Zomato Restaurant Clustering and Sentiment Analysis

Tito Varghese

Data Science Trainees,
Alma Better , Banglore

Abstract

Zomato is an Indian restaurant aggregator and food delivery start-up founded by Deepinder Goyal and Pankaj Chaddah in 2008. Zomato provides information, menus and user-reviews of restaurants, and also has food delivery options from partner restaurants in select cities.

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 vizualized 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.

This could help in clustering the restaurants into segments. Also the data

has valuable information around cuisine and costing which can be used in cost vs. benefit analysis

Data could be used for sentiment analysis. Also the metadata of reviewers can be used for identifying the critics in the industry.

Attribute

Zomato Restaurant names and Metadata

Use this dataset for clustering part

---

1. Name : Name of Restaurants
2. Links : URL Links of Restaurants
3. Cost : Per person estimated Cost of dining
4. Collection : Tagging of Restaurants w.r.t. Zomato categories
5. Cuisines : Cuisines served by Restaurants
6. Timings : Restaurant Timings

Zomato Restaurant reviews

Merge this dataset with Names and Matadata and then use for sentiment analysis part

---

1. Restaurant : Name of the Restaurant
2. Reviewer : Name of the Reviewer
3. Review : Review Text
4. Rating : Rating Provided by Reviewer
5. MetaData : Reviewer Metadata - No. of Reviews and followers
6. Time: Date and Time of Review
7. Pictures : No. of pictures posted with review

Introduction

This is our Unsupervised Capstone Project, hence we will be looking into multiple unsupervised models and supervised models for our sentiment analysis ,at the end try to come up with a best model. We are only focussing on all that algorithm which has been taught to us till now in our class. Logistic,XGboost,Random Forest,Decision Tree,LightGBM, LDA,NNMF and a few more clustering algorithm like K-Means,Hierarchical,DBSCAN and PCA, we have implemented in this capstone project.

Pipeline to be followed:-

- Dataset Inspection & Cleaning
- Feature Engineering
- Exploratory Data Analysis
- Text Preprocessing
- Topic Modelling
- Supervised ML Classification Models
- Model Summary
- ROC-AUC Plot
- Clustering Models
- Cluster Analysis
- Cost Benfit Analysis
- Conclusion

- Challenges

Dataset Inspection & Cleaning

Data cleaning is one of the important parts. We remove the unwanted observations, fix the structural errors, manage the unwanted outliers and handle the missing data.

We have loaded the dataset using the given csv files.In this project we had two dataset i.e restaurant and review dataset.We checked the general information about data.

Zomato Restaurant Datset Inspection and Cleaning

- We observe that the restaurant dataset and it contains 105 records and 6 features.
- The data types of all the features are object.
- Two features are having missing values i.e the collections and timings features.
- The collections feature has 54 null values and timings feature has 1 null value.
- Dropping off the nan values in Collection feature will not be a good idea because out of 105 rows if we drop 54 rows it may lead to loss of data information.Hence we will replace nan value in Collection feature with Tags Undefined
- The restaurant dataset does not contain any duplicate rows.
- The most common cuisines available in the most of the restaurant is the North Indian and Chinese. Hence we can say that there is a high demand for these cuisines from the customers end.
- The most frequent working hours of the restaurant is between 11am to 11pm.
- The billing amount of rupees 500 is the most frequent cost paid by the customers on their order.
- Changed the datatype of Cost to Numeric

Zomato Review Dataset Inspection and Cleaning

- The review dataset contains 1000 records and 7 features.
- The data type of all the features except Pictures are Object data type.
- The Pictures feature is an integer data type
- Apart from Restaurant and Pictures features,rest all other features consists of null values
- We have 36 duplicate rows in review dataset,hence we will drop these duplicate rows.
- We can see that rating value has one string value 'Like',which is acts as an anomaly.hence we will replace it to nan

- Converting the data type of Rating into float and Time into Date Time.
- We had 1581 null values in Followers feature,we have replaced null values with 0.

Feature Engineering

Buliding New Features

We have derived few new features from meta data and time features in review dataset.

Few new features derived are-

Reviewer

Followers

Date

Time

Month

Week

Year

```
#feature engineering on Time feature(adding new features)
zomato_reviews_df['Hour'] = pd.DatetimeIndex(zomato_reviews_df['Time']).hour
zomato_reviews_df['Week'] = pd.DatetimeIndex(zomato_reviews_df['Time']).week
zomato_reviews_df['Month'] = pd.DatetimeIndex(zomato_reviews_df['Time']).month
zomato_reviews_df['Year'] = pd.DatetimeIndex(zomato_reviews_df['Time']).year
```

```
#feature engineering on Metadata  feature(adding new features)
zomato_reviews_df['Reviews'],zomato_reviews_df['Followers']=zomato_reviews_df['Metadata'].str.
zomato_reviews_df['Reviews'] = pd.to_numeric(zomato_reviews_df['Reviews'].str.split(' ').str[0
zomato_reviews_df['Followers'] = pd.to_numeric(zomato_reviews_df['Followers'].str.split(' ').s
```

```
#dropping the metadata after feature engineering
zomato_reviews_df = zomato_reviews_df.drop(['Metadata'], axis =1)
```

We have created new feature Sentiment using a function rating bin in our review dataset while performing sentiment Analysis

```
#used a function to convert the rating feature into binary values(0 and 1)
def rating_bin(rating):
  if rating >=3.5:
    return 1
    # positive sentiment
  else:
    return 0
    # negative sentiment
```

```
#new feature sentiment created using the rating_bin function
df_sentiment['Sentiment'] = df_sentiment['Rating'].apply(lambda x:rating_bin(x))
df_sentiment.head()
```

	Rating	Reviews	Sentiment
0	5.0	ambience good food good saturday lunch cost ef...	1
1	5.0	ambience good pleasant evening service prompt ...	1
2	5.0	try great food great ambience thnx service pra...	1
3	5.0	soumen das arun great guy behavior sincerety g...	1
4	5.0	food goodwe order kodi drumstick basket mutton...	1

We have done Binning on Cuisine feature before performing clustering in restaurant dataset inorder to categorize all cuisines available into different bins

```
# Binning all the cuisines into their respective cuisine categories
cuisine_category_list=[]
for i in cluster_df['Cuisines']:
  if (i=='Hyderabadi')|(i=='North Indian')|(i=='Modern Indian')|(i=='Biryani')|(i=='Mughlai')|(i=='South Indian')|(i=='Andhra')|(i=='North Eastern')|(i=='Seafood')|(i=='Goan'):
    cuisine_category_list.append('Indian Food')
  if (i=='Kebab')|(i=='BBQ')|(i=='Mughlai')|(i=='Salad')|(i=='Healthy Food'):
    cuisine_category_list.append('Starter Food')
  if (i=='Sushi')|(i=='Thai')|(i=='Indonesian')|(i=='Malaysian')|(i=='Chinese')| (i=='Asian')|(i=='Japanese'):
    cuisine_category_list.append('South-East Asian Food')
  if (i=='Lebanese')|(i=='Italian')|(i=='European')|(i=='Mediterranean')|(i=='American')|(i=='Arabian')|(i=='Mexican')|(i=='Spanish')|(i=='Continental'):
    cuisine_category_list.append('Continental food')
  if (i=='Momos')|(i=='Street Food')|(i=='Pizza')|(i=='Wraps')|(i=='Burger')|(i=='Fast Food')|(i=='Finger Food'):
    cuisine_category_list.append('Fast Food')
  if (i=='Bakery')|(i=='Beverages')|(i=='Desserts')|(i=='Juices')|(i=='Ice Cream')|(i=='Mithai')|(i=='Cafe'):
    cuisine_category_list.append('Beverges n Desserts')
```

```
] #Storing the cuisine category list in a dataframe
cuisine_category_df=pd.DataFrame(cuisine_category_list)
cuisine_category_df.columns=['cuisine']
cuisine_category_df
```

Exploratory Data Analysis

Exploratory data analysis is a crucial part of data analysis. It involves exploring and analyzing the dataset given to find patterns, trends and conclusions to make better decisions related to the data, often using statistical graphics and other data visualization tools to summarize the results. Python

libraries like pandas are used to explore the data and matplotlib and seaborn to visualize it.

Some important aspects to include in the project are as follows:

Best restaurants in the city on the basis of Ratings

The Most Popular Cuisines

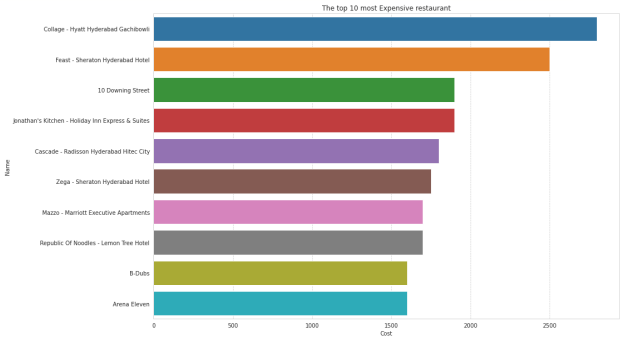
The Most Common Tags given to the Restaurants

The most affordable and the expensive Restaurants

Cuisines and their Costing

Active Reviewer

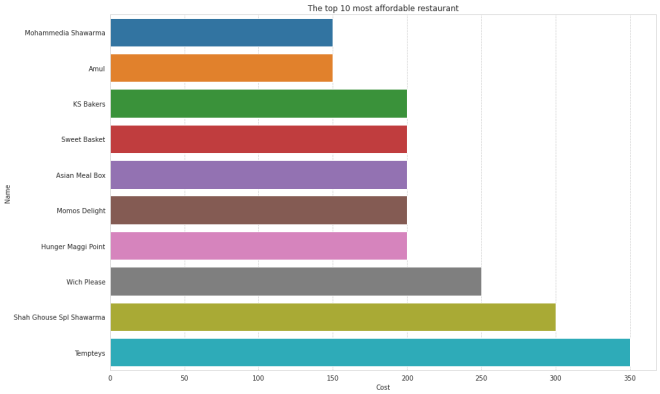
The top 10 most expensive restaurants



From the above graph,we can conclude that the most expensive restaurant in the city is Hyatt Hyderabad.

