Gaining insights from Restaurant Aggregator(s) to predict a Restaurant's Performance

Submitted by
Gunjan Panda
Pragya Yadav
Shubhangi Agrawal

Abstract

When opening a new restaurant, it is critical to know what areas to best invest in to make your venture successful.[1] This depends on various factors like location, cuisines offered, theme, etc. This project aims to predict the performance of a restaurant on gaining insights from a restaurant aggregator, Zomato. The data studied and analyzed is of the restaurants in Bangalore till 2019, which after cleaning was trained over different classification models for further predictions.

1 Introduction

1.1 Problem Statement

The Indian Restaurant Market is said to be one of the fastest growing in the world. As per the National Restaurant Association of India (NRAI) report, the market is growing at a compounded rate of 9%. This is because opening a restaurant in the near future would draw new opportunities for the budding restaurant owners despite the challenges faced by the restaurant owners in recent years.[3] Thus, opening a restaurant requires knowledge of various critical areas to invest in, in order to make the venture successful. This is often not straightforward as it depends on various factors like location, cost, cuisine and other services they offer like home delivery etc.[1] With the number of restaurants opening each day, a new restaurant owner with the help of these factors can determine how to best attract more customers thus scoping out their competition and improve their overall rating. The aim of this project is to gain insights from a restaurant aggregator (Zomato) and predict the performance of a restaurant based on three categories: Excellent, Very Good and Satisfactory.

1.2 Data

The dataset used for this prediction has been taken from Kaggle which contains information regarding the restaurants in Bangalore that was fetched using web scraping in March, 2019 and later published on Kaggle. The data was fetched from the zomato website to gain insights from the restaurants already open in Bangalore and how they perform, to further predict the performance of a new restaurant that one can open in Bangalore, based on various parameters.

The dataset is available at

`https://www.kaggle.com/datasets/absin7/zomato-bangalore-dataset`

Table 1 shows the features present in the dataset (51717 X 17).

Table 1: Dataset features

Dataset	
Variables	Description
url	url of the restaurant in the zomato website
address	address of the restaurant in Bangalore
name	restaurant name
online_order	restaurant takes/doesn't take online orders
book-table	restaurants have/don't have the facility to book tables
rate	overall rating of the restaurant out of 5
votes	number of votes by customer for each restaurant
phone	phone number of the restaurant
location	neighborhood in which the restaurant is located
rest-type	restaurant type
dish_liked	dish of a restaurant liked by people
cuisines	cuisines served in restaurants
approx_cost(for two people)	approximate cost for two at a restaurant
reviews_list	list of reviews by customers
menu_item	items in menu of a restaurant
listed_in(type)	what kind of restaurant is it
listed_in(city)	city where restaurant is located

2 Methodology

2.1 Pre-Processing

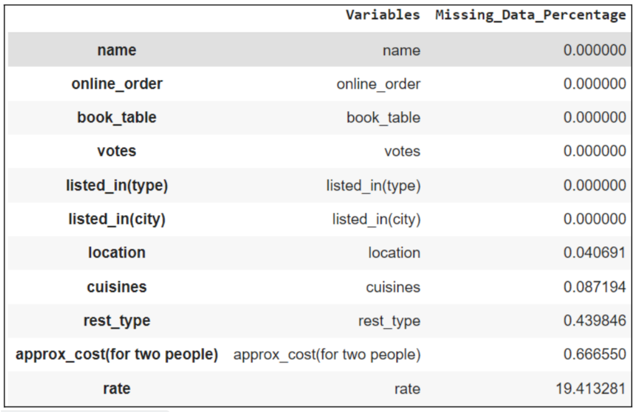
Being the first and the crucial step in building any machine learning model, this step ensures that the raw data so collected is transformed into a clean dataset whose preparation in detail is discussed in the following subsections. Out of the 17 features present in the dataset, 6 features namely, url, address, phone, menu_item, dish_liked, reviews_list, were dropped as these will not help in providing any useful insights to the data leaving us with 11 features for further analysis.

2.1.1 Dropping Duplicates

A total of 108 duplicate rows were found and later dropped from the data. This is after the model was checked with and without the duplicates being included and it was observed that the latter gave a higher accuracy which is why they were dropped.

2.1.2 Missing Value Analysis

For the variable 'rate', the missing values were imputed by the median of its observed values. This is because the data has outliers and median is immune to outliers. According to our data, the missing value %age of the variables can be seen in Figure 1.



	Variables	Missing_Data_Percentage
name	name	0.000000
online_order	online_order	0.000000
book_table	book_table	0.000000
votes	votes	0.000000
listed_in(type)	listed_in(type)	0.000000
listed_in(city)	listed_in(city)	0.000000
location	location	0.040691
cuisines	cuisines	0.087194
rest_type	rest_type	0.439846
approx_cost(for two people)	approx_cost(for two people)	0.666550
rate	rate	19.413281

Figure 1: Missing values

It is observed in Figure 1 that, ‘location’, ‘cuisines’, ‘approx_cost(for two people)’, ‘rest_type’ and ‘rate’ have missing values out of which ‘rate’ has the highest %age of missing values. Since, the %age of missing values for other variables is even less than a %, they were dropped directly. For the variable ‘rate’, the missing values were imputed by the mean of its observed values.

