**Capstone Project**

**Zomato Restaurant Clustering And Sentiment Analysis**

**Technical Documentation**

**By**

# 

**Team**

**Rahul Kumar Soni, Lakdawala Ali Asgar**

**AlmaBetter**

****

**Date: 26th October 2021.**

**Table of Contents**

[**Abstract:**](#_heading=h.30j0zll) **4**

[**Problem Statement:**](#_heading=h.1fob9te) **4**

[**Introduction:**](#_heading=h.llj7yvkftmv0) **4**

[**Data Summary:**](#_heading=h.3znysh7) **5**

[Zomato Restaurant names and Metadata](#_heading=h.29htfo89gr3j) 5

[Zomato Restaurant reviews](#_heading=h.ohpy666eqcev) 5

[**Steps involved:**](#_heading=h.2et92p0) **5**

[Null values Treatment](#_heading=h.3dy6vkm) 6

[Outliers treatment](#_heading=h.uvvpq2phydpx) 6

[Exploratory Data Analysis](#_heading=h.tyjcwt) 6

[Encoding of categorical columns](#_heading=h.4d34og8) 6

[Standardization of features](#_heading=h.3rdcrjn) 6

[Fitting different models](#_heading=h.lnxbz9) 6

[Clustering](#_heading=h.mptcd6usljax) 6

[Sentiment analysis unsupervised](#_heading=h.9ydmxy3y1ko0) 6

[Sentiment analysis supervised](#_heading=h.f5bz7sd043zz) 6

[Tuning the hyperparameters for better recall](#_heading=h.35nkun2) 7

[Features Explainability](#_heading=h.44sinio) 7

[**Algorithms:**](#_heading=h.2jxsxqh) **7**

[K means clustering:](#_heading=h.21qxp5c2zsyo) 7

[Hierarchical clustering:](#_heading=h.c9q6o2o0vmuk) 7

[LDA:](#_heading=h.lqutas7flk6v) 8

[Non-negative matrix Factorization:](#_heading=h.73p1g6qv61v3) 8

[Logistic Regression:](#_heading=h.z337ya) 9

[MultinomialNB:](#_heading=h.1y810tw) 9

[Random Forest Classification:](#_heading=h.4i7ojhp) 9

[XGBoost Classification:](#_heading=h.1ci93xb) 10

[LightGBM Classification:](#_heading=h.2bn6wsx) 10

[**Model performance:**](#_heading=h.8ebp863vawzg) **11**

[Accuracy](#_heading=h.3as4poj) 11

[Precision](#_heading=h.49x2ik5) 11

[Recall](#_heading=h.147n2zr) 11

[F1-Score](#_heading=h.3o7alnk) 11

[AUC-ROC](#_heading=h.tr9uqhwkmc0q) 12

[**Hyperparameter tuning:**](#_heading=h.23ckvvd) **12**

[**Conclusion**](#_heading=h.k175mjffwtw5) **13**

# Abstract:

In today’s digital world, food apps like Zomato are widely used because it provides a platform for people to share their opinion about the restaurants and cafes they have visited. This paper includes an analysis of client ratings and reviews in Zomato utilizing content mining. Utilizing content mining, break down the content audits/reviews from the client with a specific end goal to create productive results and legit surveys. The rating has a review of the restaurant which can be used for sentiment analysis. Based on this, writers want to discuss the sentiment of the review to be predicted. The method used for preprocessing the review is to make all words lowercase, tokenization, remove numbers and punctuation, stop words, and lemmatization. Then after that, we create a word to vector with the term frequency-inverse document frequency (TF-IDF). The data that we process are 10,000 reviews. After that, we make positive reviews that have a rating of 3.5 and above, negative reviews that have a rating of 3 and below. We have used Split Test, 75% Data Training and 25% Data Testing. The metrics used to determine classifiers are precision, recall, accuracy, F1 score.

# 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 make some useful conclusions in the form of Visualizations. Also, cluster the Zomato restaurants into different segments. The data is visualized as it becomes easy to analyze data in an instant. The Analysis also solves some of the business cases that can directly help the customers find the Best restaurant in their locality and for the company to grow up and work in 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 costs 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.