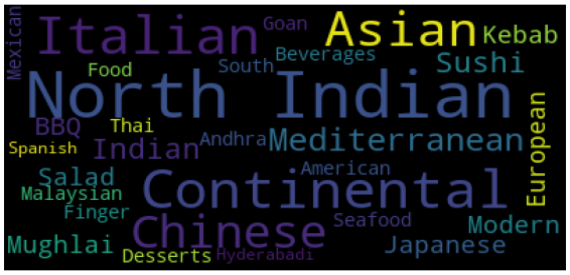
Wordcloud of 30 expensive Restaurants

The top 10 most affordable restaurants

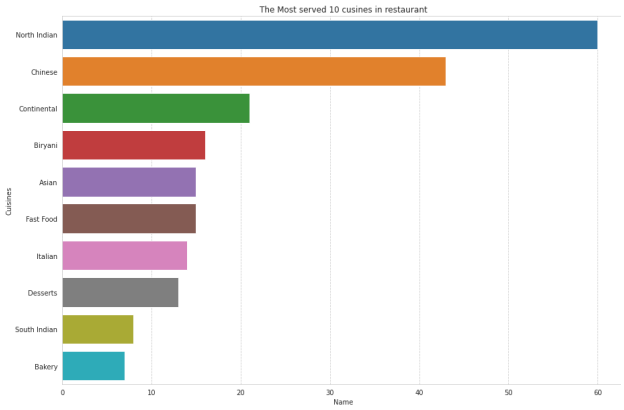


From the above graph,we can conclude that the most affordable restaurant in the city is Mohammedia Shawarma restaurant.

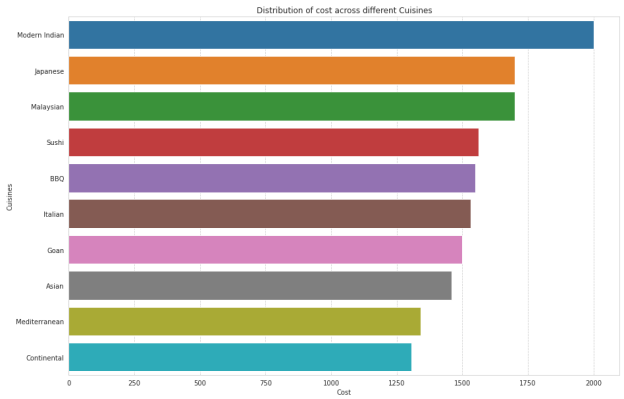
Word Cloud of Top Cuisines



The top 10 cuisines highly on demand in restaurants

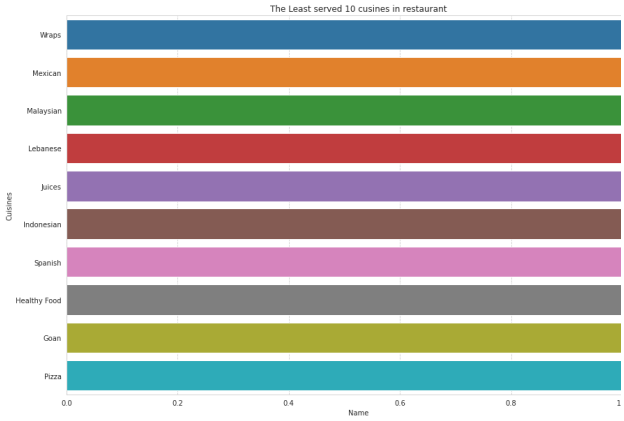


The above figure tells us that the most popular Cuisines are North Indian and Chinese.



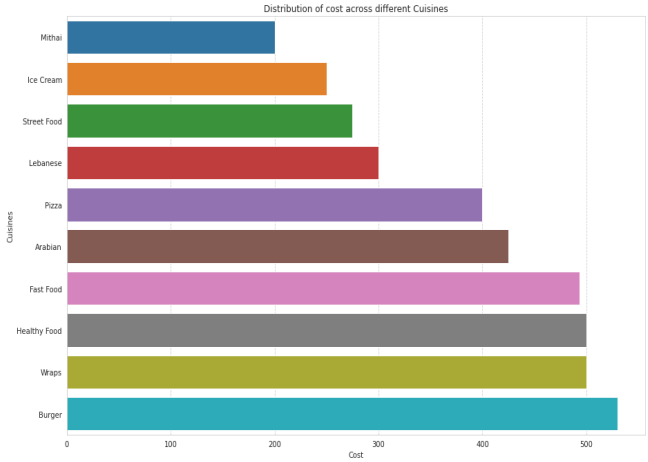
The above figure shows that the most expensive cuisine is the Moder India Cuisine which cost around 2000 inr.

The least on demand 10 cuisines in restaurants



The above figure shows the least demand cuisine is Wraps

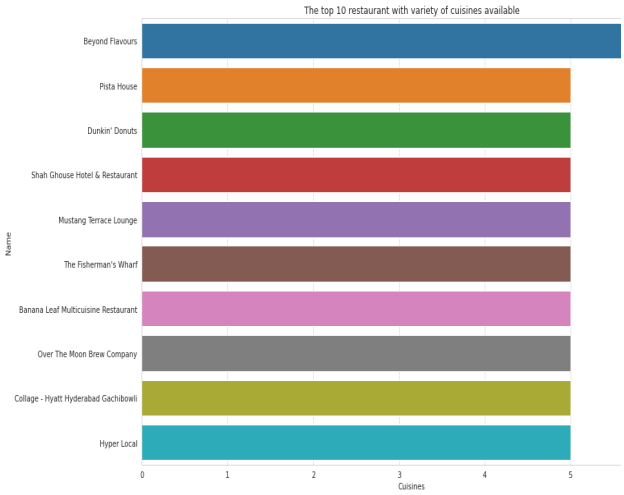
The top 10 most affordable cuisines



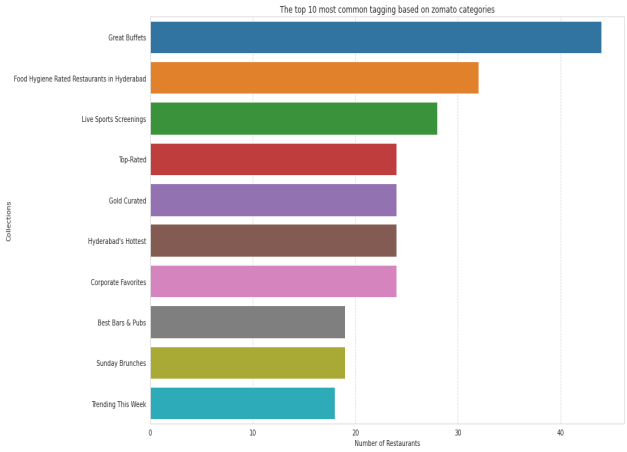
The above figure shows that the most affordable cuisine is the Mithai which cost around 200 inr.

The top 10 most expensive cuisines

The Top 10 restaurant with variety of cuisines available

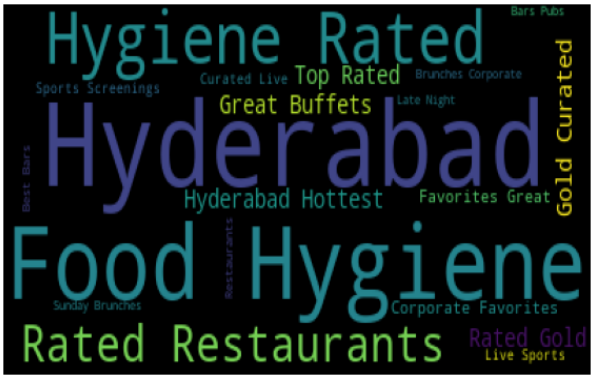


The above figure shows that the Beyond Flavour restaurant with highest no. of Cuisines Variety available restaurant.

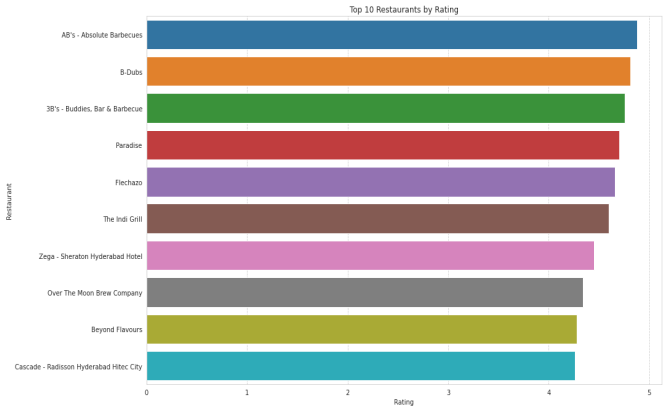


The above figure shows that the most common tag used in restaurant is Great Buffet.

WordCloud Top 30 Restaurant Tags



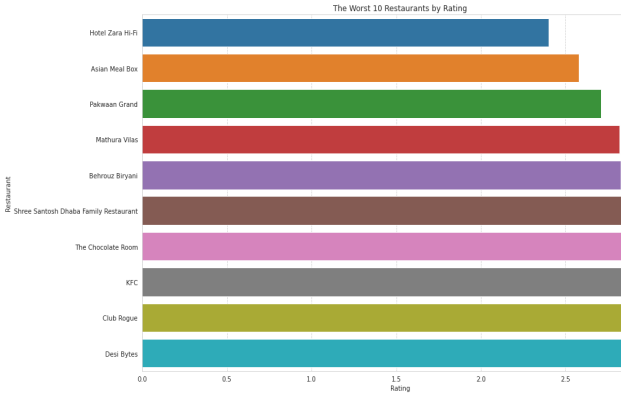
The Top 10 restaurants based on the ratings



The above figure shows that the AB's Absoulte Barbecue as the best restaurant on the basis of ratings.

