chinese, Continental,European, Mediterranean are most expensive cuisines

ANALYSIS OF RATING VARIABLE W.R.T PICTURES



People who are giving Rating more than 3 are oftenly posting pictures too

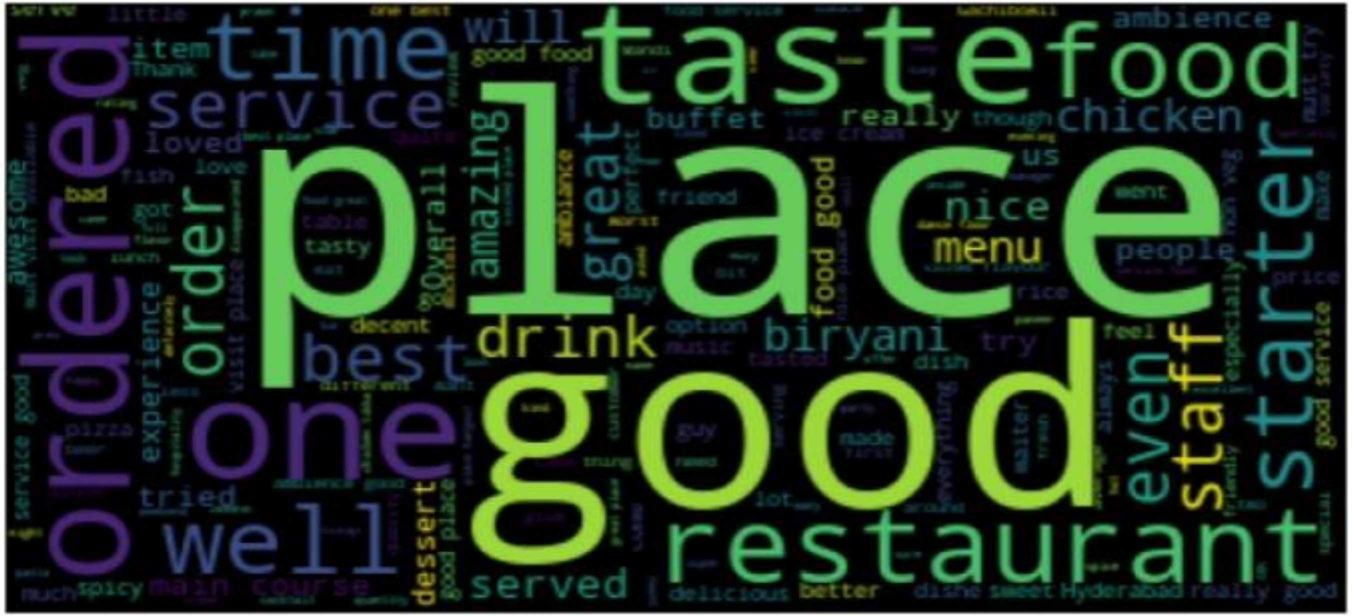
ANALYSIS OF DAY VARIABLE W.R.T NO_OF_REVIEWS



No_of _reviews received on Friday and Wednesday are more as compared to the other days

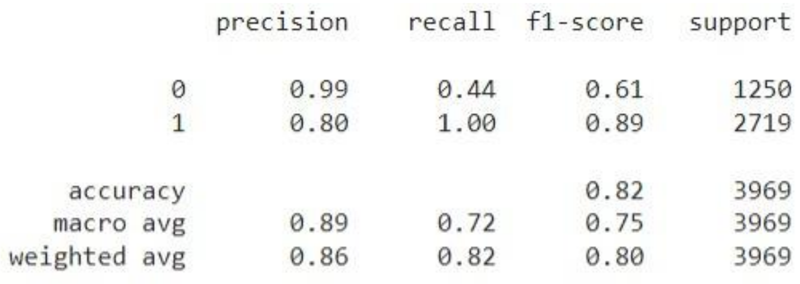
Most common words in reviews

```
Most common words in review
```