# Introduction:

In today’s digitized modern world, the popularity of food apps is increasing due to their functionality to view, book, and order food with a few clicks on the phone for their favorite restaurant or cafes, by surveying the user ratings and reviews of the previously visited customers. Food apps like Zomato provide a secular part where users can rate their experience of the visited restaurant or café. Zomato also provides columns for writing classified user reviews. Sharing on the internet is something we usually do. Giving a review is also a useful activity so that other people on the internet can find out something else and see opinions about things. The usual things are reviewed by someone in the form of experiences, places, objects, and others. When giving a review we usually use text to explain something that we experience with an item, place, or event that we normally experience.

Zomato is a site where someone can give a review of a restaurant, how the restaurant is, and someone's opinion about the restaurant. Restaurant customer satisfaction can be analyzed by their review on Zomato. Sometimes, restaurants see the reviews in Zomato, but they don't get if the reviews are positive or negative to their restaurants. Reviews on Zomato are still in the form of text and can be classified with positive, negative, or neutral ratings. Zomato doesn’t have an analysis of how users interact with the reviews and what words will indicate whether they like it or not. We need to extract the words in review and analyze them so we can know how users interact in Zomato and get customers' satisfaction by their reviews.

In this paper, we propose a method to analyze users’ sentiment of Zomato Restaurants. We are using different classifiers to classify the sentiments of users based on their reviews. We also find words that affect the classifier model. Also, we focus on mining customer reviews, authenticating them, and classifying them into positive and negative reviews. We also clustered the restaurants based on their cuisines

# Data Summary:

## Zomato Restaurant names and Metadata

* 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

## Zomato Restaurant reviews

* 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 the review

# Steps involved:

## Null values Treatment

The data set had null values which out of which we replace some with the mean of the feature some by zero and dropped some observations which were almost filled with null values.

## Outliers treatment

Isolation Forests(IF), similar to Random Forests, are built based on decision trees. And since there are no predefined labels here, it is an unsupervised model.IsolationForests were built based on the fact that anomalies are the data points that are “few and different”.In an Isolation Forest, randomly sub-sampled data is processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they require more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate anomalies as it was easier for the tree to separate them from other observations.

## Exploratory Data Analysis

We performed univariate and bivariate analyses. This process helped us figure out various aspects and relationships among variables. It gave us a better idea of which feature behaves in which manner.

## Encoding of categorical columns

We used One Hot Encoding(converting to dummy variables) to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to the numerical format.

## Standardization of features

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

## Fitting different models

For modeling, we tried various algorithms like:

### Clustering

1. K means clustering
2. Hierarchical clustering

### Sentiment analysis unsupervised

1. LDA
2. Non-negative matrix Factorization

### Sentiment analysis supervised

1. MultinomialNB
2. Logistic Regression
3. Decision Tree
4. Random Forest Classification
5. XGBoost Classification
6. LightGBM Classification

## Tuning the hyperparameters for better recall

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in the case of tree-based models like Random Forest Classifier and XGBoost classifier.

## 

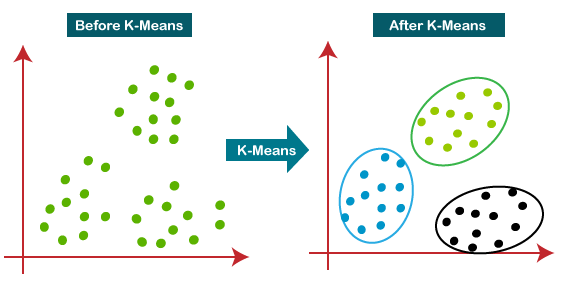
## Features Explainability

We have applied SHAP on the XGBoost and CatBoost model to determine the features that were most important while predicting an instance and also build a feature importance graph to find out which features were important and which were redundant in a model