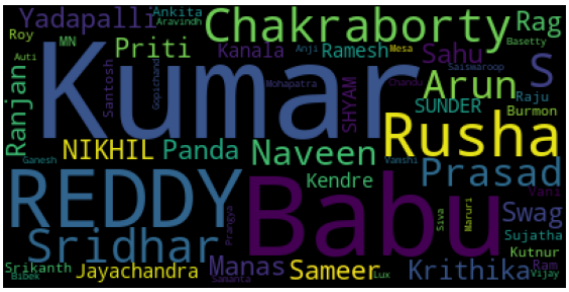
The top 10 most common tagging based on zomato categories

The Worst 10 restaurants based on the ratings

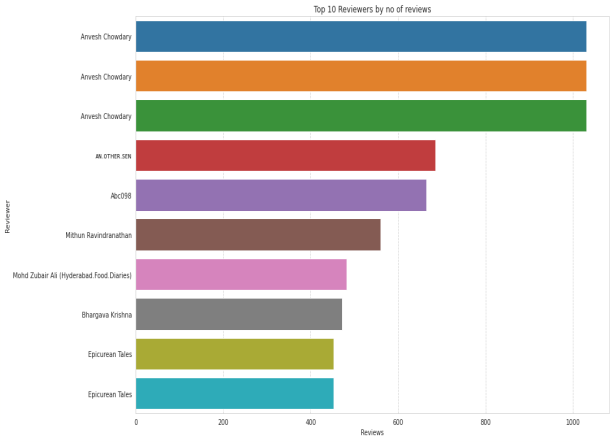


The above figure shows that the Hotel Zara HiFi as the worst restaurant on the basis of ratings.

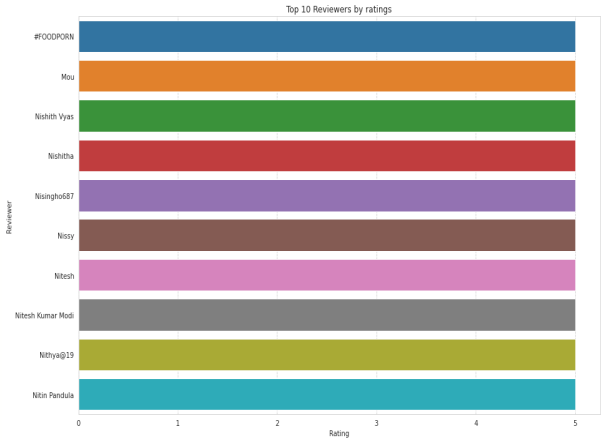
WordCloud Most Rated Reviewer



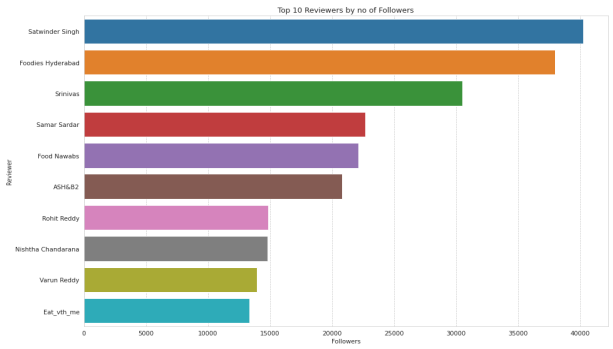
The Top 10 Reviewers based on number of reviews



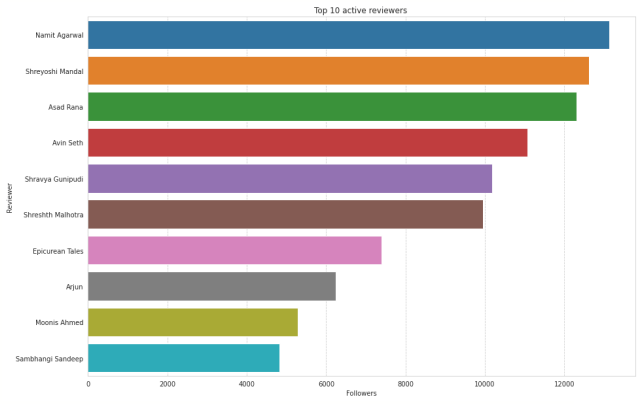
The Top 10 Reviewers based on ratings



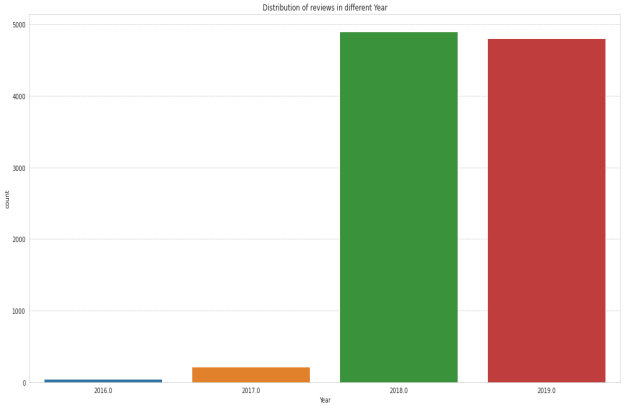
The Top 10 Reviewers based on rating mean , reviews and followers sum



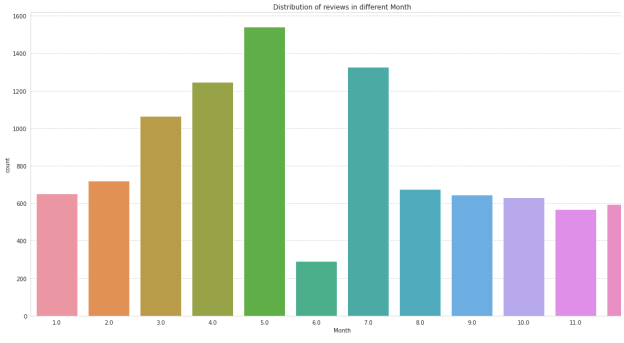
The Top 10 Active Reviewers



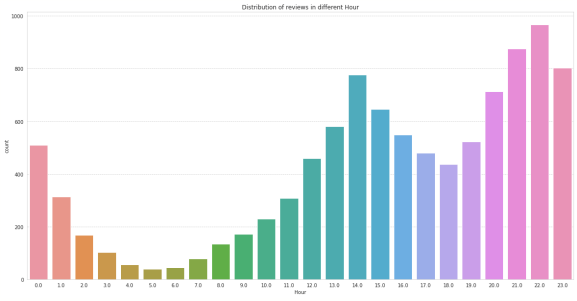
Distribution of reviews in different year



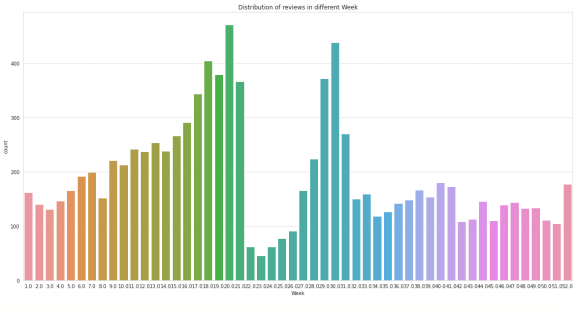
Distribution of reviews in different month



Distribution of reviews in different hour



Distribution of reviews in different week



EDA Findings:

The collage-Hyatt Hyderbad Gachibowli is the most expensive restaurant available and the most affordable restaurant for the customers are the Amul and Mohammedia Shawarma.

The North Indian Cuisine is one of the cuisine highly in demand followed by Chinese Cuisine and the least in demand cuisines are Wraps and Mexican.

The most expensive Cuisine is the Modern India cuisine which cost around 2000 rupees and the least expensive item available at a cost of 200 rupees is the Mithai .

The Beyond Flavours Restaurant is the only restaurant with six different variety of cuisines available.

Great Buffet is one of the most common Tags given to the zomato restaurants with nearly more than 40 restaurants.

The Restaurant with the highest rating of nearly 4.8 and good reviews is the AB's Absolute Barbecues. On the contrary, the restaurant with worst reviews and rating is the Hotel Zara Hifi with a rating less than 2.5.

Anvesh Choudhary is one of the top reviewer based on number of reviews given and Foodporn is the reviewer who has given highest rating.

Namit Aggarwal is one of the active reviewer based on number of followers and ratings provided. He has an followers more than 12000 ,so we can consider his reviews with utmost importance for sure.

Distribution of reviews based on different months shows that there is a progressive shift in number of reviews from the month of Jan to May ,thereafter a sudden dip in the number of reviews in the month of June.It may be possibly happened due to some internal technical glicth in the zomato site because of which they may not able to collect reviews from the customers end.

Distribution of reviews based on different hours in a day shows us that the most of the reviews been given during afternoon time between 12-16 hrs,hence we can say mostly during lunch hours we can see high number of demand from customer end.

We are having the distribution of reviews from the year 2016 to 2019 in our dataset and the year 2018 has gained the highest number of reviews followed by the year 2019.

Text Preprocessing

To prepare the text data for the model building we perform text preprocessing. It is the very first step of NLP projects. Some of the preprocessing steps are:

Removing punctuations like . , ! $( ) * % @

Removing URLs

Removing Stop words

Lower casing

Tokenization

Lemmatization

Natural Language Processing (NLP) is a branch of Data Science which deals with Text data. Apart from numerical data, Text data is available to a great extent which is used to analyze and solve business problems. But before using the data for analysis or prediction, processing the data is important.

To prepare the text data for the model building we perform text preprocessing. It is the very first step of NLP projects. Some of the preprocessing steps are:

- Removing punctuations like . , ! $( ) * % @
- Removing URLs
- Removing Stop words
- Lower casing
- Tokenization
- Stemming

- Lemmatization

We need to use the required steps based on our dataset.

Steps to clean the data

Punctuation Removal:

In this step, all the punctuations from the text are removed. string library of Python contains some pre-defined list of punctuations such as '!"#$%&'()*+,-./:;?@[\]^_`{|}~'

Lowering the text:

It is one of the most common text preprocessing Python steps where the text is converted into the same case preferably lower case. But it is not necessary to do this step every time you are working on an NLP problem as for some problems lower casing can lead to loss of information.

Tokenization:

In this step, the text is split into smaller units. We can use either sentence tokenization or word tokenization based on our problem statement.

Stop word removal:

Stopwords are the commonly used words and are removed from the text as they do not add any value to the analysis. These words carry less or no meaning.