TFIDF Vectorizer

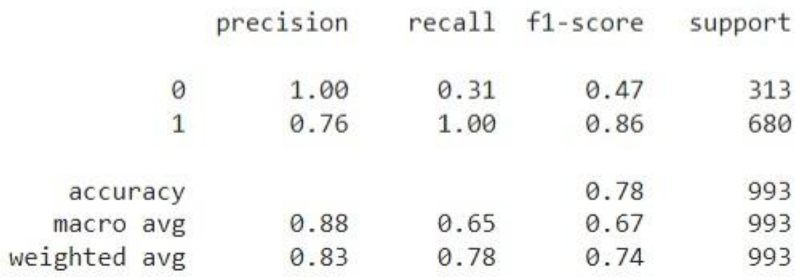
MultinomialNB



```
              precision    recall  f1-score   support

           0       0.99      0.44      0.61      1250
           1       0.80      1.00      0.89      2719

    accuracy                           0.82      3969
   macro avg       0.89      0.72      0.75      3969
weighted avg       0.86      0.82      0.80      3969
```

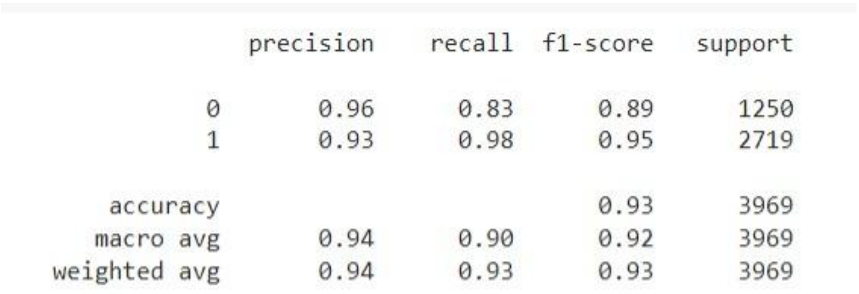


```
              precision    recall  f1-score   support

           0       1.00      0.31      0.47       313
           1       0.76      1.00      0.86       680

    accuracy                           0.78       993
   macro avg       0.88      0.65      0.67       993
weighted avg       0.83      0.78      0.74       993
```

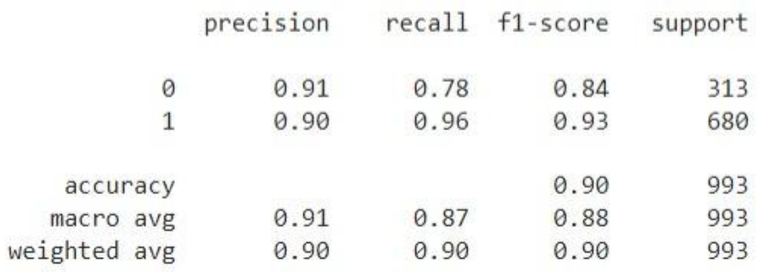
LogisticRegression



```
              precision    recall  f1-score   support

           0       0.96      0.83      0.89      1250
           1       0.93      0.98      0.95      2719

    accuracy                           0.93      3969
   macro avg       0.94      0.90      0.92      3969
weighted avg       0.94      0.93      0.93      3969
```



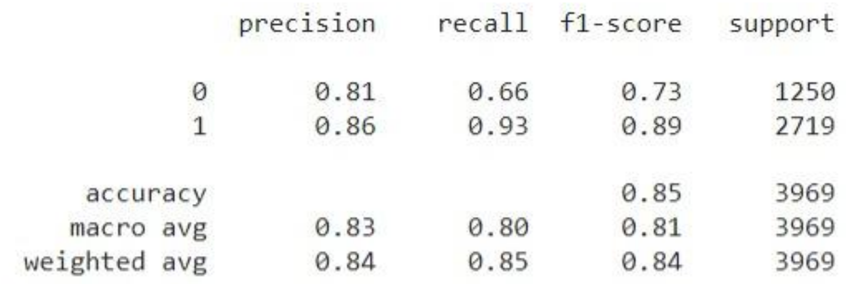
```
              precision    recall  f1-score   support

           0       0.91      0.78      0.84       313
           1       0.90      0.96      0.93       680

    accuracy                           0.90       993
   macro avg       0.91      0.87      0.88       993
weighted avg       0.90      0.90      0.90       993
```