# Algorithms:

## K means clustering:

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

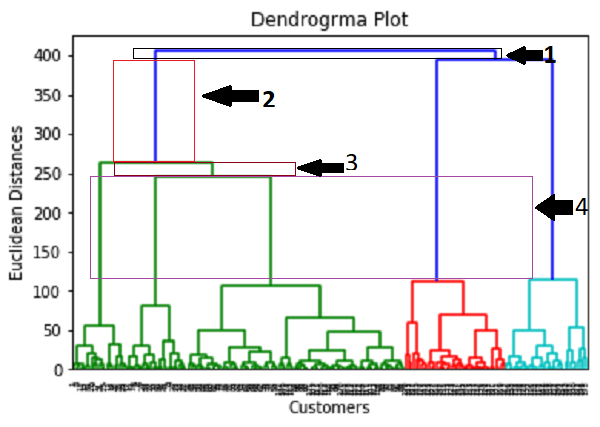


## Hierarchical clustering:

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

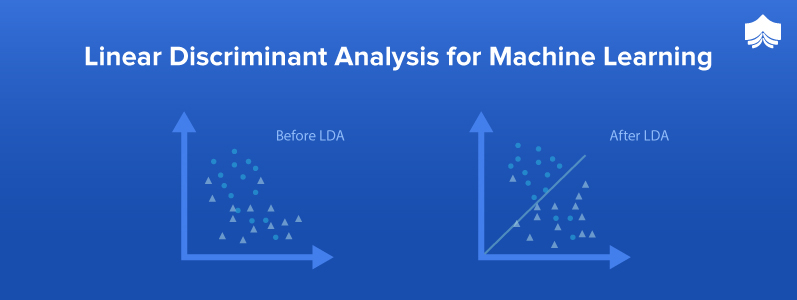
Agglomerative hierarchical algorithms − In agglomerative hierarchical algorithms, each data point is treated as a single cluster and then successively merge or agglomerate (bottom-up approach) the pairs of clusters. The hierarchy of the clusters is represented as a dendrogram or tree structure.

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.



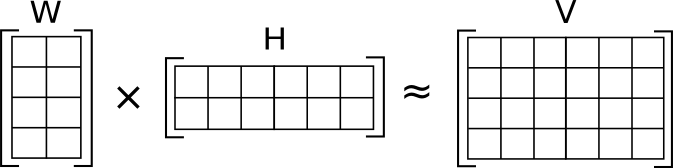
## LDA:

Linear Discriminant Analysis or LDA is a dimensionality reduction technique. It is used as a pre-processing step in [Machine Learning](https://www.knowledgehut.com/blog/data-science/what-is-machine-learning) and applications of pattern classification. The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs. The original technique was developed in the year 1936 by Ronald A. Fisher and was named Linear Discriminant or Fisher's Discriminant Analysis. The original Linear Discriminant was described as a two-class technique. The multi-class version was later generalized by C.R Rao as a Multiple Discriminant Analysis. They are all simply referred to as Linear Discriminant Analysis.LDA is a supervised classification technique that is considered a part of crafting competitive machine learning models. This category of dimensionality reduction is used in areas like image recognition and predictive analysis in marketing.



## Non-negative matrix Factorization:

NMF stands for non-negative matrix factorization, a technique for obtaining low-rank representation of matrices with non-negative or positive elements. Such matrices are common in a variety of applications of interest. For example, images are nothing but matrices of positive integer numbers representing pixel intensities. In information retrieval and text mining, we rely on term-document matrices for representing document collections. In recommendation systems, we have utility matrices showing customers’ preferences for items.

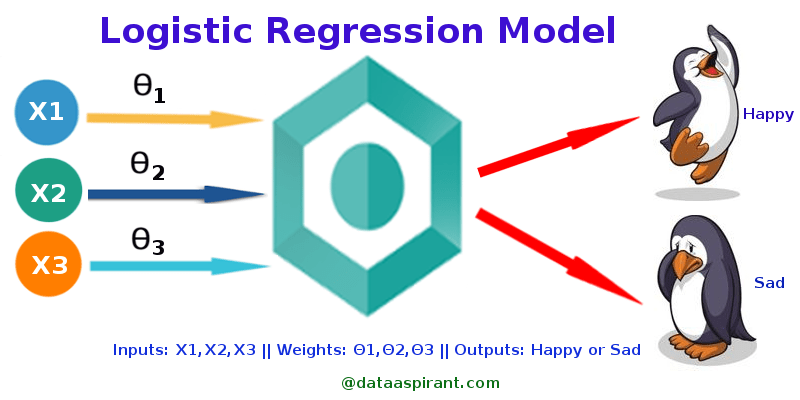


## Logistic Regression:

Logistic regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example,

* To predict whether an email is a spam (1) or (0)
* Whether the tumor is malignant (1) or not (0)