NLTK library consists of a list of words that are considered stopwords for the English language. Some of them are : [i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now, d, ll, m, o, re, ve, y, ain, aren't, could, couldn't, didn't, didn't]

But it is not necessary to use the provided list as stopwords as they should be chosen wisely based on the project.

Lemmatization:

It stems the word but makes sure that it does not lose its meaning. Lemmatization has a pre-defined

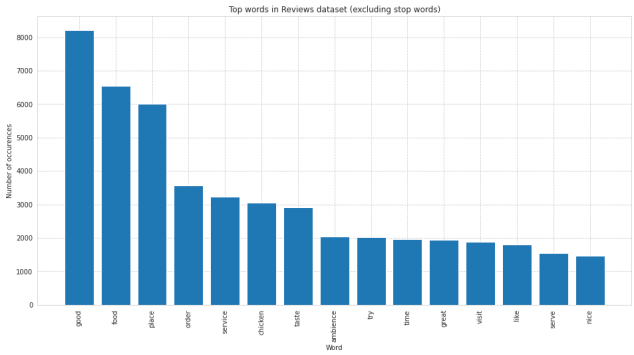
dictionary that stores the context of words and checks the word in the dictionary while diminishing.

Topic Modelling

Topic modelling is recognizing the words from the topics present in the document or the corpus of data. This is useful because extracting the words from a document takes more time and is much more complex than extracting them from topics present in the document. For example, there are 1000 documents and 500 words in each document. So to process this it requires 500*1000 = 500000 threads. So when you divide the document containing certain topics then if there are 5 topics present in it, the processing is just 5*500 words = 2500 threads.

This looks simple than processing the entire document and this is how topic modelling has come up to solve the problem and also visualizing things better.

First we develop a list of the top words used by reviewers in reviews textual data, giving us a glimpse into the core vocabulary of the source data. Stop words are omitted here to avoid any trivial conjunctions, prepositions, etc.



First, let's get familiar with NLP so that Topic modelling gets easier to unlock

Topic modelling is done using LDA(Latent Dirichlet Allocation). Topic modelling refers to the task of identifying topics that best describes a set of documents. These topics will only emerge during the topic modelling process (therefore called latent). And

one popular topic modelling technique is known as Latent Dirichlet Allocation (LDA).

Topic modelling is an unsupervised approach of recognizing or extracting the topics by detecting the patterns like clustering algorithms which divides the data into different parts. The same happens in Topic modelling in which we get to know the different topics in the document. This is done by extracting the patterns of word clusters and frequencies of words in the document.

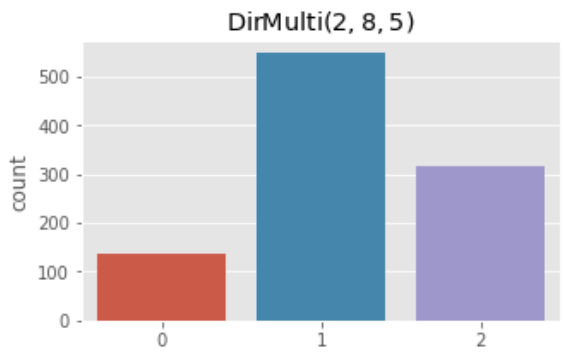
So based on this it divides the document into different topics. As this doesn't have any outputs through which it can do this task hence it is an unsupervised learning method. This type of modelling is very much useful when there are many documents present and when we want to get to know what type of information is present in it. This takes a lot of time when done manually and this can be done easily in very little time using Topic modelling.

What is LDA and how is it different from others?

Latent Dirichlet Allocation:

In LDA, latent indicates the hidden topics present in the data then Dirichlet is a form of distribution. Dirichlet distribution is different from the normal distribution. When ML algorithms are to be applied the data has to be normally distributed or follows Gaussian distribution. The normal distribution represents the data in real numbers format whereas Dirichlet distribution represents the data such that the plotted data sums up to 1. It can also be said as Dirichlet distribution is a probability distribution that is sampling over a probability simplex instead of sampling from the space of real numbers as in Normal distribution.

For example,

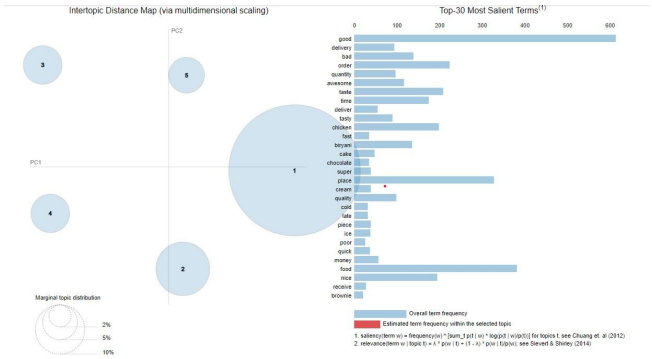


Normal distribution tells us how the data deviates towards the mean and will differ according to the variance present in the data. When the variance is high then the values in the data would be both smaller and larger than the mean and can form skewed distributions. If the variance is small then samples will be close to the mean and if the variance is zero it would be exactly at the mean.

Now when the LDA is clear than now the Topic Modelling in LDA? Yes, it would be, let's look into this one.

Now when topic modelling is to get the different topics present in the document. LDA comes to as a savior for doing this task easily instead of performing many things to achieve it. As LDA brings the words in the topics with their distribution using Dirichlet distribution. Hence the name Latent Dirichlet Allocation. The words assigned(or allocated) to the topic with their distribution using Dirichlet distribution.

LDA has three important hyperparameters. They are 'alpha' which represents document-topic density factor, 'beta' which represents word density in a topic, 'k' or the number of components representing the number of topics you want the document to be clustered or divided into parts.



The top 15 words in each topics in LDA

Non-Negative Matrix Factorization

Non-negative Matrix Factorization (NNMF) or the positive matrix analysis is another NLP technique for topic modeling. NNMF differs from LDA because it depends on creating two matrices from random numbers. The first matrix represents the relationship between words and topic while the second matrix represents the relationship between the topic and documents that forms the mathematical basis for categorizing texts as happened in LDA.NNMF is faster and more accurate than LDA because NNMF selects random correlation values between words and topics and training is run based in words exist or not which enable for adjusting weights as the training repeated. NNMF is more favorable for its dimension reduction

The top 15 words in each topics in NNMF

So we can see above, we got words in the first topic like price, taste,spicy, quality and service which suggests that this topic is about the food attributes while in fourth topic for example, the most valuable words are ‘delivery, fast, guy, excellent,'late’ and that suggest that topic 4 is about delivery service.

Supervised ML Classification Model Formulation

Data Preparation for Model Formulation

Data Preparation is the process of cleaning and transforming raw data to make predictions accurately through using ML algorithms. Although data preparation is considered the most complicated stage in ML, it reduces process complexity later in real-time projects.

Everyone must explore a few essential tasks when working with data in the data preparation step. These are as follows:

Data cleaning: This task includes the identification of errors and making corrections or improvements to those errors.

Feature Selection: We need to identify the most important or relevant input data variables for the model.

Data Transforms: Data transformation involves converting raw data into a well-suitable format for the model.

Feature Engineering: Feature engineering involves deriving new variables from the available dataset.

Dimensionality Reduction: The dimensionality reduction process involves converting higher dimensions into lower dimension features without changing the information.

Each machine learning project requires a specific data format. To do so, datasets need to be prepared well before applying it to the projects. Sometimes, data in data sets have missing or incomplete information, which leads to less accurate or incorrect predictions. Further, sometimes data sets are clean but not adequately shaped, such as aggregated or pivoted, and some have less business context. Hence, after collecting data from various data sources, data preparation needs to transform raw data. Below are a few significant advantages of data preparation in machine learning as follows:

- It helps to provide reliable prediction outcomes in various analytics operations.
- It helps identify data issues or errors and significantly reduces the chances of errors.
- It increases decision-making capability.
- It reduces overall project cost (data management and analytic cost).
- It helps to remove duplicate content to make it worthwhile for different applications.
- It increases model performance.

Our approach in this project will be to to take into consideration Rating and Reviews features for our Sentimentl Analysis

We have Reviews feature as our Indepensent feature and Sentiment feature as our dependant variable which is a binary class(0 negative and 1 for positive)

The next step will be to split the independent and dependent variable in the ratio of 75:25 to training and test data.

The next step will be to create tokens for text data(review feature).We us Tfidvectorizer to create the tokens.

```
# creating tokens for text data
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words='english',min_df=0.01,max_df=0.95)
X_train= vectorizer.fit_transform(X_train)
X_test=vectorizer.transform(X_test)
```

After splitting and tokenization the text data, we build a function to train the model using multiple supervised learning algorithms and classify the different models on the basis of their test accuracy.

For ML models to give reasonable results, we not only need to feed in large quantities of data but also have to ensure the quality of data.

Though making sense out of raw data is an art in itself and requires good feature engineering skills and domain knowledge (in special cases), the quality data is of no use until it is properly used. The major problem which ML/DL practitioners face is how to divide the data for training and testing. Though it seems like a simple problem at first, its complexity can be gauged only by diving deep into it.

Poor training and testing sets can lead to unpredictable effects on the output of the model. It may lead to overfitting or underfitting of the data and our model may end up giving biased results.

The data should ideally be divided into 3 sets – namely, train, test, and holdout cross-validation set.

Train Set:

The train set would contain the data which will be fed into the model. In simple terms, our model would learn from this data. For instance, a Regression model would use the examples in this data to find gradients in order to reduce the cost function. Then these gradients will be used to reduce the cost and predict data effectively.

Cross validation Set:

The development set is used to validate the trained model. This is the most important setting as it will form the basis of our model evaluation. If the difference between error on the training set and error on the dev set is huge, it means the model as high variance and hence, a case of over-fitting.

Test Set:

The test set contains the data on which we test the trained and validated model. It tells us how efficient our overall model is and how likely is it going to predict something which does not make sense. There are a plethora of evaluation metrics (like precision, recall, accuracy, etc.) which can be used to measure the performance of our model.