TFIDF Vectorizer

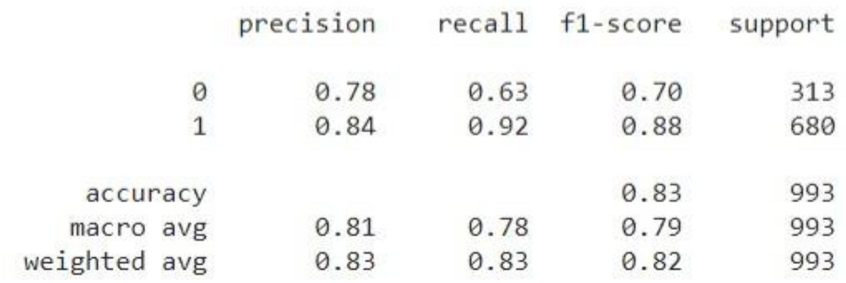
DecisionTreeClassifier



```
              precision    recall  f1-score   support

           0       0.81      0.66      0.73      1250
           1       0.86      0.93      0.89      2719

    accuracy                           0.85      3969
   macro avg       0.83      0.80      0.81      3969
weighted avg       0.84      0.85      0.84      3969
```

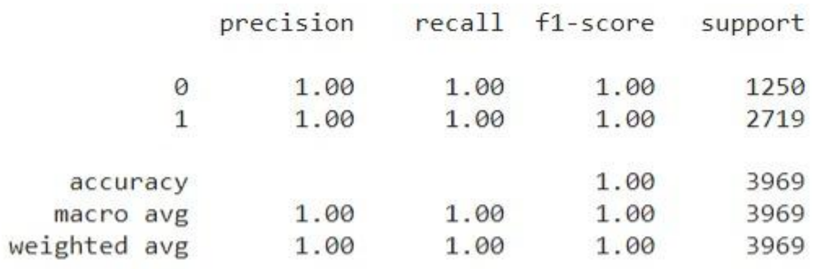


```
              precision    recall  f1-score   support

           0       0.78      0.63      0.70       313
           1       0.84      0.92      0.88       680

    accuracy                           0.83       993
   macro avg       0.81      0.78      0.79       993
weighted avg       0.83      0.83      0.82       993
```

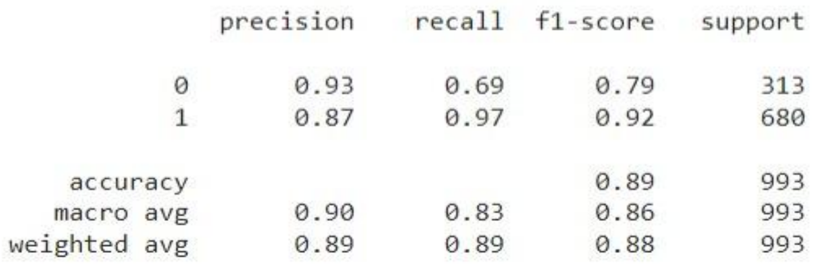
RandomForestClassifier



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1250
           1       1.00      1.00      1.00      2719

    accuracy                           1.00      3969
   macro avg       1.00      1.00      1.00      3969
weighted avg       1.00      1.00      1.00      3969
```



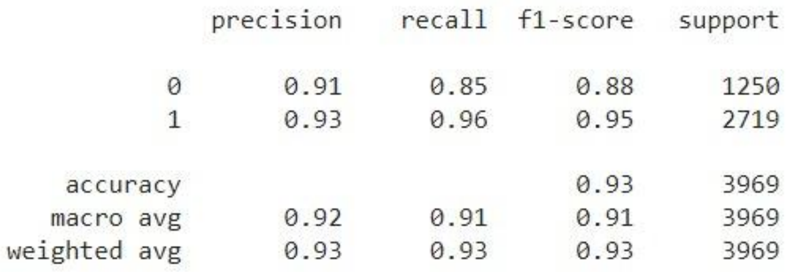
```
              precision    recall  f1-score   support