## 

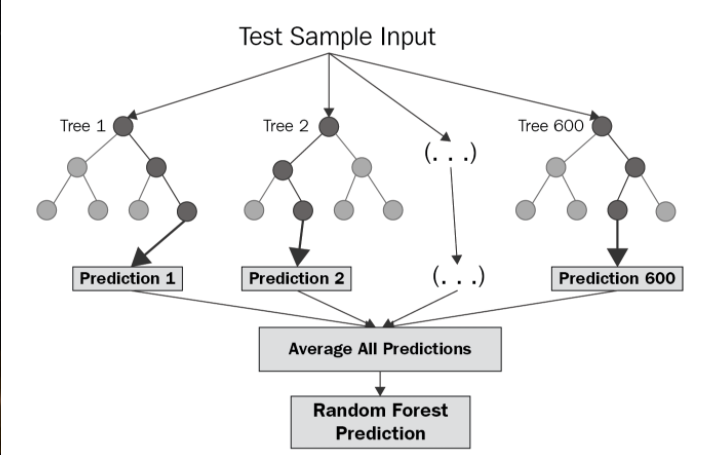
## MultinomialNB:

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle, and that is each feature being classified is not related to any other feature. The presence or absence of a feature does not affect the presence or absence of the other feature.

## Random Forest Classification:

Random Forest is a bagging type of Decision Tree Algorithm that creates several decision trees from a randomly selected subset of the training set and n features, collects the values from these subsets, and then averages the final prediction out of all n number of decision trees



## XGBoost Classification:

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

The implementation of the algorithm was engineered for the efficiency of computing time and memory resources. A design goal was to make the best use of available resources to train the model. Some key algorithm implementation features include:

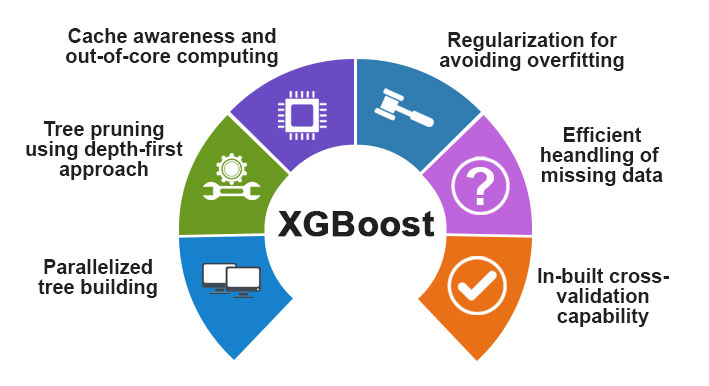
* **Sparse Aware** implementation with automatic handling of missing data values.
* **Block Structure** to support the parallelization of tree construction.
* **Continued Training** so that you can further boost an already fitted model on new data.

XGBoost is free open source software available for use under the permissive Apache-2 license.

Why Use XGBoost?

The two reasons to use XGBoost are also the two goals of the project:

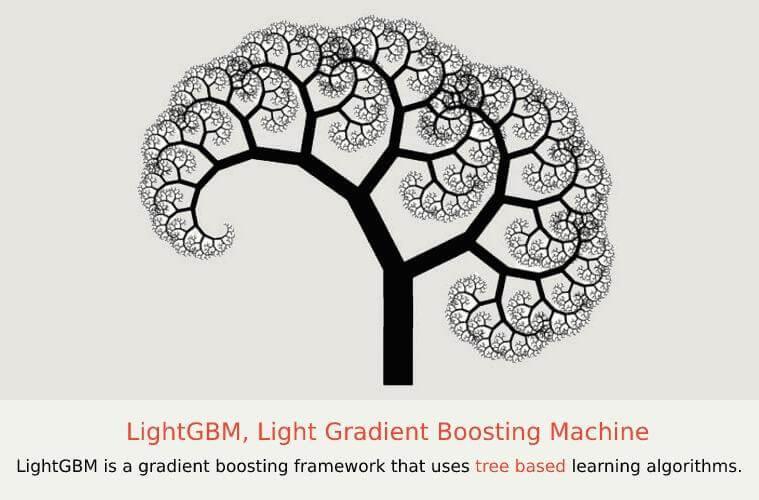
1. Execution Speed.
2. Model Performance.