As we know most of the supervised and unsupervised learning methods make decisions according to the data sets applied to them and often the algorithms calculate the distance between the data points to make better inferences out of the data.

In the machine learning algorithms if the values of the features are closer to each other there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower.

So if the data in any conditions has data points far from each other, scaling is a

technique to make them closer to each other or in simpler words, we can say that the scaling is used for making data points generalized so that the distance between them will be lower.

As we know, most of the machine learning models learn from the data by the time the learning model maps the data points from input to output. And the distribution of the data points can be different for every feature of the data. Larger differences between the data points of input variables increase the uncertainty in the results of the model.

The machine learning models provide weights to the input variables according to their data points and inferences for output. In that case, if the difference between the data points is so high, the model will need to provide the larger weight to the points and in final results, the model with a large weight value is often unstable. This means the model can produce poor results or can perform poorly during learning.

We have used a function to compare different models based on the test accuracy of different models.We have following algorithm in our project to build the model

Logistic Regression, Decision Tree,LightGBM, XGBoost and Random Forest.

HyperParameter Tuning

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.

There is another kind of parameter, known as **Hyperparameters**, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two best strategies for Hyperparameter tuning are:

GridSearchCV

In GridSearchCV approach, the machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for the best set of hyperparameters from a grid of hyperparameters values.

In this project,we have used GridSearchCv and RandomisedCV for finding the best parameters for our classification model algorithm like RandomForest , KNN,SVM and Xgboost model to find the best parameter inorder to get a optimal and best model.

Logistic Regression

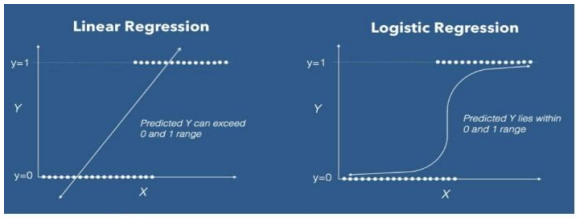
Logistic Regression is a “Supervised machine learning” algorithm that can be used to model the probability of a certain class or event. It is used when the data is linearly separable and the outcome is binary or dichotomous in nature.

That means Logistic regression is usually used for Binary classification problems.

Binary Classification refers to predicting the output variable that is discrete in two classes.

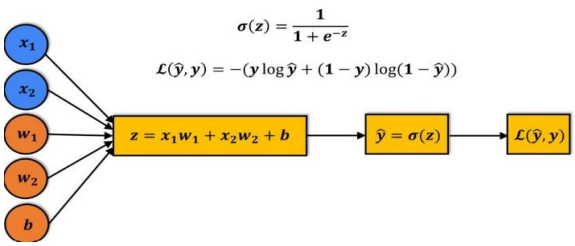
A few examples of Binary classification are Yes/No, Pass/Fail, Win/Lose, Cancerous/Non-cancerous, etc.

A linear equation (z) is given to a sigmoidal activation function (σ) to predict the output (ŷ).



Source

The image that depicts the working of the Logistic regression model



To evaluate the performance of the model, we calculate the loss. The most commonly used loss function is the mean squared error.

But in logistic regression, as the output is a probability value between 0 or 1, mean squared error wouldn't be the right choice. So, instead, we use the cross-entropy loss function.

The cross-entropy loss function is used to measure the performance of a classification model whose output is a probability value.

Inorder to optimize our logistic regression model we have used hyperparameter tuning using GridSearchCV with a cross validation value of 3.

The parameters we have selected in logistic regression model for tuning were penalty,C_value and class_iter.

The best parameters found out to be : {'C': 0.1, 'max_iter': 1000, 'penalty': 'l2'}

The accuracy ,precision,recall ,f1score and ROC score with default parameter and after hyperparameter tuning and the confusion and classification report after tuning as follows-

```
test score
**************************************************
       The accuracy is  0.8308557653676175
       The precision is  0.8133116883116883
       The recall is  0.9518682710576314
       The f1 is  0.8771520280128392
       the auc  is  0.7863736959683761

                              classification report
 **************************************************
              precision    recall  f1-score   support

           0       0.88      0.62      0.73       910
           1       0.81      0.95      0.88      1579

    accuracy                           0.83      2489
   macro avg       0.85      0.79      0.80      2489
weighted avg       0.84      0.83      0.82      2489
```

Decision Tree

Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

Before learning more about decision trees let's get familiar with some of the terminologies.

Root Nodes – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

Decision Nodes – the nodes we get after splitting the root nodes are called Decision Node

Leaf Nodes – the nodes where further splitting is not possible are called leaf nodes or terminal nodes

Sub-tree – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.

Pruning – is nothing but cutting down some nodes to stop overfitting.

The parameters we have selected in decision tree model for tuning were max_depth,criterion and max_leaf nodes.

The best parameters found out to be :

max_depth=10,max_leaf_nodes=50,criterion='entropy'

The accuracy ,precision,recall ,f1score and ROC score with default parameter and after hyperparameter tuning and the confusion and classification report after tuning as follows-

```
test score
**************************************************
        The accuracy is  0.7661711530735235
        The precision is  0.8375084631008801
        The recall is  0.7834072197593414
        The f1 is  0.8095549738219895
        the auc  is  0.759835478011539

                                classification report
**************************************************
              precision    recall  f1-score   support

           0       0.66      0.74      0.70       910
           1       0.84      0.78      0.81      1579

    accuracy                           0.77      2489
   macro avg       0.75      0.76      0.75      2489
weighted avg       0.77      0.77      0.77      2489
```

XGBoost Classifier

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

XGBoost (Extreme Gradient Boosting) belongs to a family of boosting algorithms and uses the gradient boosting (GBM) framework at its core. It is an optimized distributed gradient boosting library.

Inorder to optimize our XGBoost model we have used hyperparameter tuning using GridSearchCV with a cross validation value of 3.

The parameters we have selected in svm model for tuning were max_depth,n_estimator, and we have used criterion

The best parameters found out to be : {'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 150}

The accuracy ,precision,recall ,f1score and ROC score with the best parameter and the confusion and classification report after tuning as follows-

```
test score
**************************************************
       The accuracy is  0.8461229409401366
       The precision is  0.8576555023923444
       The recall is  0.9081697276757441
       The f1 is  0.8821900953552753
       the auc  is  0.8233156330686413

                              classification report
**************************************************
              precision    recall  f1-score   support

           0       0.82      0.74      0.78       910
           1       0.86      0.91      0.88      1579

    accuracy                           0.85      2489
   macro avg       0.84      0.82      0.83      2489
weighted avg       0.84      0.85      0.84      2489
```

Random Forest

Random forest is a technique used in modeling predictions and behavior analysis and is built on decision trees. It contains many decision trees representing a distinct instance of the classification of data input into the random forest. The random forest technique considers the instances individually, taking the one with the

majority of votes as the selected prediction.

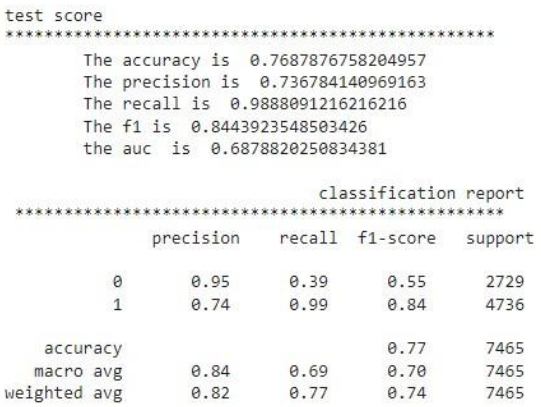
The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees.Using RandomForest Ensemble technique to predict the trip_duration after applying hyperparameter tuning with help of RandomizedSearchcv (cross-validation technique).

Inorder to optimize our Random Forest model we have used hyperparameter tuning using GridSearchCV with a cross validation value of 3.

The parameters we have selected in random forest model for tuning were max_depth,max_features,min_samples _leaf,min_samples_split,n_estimators and we have used gini as the criterion.

The best parameters found out to be : {'criterion': 'entropy', 'max_depth': 7, 'n_estimators': 125}

The accuracy ,precision,recall ,f1score and ROC score with the best parameter and the confusion and classification report after tuning as follows-



```
test score
**************************************************
       The accuracy is  0.7687876758204957
       The precision is  0.736784140969163
       The recall is  0.9888091216216216
       The f1 is  0.8443923548503426
       the auc  is  0.6878820250834381

                                 classification report
 **************************************************
              precision    recall  f1-score   support

           0       0.95      0.39      0.55      2729
           1       0.74      0.99      0.84      4736

    accuracy                           0.77      7465
   macro avg       0.84      0.69      0.70      7465
weighted avg       0.82      0.77      0.74      7465
```

LightGBM

The main features of the LGBM model are as follows :

- Higher accuracy and a faster training speed.
- Low memory utilization
- Comparatively better accuracy than other boosting algorithms and handles overfitting much better while working with smaller datasets.

- Parallel Learning support.
- Compatible with both small and large datasets

With the above-mentioned features and advantages of LGBM, it has become the default algorithm for machine learning competitions when someone is working with a tabular kind of data regarding both regression and classification problems.

A Gradient Boosting Decision tree or a GBDT is a very popular machine learning algorithm that has effective implementations like XGBoost and many optimization techniques are actually adopted from this algorithm. The efficiency and scalability of the model are not quite up to the mark when there are more features in the data. For this specific behavior, the major reason is that each feature should scan all the various data instances to make an estimate of all the possible split points which is very time-consuming and tedious.

To solve this problem, The LGBM or Light Gradient Boosting Model is used. It uses two types of techniques which are gradient Based on side sampling or GOSS and Exclusive Feature bundling or EFB. So GOSS will actually exclude the significant portion of the data part which have small gradients and only use the remaining data to estimate the overall information gain. The data instances which have large gradients actually play a greater role for computation on information gain. GOSS can get accurate results with a significant information gain despite using a smaller dataset than other models.

With the EFB, It puts the mutually exclusive features along with nothing but it will rarely take any non-zero value at the same time to reduce the number of features. This impacts the overall

result for an effective feature elimination without compromising the accuracy of the split point.