           0       0.93      0.69      0.79       313
           1       0.87      0.97      0.92       680

    accuracy                           0.89       993
   macro avg       0.90      0.83      0.86       993
weighted avg       0.89      0.89      0.88       993
```

Bag Of Words

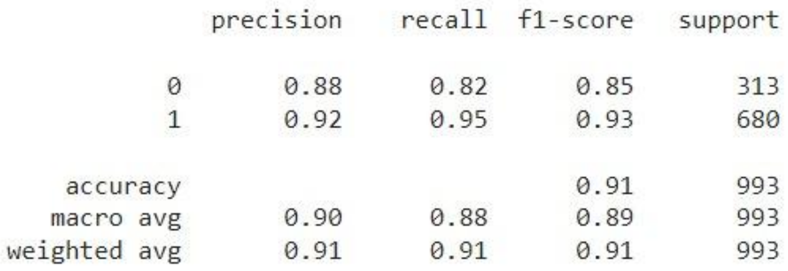
MultinomialNB



```
              precision    recall  f1-score   support

           0       0.91      0.85      0.88      1250
           1       0.93      0.96      0.95      2719

    accuracy                           0.93      3969
   macro avg       0.92      0.91      0.91      3969
weighted avg       0.93      0.93      0.93      3969
```

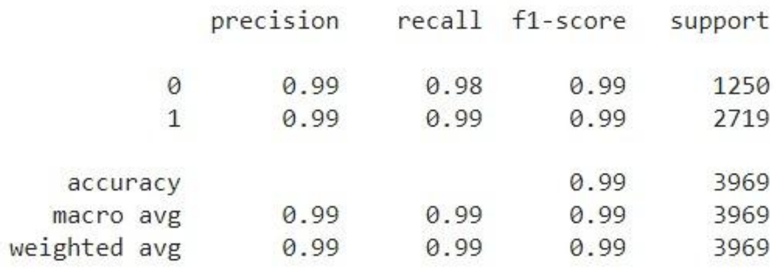


```
              precision    recall  f1-score   support

           0       0.88      0.82      0.85       313
           1       0.92      0.95      0.93       680

    accuracy                           0.91       993
   macro avg       0.90      0.88      0.89       993
weighted avg       0.91      0.91      0.91       993
```

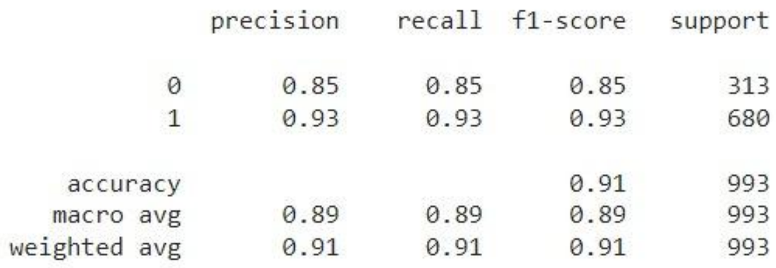
LogisticRegression



```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      1250
           1       0.99      0.99      0.99      2719

    accuracy                           0.99      3969
   macro avg       0.99      0.99      0.99      3969
weighted avg       0.99      0.99      0.99      3969
```



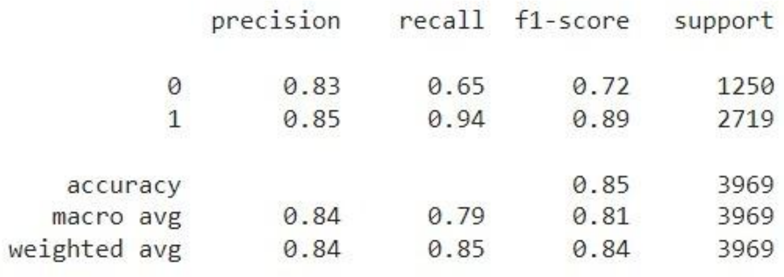
```
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       313
           1       0.93      0.93      0.93       680

    accuracy                           0.91       993
   macro avg       0.89      0.89      0.89       993
weighted avg       0.91      0.91      0.91       993
```

Bag Of Words

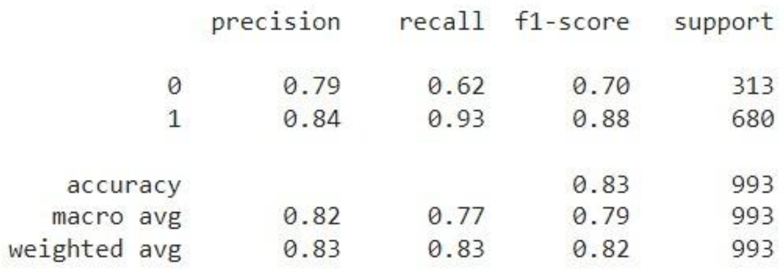
DecisionTreeClassifier