## LightGBM Classification:

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

* Faster training speed and higher efficiency.
* Lower memory usage.
* Better accuracy.
* Support of parallel, distributed, and GPU learning.
* Capable of handling large-scale data.



# 

# 

# 

# 

# Model performance:

The model can be evaluated by various metrics such as:

## Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost the same. Therefore, you have to look at other parameters to evaluate the performance of your model. For our model, we have got 0.803 which means our model is approx. 80% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

## Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP

## Recall

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

Recall = TP/TP+FN

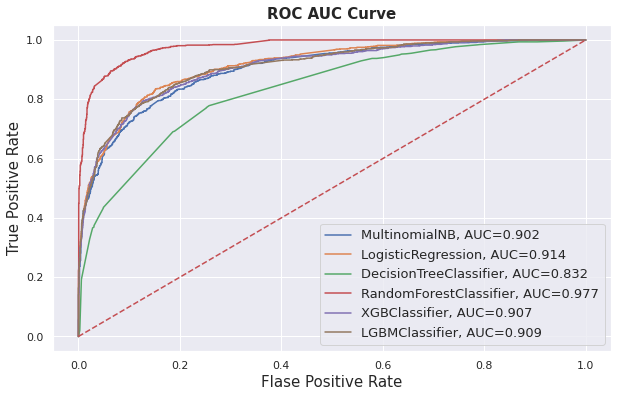
## F1-Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar costs. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

## AUC-ROC

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**.**



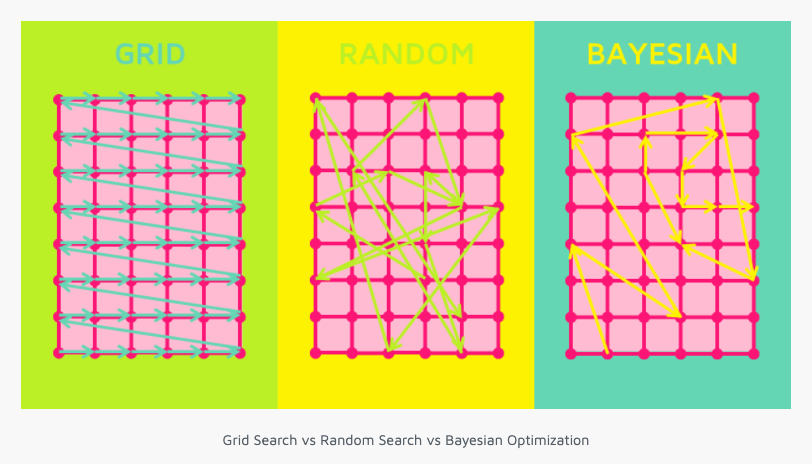
# Hyperparameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions of impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects the performance, stability, and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV for hyperparameter tuning. This also results in cross-validation and in our case we divided the dataset into different folds.

Grid Search CV-Grid:

Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.



# 

# Conclusion

That's it! We reached the end of our exercise.Starting with loading the data so far we have done EDA, null values treatment, encoding of categorical columns, feature selection, and then model building.

For clustering, we have decided on 3 clusters after the Silhouette score plot and elbow plot where we used KMeans clustering and Hierarchical clustering algorithms.

For sentiment analysis, we used both supervised and unsupervised techniques. In unsupervised we converted our 4 clusters to 2 cluster i.e positive sentiment and negative sentiment after careful analysis of the clusters.In supervised learning we converted the ratings above 3.5 as positive and the rating below 3.5 as negative after which we got the best model as logistic regression and Lightgbm classifier.