By combining the two changes, it will fasten up the training time of any algorithm by 20 times. So LGBM can be thought of as gradient boosting trees with the combination for EFB and GOSS.

The parameters we have selected in LightGBM model for tuning were max_depth and n_estimators

The best parameters found out to be : {'max_depth': 25, 'n_estimators': 100}

The accuracy ,precision,recall ,f1score and ROC score with the best parameter and the confusion and classification report after tuning as follows-

```
test score
**************************************************
       The accuracy is  0.8461229409401366
       The precision is  0.8602409638554217
       The recall is  0.9043698543381887
       The f1 is  0.8817536276628589
       the auc  is  0.8247123996965668

                                 classification report
**************************************************
              precision    recall  f1-score   support

           0       0.82      0.75      0.78       910
           1       0.86      0.90      0.88      1579

    accuracy                           0.85      2489
   macro avg       0.84      0.82      0.83      2489
weighted avg       0.84      0.85      0.84      2489
```

Model Summary

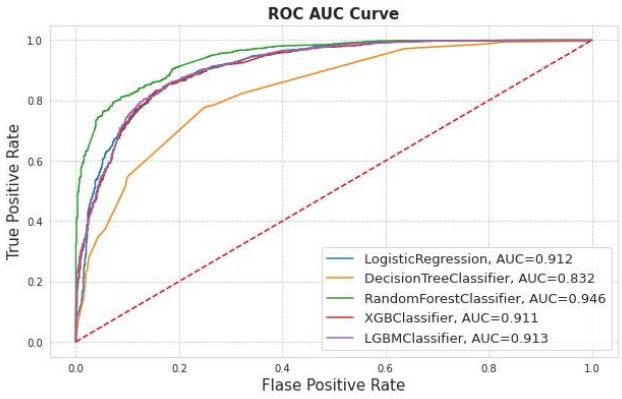
	Models	accuracy	precision	recall	f1	roc_auc
0	Logistic Regrestion	0.830856	0.813312	0.951868	0.877152	0.786374
1	Desision Tree	0.766171	0.837508	0.783407	0.809555	0.759835
2	Random forest	0.768788	0.736784	0.988809	0.844392	0.687882
3	XGboost	0.846123	0.857656	0.908170	0.882190	0.823316
4	LightGBM	0.846123	0.860241	0.904370	0.881754	0.824712

The best model for sentiment Analysis based on the test accuracy score will be LightGBM and Logistic Regression Model.

ROC_AUC Graph

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

AUC-ROC curve is a performance measurement for the *classification problems* at *various threshold* settings. ROC is a probability curve, and AUC represents the degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at *predicting 0s as 0s and 1s as 1s*. By analogy, Higher the AUC, better the model is at distinguishing between *patients with the disease and no disease*.



Clustering Models

Clustering is done on the basis of similarities between the data points. The similarities are understood by how closely distanced these points are. The following are some hypotheses that can be generated by finding some similarities in the visualized data:

Restaurants with similar kinds of ratings can be clustered together. Ratings are done by people on the basis of food quality, service, packaging among other things.

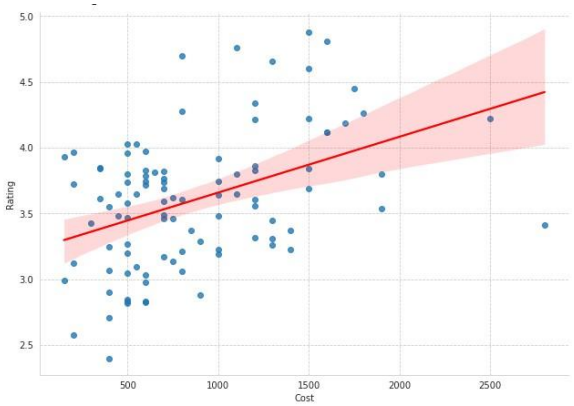
Restaurants with high ratings would also probably be expensive and would be having a similar pricing strategy as well. They can be clustered according to the costs.

Restaurants having some of the most popular cuisines can be clustered together and restaurants with exotic cuisines such as Indonesian, Mexican, Japanese, etc can be clustered as they are really low in number

For Clustering the restaurants,intially we have done data preprocessing and feature selection.

As a result we find a positive correlation between Cost and Rating features,hence we choose these two features ,along with the cuisine features as the data for implementing clustering algorithm.

We have also used Minmax Scaler inorder to scaled down Cost and Rating features before applying clustering algorithm.



We have used four clustering algorithms in this project-

K-Means

Principal Component Analysis

Hierarchical Clustering

DBSCAN

We will look into each clustering algorithm one by one.

K-Means Clustering

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the

minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way kmeans algorithm works is as follows:

Specify number of clusters *K*.

Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.

Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

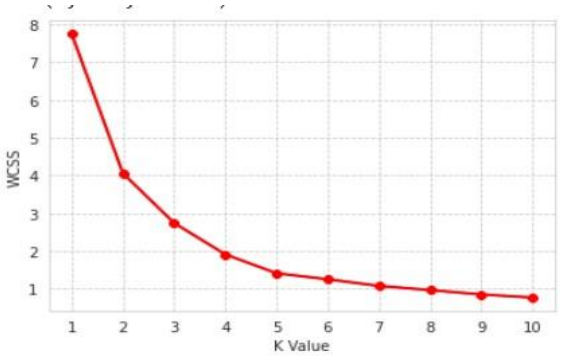
Compute the sum of the squared distance between data points and all centroids.

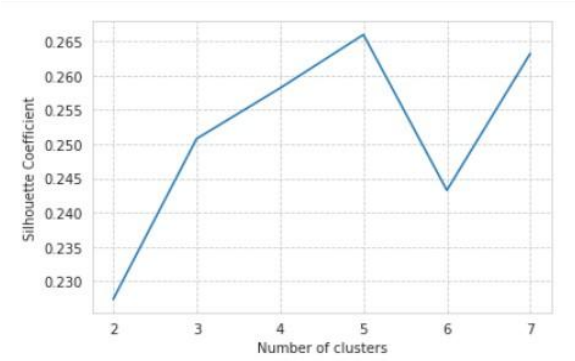
Assign each data point to the closest cluster (centroid).

Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach kmeans follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.

In-order to find out the optimal number of cluster,we have used both elbow method and silhouette method.





From the elbow method and silhouette method graphs,we have got 5 as the optimal number of clusters in K-means Clustering.

Principal Component Analysis

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.

The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower-dimensional space.

Now that you have understood the basics of PCA, let's look at the next topic on PCA in Machine Learning.

Applications of PCA in Machine Learning

PCA is used to visualize multidimensional data.

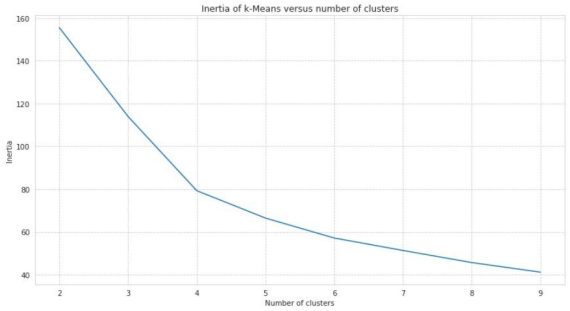
It is used to reduce the number of dimensions in healthcare data.

PCA can help resize an image.

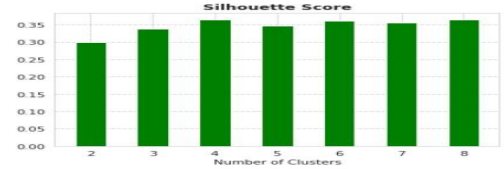
It can be used in finance to analyze stock data and forecast returns.

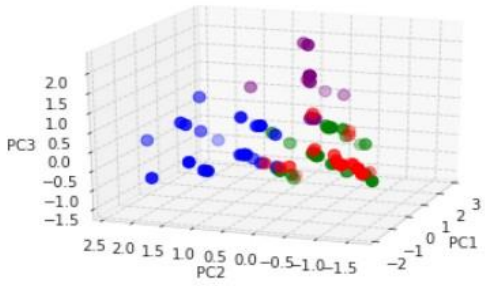
PCA helps to find patterns in the high-dimensional datasets.

In-order to find out the optimal number of cluster,we have used both elbow method and silhouette method.




```
Parameter: {'n_clusters': 2} Score 0.2997951415660126
Parameter: {'n_clusters': 3} Score 0.3380779519988871
Parameter: {'n_clusters': 4} Score 0.3650580517654094
Parameter: {'n_clusters': 5} Score 0.34667741109516814
Parameter: {'n_clusters': 6} Score 0.3604063459503715
Parameter: {'n_clusters': 7} Score 0.3561844373513979
Parameter: {'n_clusters': 8} Score 0.3649019696274996
Text(0.5, 0, 'Number of Clusters')
```





From the elbow method and silhouette method graphs,we have got 4 as the optimal number of clusters in Principal Component Analysis.

Hierarchical Clustering

Hierarchical clustering is another unsupervised learning algorithm that is used to group together the unlabeled data points having similar characteristics. Hierarchical clustering algorithms falls into following two categories.

Agglomerative hierarchical algorithms − In agglomerative hierarchical algorithms, each data point is treated as a single cluster and then successively merge or agglomerate (bottom-up approach) the pairs of clusters. The hierarchy of the

clusters is represented as a dendrogram or tree structure.

Divisive hierarchical algorithms − On the other hand, in divisive hierarchical algorithms, all the data points are treated as one big cluster and the process of clustering involves dividing (Top-down approach) the one big cluster into various small clusters.

Steps to Perform Agglomerative Hierarchical Clustering

We are going to explain the most used and important Hierarchical clustering i.e. agglomerative. The steps to perform the same is as follows −

Step 1 − Treat each data point as single cluster. Hence, we will be having, say K clusters at start. The number of data points will also be K at start.

Step 2 − Now, in this step we need to form a big cluster by joining two closet datapoints. This will result in total of K-1 clusters.

Step 3 − Now, to form more clusters we need to join two closet clusters. This will result in total of K-2 clusters.