```
              precision    recall  f1-score   support

           0       0.83      0.65      0.72      1250
           1       0.85      0.94      0.89      2719

    accuracy                           0.85      3969
   macro avg       0.84      0.79      0.81      3969
weighted avg       0.84      0.85      0.84      3969
```

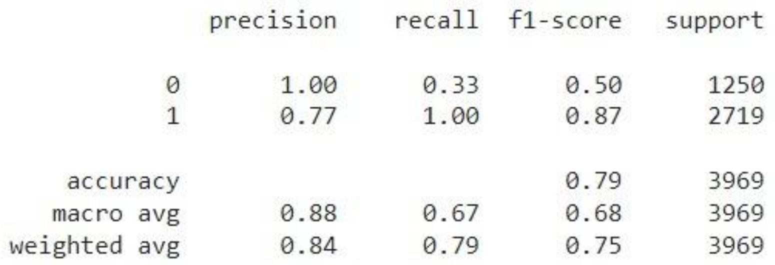


```
              precision    recall  f1-score   support

           0       0.79      0.62      0.70       313
           1       0.84      0.93      0.88       680

    accuracy                           0.83       993
   macro avg       0.82      0.77      0.79       993
weighted avg       0.83      0.83      0.82       993
```

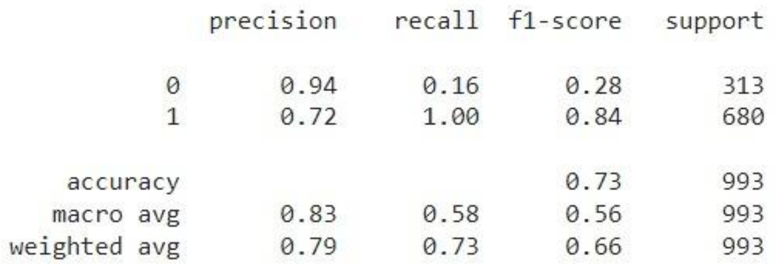
RandomForestClassifier



```
              precision    recall  f1-score   support

           0       1.00      0.33      0.50      1250
           1       0.77      1.00      0.87      2719

    accuracy                           0.79      3969
   macro avg       0.88      0.67      0.68      3969
weighted avg       0.84      0.79      0.75      3969
```



```
              precision    recall  f1-score   support

           0       0.94      0.16      0.28       313
           1       0.72      1.00      0.84       680

    accuracy                           0.73       993
   macro avg       0.83      0.58      0.56       993
weighted avg       0.79      0.73      0.66       993
```

Challenges

- Merging of Two datasets
- Handled row dataset with outliers and missing values.
- plotting of Graphs to analyse.
- Feature engineering
- Feature selection
- Find Best cluster
- Use of TF IDF And Word2vec vectorizer
- Optimising the model
- Calculation of Accuracy score.

Conclusion

- Per person estimated Cost of dining in 2017 was too high so it was expensive year
- Cost per person was high avg.750 RS in january and very low avg. 600RS in july
- collage-Hyatt Hyderabad Gachibowli' and 'Feast sheraton hyderabad hotel are most expensive hotel
- North Indian, Chinese, Continental,European, Mediterranean are most rated cuisines
- Monday and Tuesday was the most expensive days in week
- people prefered posting reviews on friday,wednesday & sunday we call it traffic days
- Most Cheapest Restaurants are Amul & Mohammedia Shawarma

Conclusion

- From elbow method we got 2 number of cluster is best among all.
- By using dendrogram we could found 2 as optimal number of cluster
- For 2 clusters silhouette_score is : 0.70

After applying Several Regression models such as MultinomialNB, Logistic Regression, DecisionTreeClassifier and Random forest Regression has yielded us Best Accuracy compared to all the other models which is of 99% for TFIDF Vectorizer

& for Bag of words we applied ,Logistic Regression, DecisionTreeClassifier and Random forest Regression and LogisticRegression gave us 98% accuracy for train dataset

References.

- Matplotlib Bars (w3schools.com)
- Seaborn (w3schools.com)
- Python Word Clouds Tutorial: How to Create a Word Cloud - DataCamp