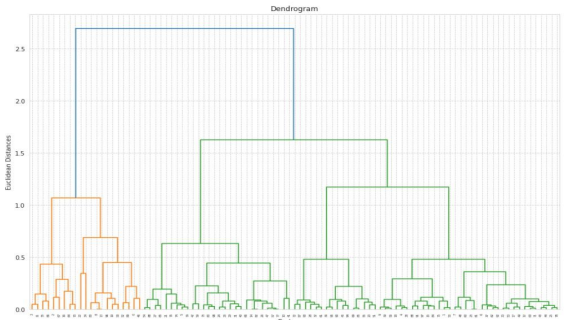
Step 4 − Now, to form one big cluster repeat the above three steps until K would become 0 i.e. no more data points left to join.

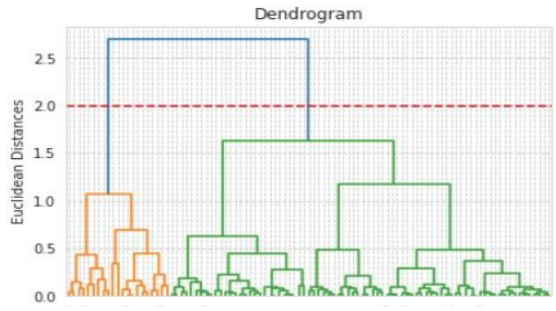
Step 5 − At last, after making one single big cluster, dendrograms will be used to divide into multiple clusters depending upon the problem.

Role of Dendrograms in Agglomerative Hierarchical Clustering

As we discussed in the last step, the role of dendrogram starts once the big cluster is formed. Dendrogram will be used to split the clusters into multiple cluster of related data points depending upon our problem.

In-order to find out the optimal number of cluster,we have used Agglomerative Hierarchical Clustering and plotted the dendogram.





From the dendogram graphs,we have got 2 as the optimal number of clusters in Hierarchical Clustering Algorithm.

DBSCAN

DBSCAN is an important Clustering technique for Machine Learning (ML) and Data Science in general.

DBSCAN falls under unsupervised learning, thus opening up more possibilities and increasing the range of applying data. This is especially true for information that can be extracted from data.

The importance of using Density-Based Clustering in Data Mining and Data Analytics can be seen throughout major implementations of Machine Learning techniques.

The advantages of DBSCAN in Data Mining, Analytics and ML can be seen throughout various services powered by clustering methodologies.

For instance, DBSCAN is a highly preferred learning methodology that promotes pattern recognition, behavioural analytics, market research, data analysis, and image processing.

Density-Based clustering groups data points that are similar in nature into a single cluster based on how densely grouped they are. In order for Density-Based Clustering to work, there must be a certain number of data points between the clusters as well or in the ‘neighbourhood’.

Density-Based clustering can not only accurately cluster the data points in a population or dataset but also works well with noise. When compared with K-means or Hierarchical clustering, the DBSCAN algorithm handles noise the best, correctly detecting it inside datasets.

DBSCAN is especially useful for working with outliers or anomalies and correctly detecting these outliers and points that stand out. DBSCAN clustering does not need the number of clusters to be specified prior to plotting and only needs two parameters, minPoints and epsilons.

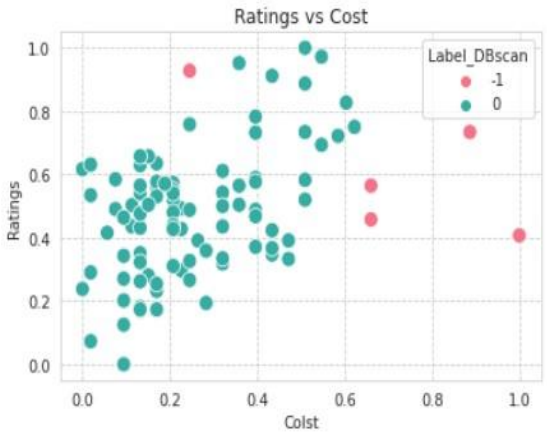
Epsilons are the radiuses of the circles that are built around every data point in order to analyse the density. Meanwhile, minPoints are specified to determine the number of data points that are required inside the plotted circle in order for data points to be identified as core points.

When the circle is represented using 3D or 2D variants, epsilons become the radiuses of these hyperspheres, while minPoints determine the minimum data points demanded inside these hyperspheres. There are three types of data points that we need to know about when working with DBSCAN clustering.

Core Point: Core points possess more value than the specified minPoints within the epsilon.

Border Point: Border points possess a lesser value than minPoints within the epsilon but in the neighbourhood of core points.

Outlier or Noise: These are points that are neither a border point nor a core point.



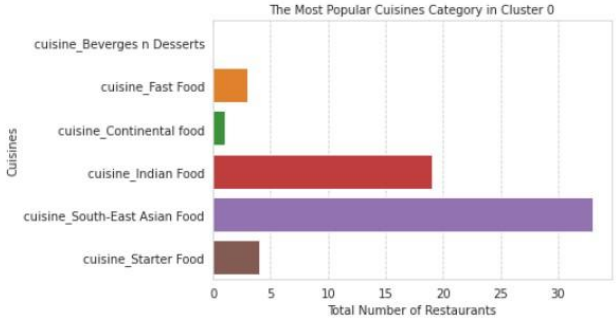
we have got 2 as the optimal number of clusters in DBSCAN as well.

Clustering Summary Table

SL No.	Model_Name	Data	Optimal_Number_of
1	K-Means with elbow method	RC	range between
2	K-Means with silhouette_score	RC	5
3	PCA (3 components) with elbow method	RC	range between
4	PCA (3 components) with silhouette_score	RC	4
5	Hierarchical clustering with dendogram	RC	2
6	Hierarchical clustering with silhouette_score	RC	2
7	DBSCAN	RC	2

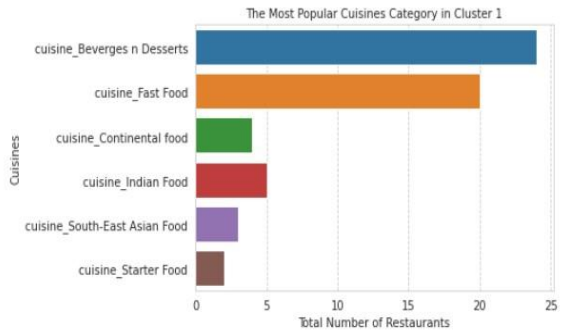
Cluster Analysis

For Cluster Analysis we are considering optimal number of clusters as 4 which we have obatained from our Principal Component Analysis thereby will be doing analysis and exploration of each clusters one by on



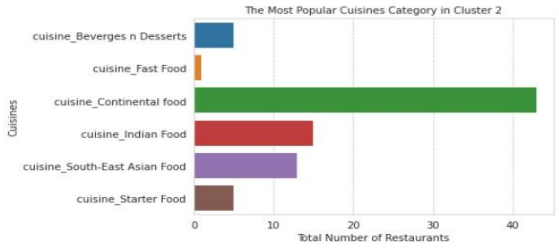
Cluster 0 Observation:

- The mostly available cuisine in the restaurants in cluster 0 is the South east asian cuisines followed by Indian cuisine.
- The restaurants in cluster 0 does not having Beverges and Deserts available.
- The average rating is 3.46 and the average cost is 923 INR which includes an outlier of cost 1150 INR and median cost of 525 INR. This means the restaurants are basically in general cheap in nature in this cluster .



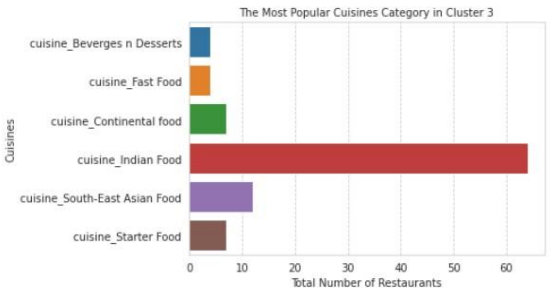
Cluster 1 Observation:

- The mostly available cuisine in the restaurants of cluster 1 is the Beverges and Desserts cuisines followed by Fast Food cuisine.
- The restaurants in cluster 1 have a few starter cuisines available
- The average rating is 3.61 and the average cost is 736 INR which includes an outlier of cost 2500 INR and median cost of 600 INR.. These restaurants are slightly higher in prices than cluster 0.



Cluster 2 Observation:

- The mostly in demand and available cuisine in the restaurants of cluster 2 is the Continental cuisines followed by Indian cuisine.
- The restaurants in cluster 2 have a few fast food cuisines available.
- The average rating is 3.82 and the average cost is 1052 INR which includes an outlier of cost 2800 INR and median cost of 1100 INR.. These restaurants are fine dining restaurants and expensive as well compared to other clusters like 0 and 1.

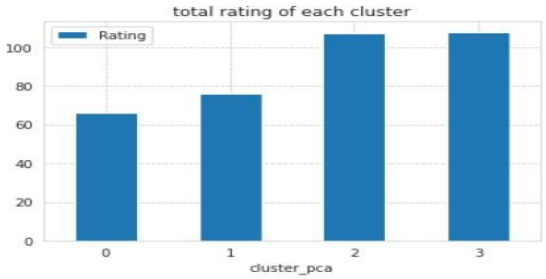


Cluster 3 Observation:

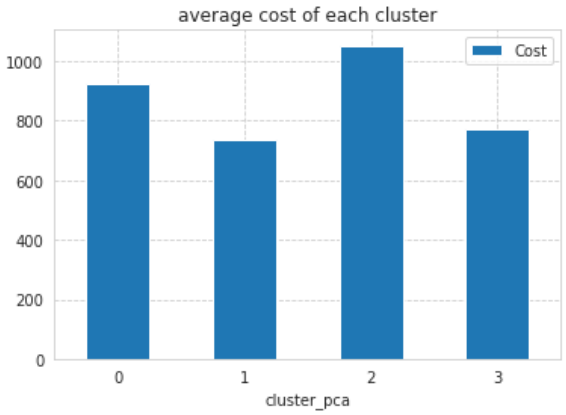
- The mostly in demand and available cuisine in the

restaurants of cluster 3 is the Indian cuisines.

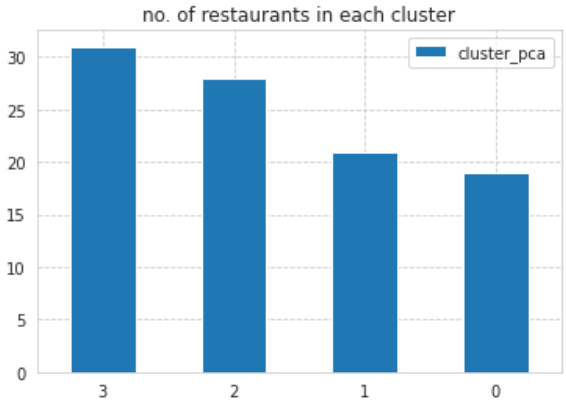
- The restaurants in cluster 2 have a very few demand of fast food and beverages.
- The average rating is 3.48 and the average cost is 769 INR which includes an outlier of cost 1700 INR and median cost of 700 INR.. These restaurants are afforadable and mostly preferred cuisine in high numbers in these restaurants.



From the above plot,we can say that both the clusters 2 and 3 were having good restaurants in their respective clusters with high ratings



From the above plot,we can see that the clusters 2 restaurants were the most expensive ones. On the other hand, Cluster 1 restaurants were the most affordable ones.



From the above plot,we can say that the clusters 3 were having pretty high numbers of restaurants in its cluster comparatively.

Cluster Analysis Summary

The Cluster 2 is the most expensive and highly rated cluster.The cuisines which are highly on demand in these restaurants are Continenatel Food and the least in demand cuisine is the Fast Food in this cluster.

The Cluster 0 is the least rated cluster.The cuisines which are highly on demand in these restaurants are South East Asian Cuisines.

The Cluster 1 is the most affordable and moderately rated cluster with Beverages and desert cuisine are highly in demand followed by fast food.

The Cluster 3 is the most popular cluster for the Indian Cuisines and restaurants in these cluster is compartively higher than other cluster.Hence the restaurants in cluster 3 is the busiest and the highly on demand cuisine. The Rating and Costing in this cluster is compartively in a balanced and moderate range,hence based on rating and costing parameter we can say that cluster 3 is one of the best choice of any average income individual.

Cost-Benefit Analysis

A Cost-Benefit Analysis is a process of analyzing the worth of a decision by estimating the costs incurred in implementing that decision and comparing them with the benefits of that decision. If the projected benefits outweigh the costs, you'll be making money out of that decision and if not, it's important to strategize a better plan.

Zomato is an Indian restaurant search and an online food delivery service. Zomato focuses on online food ordering, restaurant reservations, and loyalty programs. The target customers for the company are restaurant chains that want to reach a larger audience and application users who just want to try out local restaurants and various cuisines. Here is a simple cost-benefit analysis that can be carried out on the basis of the little information we can assume.

Costs

When tallying costs, beginning with direct costs, which include expenses directly related to the production or development of a product or service (or the implementation of a project or business decision) which is in the case of Zomato is primarily the mobile application. Maintaining the application, strategizing plans, including the restaurants, marketing, food delivering partners and customer support needs a huge team to work on. The salaries of the employees would be a direct cost.

Other indirect costs include utilities, rent, partners, advertisers, etc.

There are some other costs that are difficult to measure such as negative reviews on the platform which leads to people avoiding the application altogether, bad presence on social media, etc.

Benefits

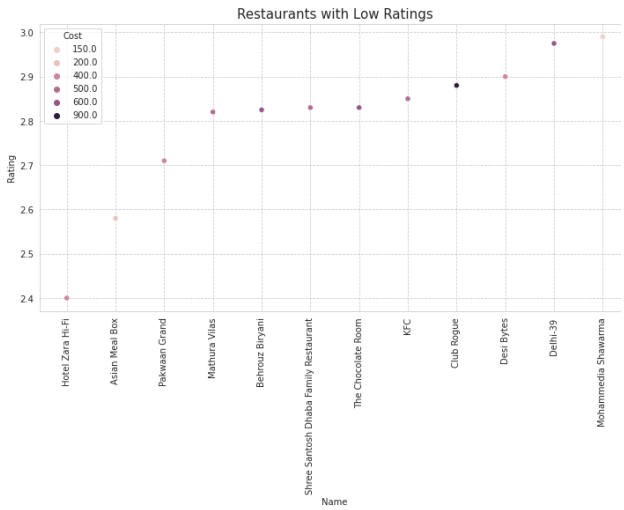
The major source of Revenue is Advertising. More and more restaurants want to promote themselves on the Zomato feed in order to gain attention and visibility from a large section of Zomato subscribers and customer base.

Through the food delivery service, Zomato charges a commission to the restaurants on the basis of orders. The company earns through restaurants that pay a commission for each delivery, which is then split among the delivery partners and the company. However, online food delivery only contributes a low percentage of income compared to other revenue streams because of the huge competition and the need to provide deep discounts, etc.

Comparison

The data that we have consists of per-person cost, cuisines available at the restaurant, and an average rating of the restaurant. If a restaurant isn't performing well in terms of rating and has a high per-person cost and a low number of popular cuisines, this is going to be a problem for Zomato. Since negative reviews would be an intangible cost to the company and with that the company will start to lose daily application users. The application users are an asset to the company, Zomato gets advertising by different restaurants because of the large audience they have.

All in all, it is important to separate out the restaurants that Zomato needs to work on in order to improve its overall customer experience and if improvement strategies don't work out, they need to delist those restaurants themselves.



- The most of the restaurants with lower ratings are providing with Indian Cuisines and adding different variety of cuisines in these restaurants may help in improving the ratings
- Mohammedia Shawarma has the highest rating with the lowest cost. It seems it is doing well in its capacity.On the contrary , Hotel Zara Hi-Fi has the lowest rating with moderately high in pricing.
- These restaurants are basically small food joints or restaurants with high prices according to the food they are serving. Efforts should be made to advertise more and analyze the reviews, especially for these restaurants, and work on them.

Conclusion

Some of the important conclusions to be drawn are:

- The most popular cuisines available in the most of the restaurant is the North Indian and Chinese.
- The most frequent working hours of the restaurant is between 11am to 11pm.
- The billing amount of rupees 500 is the most frequent cost paid by the customers on their order.
- The collage-Hyatt Hyderbad Gachibowli is the most expensive restaurant available and the most affordable restaurant for the

customers are the Amul and Mohammedia Shawarma.

- The most expensive Cuisine is the Modern India cuisine which cost around 2000 rupees and the least expensive item available at a cost of 200 rupees is the Mithai
- Great Buffet is one of the most common Tags given to the zomato restaurants with nearly more than 40 restaurants.
- Namit Aggarwal is one of the active reviewer based on number of followers and ratings provided
- The Restaurant with the highest rating of nearly 4.8 and good reviews is the AB's Absolute Barbecues. On the contrary, the restaurant with worst reviews and rating is the Hotel Zara Hifi with a rating less than 2.5.
- The Cost-benefit analysis on Zomato with a few assumptions one basis of the little business understanding that could be gathered, it can be concluded that it is important to separate out the restaurants with the lowest rating in order to improve its overall customer experience. Actions should be made to advertise more and analyze the reviews, especially for low rated restaurants, and work on them.
- Sentiment Analysis was done on the reviews and a model was trained in order to identify negative and positive sentiments. The best model for sentiment analysis we found out to be XGboost,LightGBM and Logistic Regression model.
- Restaurant Clustering was done in with just two features Cost and Rating. Kmeans Clustering gave us optimal cluster value as 5 but we have done our clustering analysis based upon the principal component analysis because the similarities in the data points within the clusters were pretty great. We have got optimal clusters as 4 clusters in PCA.

Cluster 0 - The mostly available cuisine in the restaurants in cluster 0 is the South east asian cuisines followed by Indian cuisine and the restaurants in cluster 0 does not having Beverges and Deserts available. The average rating is

3.46 and the average cost is 923 INR which includes an outlier of cost 1150 INR and median cost of 525 INR. This means the restaurants are basically in general cheaper in nature in this cluster.

Cluster 1 - The mostly available cuisine in the restaurants of cluster 1 is the Beverges and Desserts cuisines followed by Fast Food cuisine.The average rating is 3.61 and the average cost is 736 INR which includes an outlier of cost 2500 INR and median cost of 600 INR.. These restaurants are slightly higher in prices than cluster 0.

Cluster 2 - The mostly in demand and available cuisine in the restaurants of cluster 2 is the Continental cuisines followed by Indian cuisine.The average rating is 3.82 and the average cost is 1052 INR which includes an outlier of cost 2800 INR and median cost of 1100 INR.. These restaurants are fine dining restaurants and expensive as well compared to other clusters like 0 and 1.

Cluster 3 - The mostly in demand and available cuisine in the restaurants of cluster 3 is the Indian cuisines.The restaurants in cluster 2 have a very few demand of fast food and beverages.The average rating is 3.48 and the average cost is 769 INR which includes an outlier of cost 1700 INR and median cost of 700 INR.. These restaurants are afforadable and mostly preferred cuisine in high numbers in these restaurants.

Recommendations

- Ratings should be collected on a category basis such as rating for packaging, delivery, taste, quality, quantity, service, etc. This would help in targetting specific fields that are lagging.

Challenges

- Extracting new features from existing features like Metadata and Timings features in review dataset were a bit tedious job to do.
- Handling null values in both restaurant and review dataset and the text pre-processing in review feature was a challenging task.
- There were few undefined data records present in Collection feature and few duplicate records

in review dataset.

- Finding Optimum number of clusters
- Selecting different parameter for hyperparameter tuning and finding the best parameters has to follow a trial n error technique,which is again a challenging job.

References

https://www.codingninjas.com/blog/2021/07/09/dbscan-clustering-in-machine-learning/

https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_clustering_algorithms_hierarchical.htm

https://www.simplilearn.com/tutorials/machine-learning-tutorial/principal-component-analysis

https://www.ibm.com/cloud/learn/random-forest

https://www.geeksforgeeks.org/xgboost/

https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5

https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/

https://www.educative.io/blog/one-hot-encoding

https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a