2.1.3 Cleaning Data

Some columns (which were renamed later for simplicity) from the data need to be treated better which is described below. Henceforward, a more refined data is obtained after which this will be used further for visualizations.

2.1.4 Cleaning Rate feature

Ratings on Zomato are quoted in the range between 0 and 5. For the data used, some entries are bounded by other string values like ‘New’ and ‘-’ which are replaced by ‘Nan’ and these missing values are treated as described in the above section. Also, the rating is given in fractional form (p/q) which is treated by retaining just the numerator (Eg: 3.8/5 → 3.8). After this, the ‘rate’ variable contains float values in the range 0 to 5.

2.1.5 Cleaning CostForTwo feature

This feature can be a large number and the given string is defined as per the Indian Numeral System where ‘,’ is also included. This ‘,’ is replaced by an empty string ‘ ’ and the entire column is then converted to numeric type ie. float (Eg: ‘1,300’ → 1300.)

2.1.6 Cleaning Restaurant_Type feature

It was observed that this variable consists of certain categories where the number of restaurants that fall under those respective categories were less in number ie. less than 100, which were mapped to a separate category ‘Others’.

2.1.7 Cleaning Location feature

It was noticed that the feature ‘Location’ served the purpose for the feature ‘City’ and so ‘City’ was dropped from the data. Also, it was observed that there existed localities in Bangalore where the number of restaurants were less (less than 100) and so, those localities were clubbed together into a separate category ‘Others’

2.2 Data Visualization(s)

The pre-processed data contains 10 features (Categorical:7, Numerical:3). This section contains the Exploratory Data Analysis to gain insights about the data.

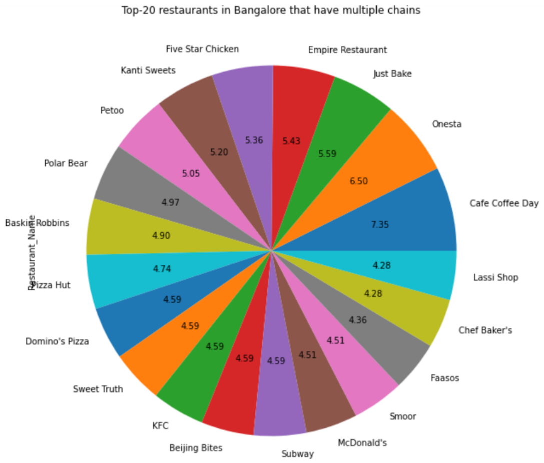


Figure 2: Top-20 restaurants

The pie chart in Figure 2, shows the Top-20 restaurants in Bangalore. These are mainly franchises of the major food chains in the state. 'Cafe Coffee Day' having the highest distribution indicates that the number of outlets of this coffee chain is the highest. Also, the headquarter of 'Cafe Coffee Day' being Bangalore justifies it having the most number of outlets.

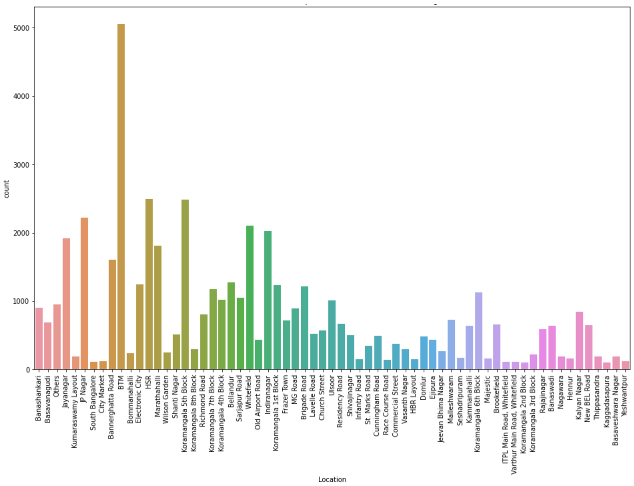


Figure 3: Number of restaurants in different localities in bangalore

Figure 3 displays the overall restaurant count per locality from which it can be inferred that 'BTM' has the highest number of restaurants in its vicinity. This is because BTM is one of the most popular residential and commercial places in Bangalore and is known for its music venues, cafes and boutiques.

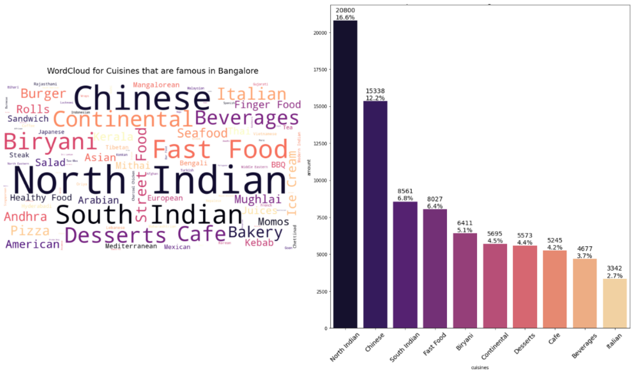


Figure 4: Top 10 cuisines served by restaurants in bangalore

Figure 4 shows the Top 10 cuisines served by restaurants in Bangalore. We can see that the cuisine most served is ‘North Indian’. This could be because the population residing in Bangalore accounts for a majority relocating from Northern India for studies, job, businesses away from home which are the most dependent on restaurant food. This could fetch such restaurants more profit due to its high demand.

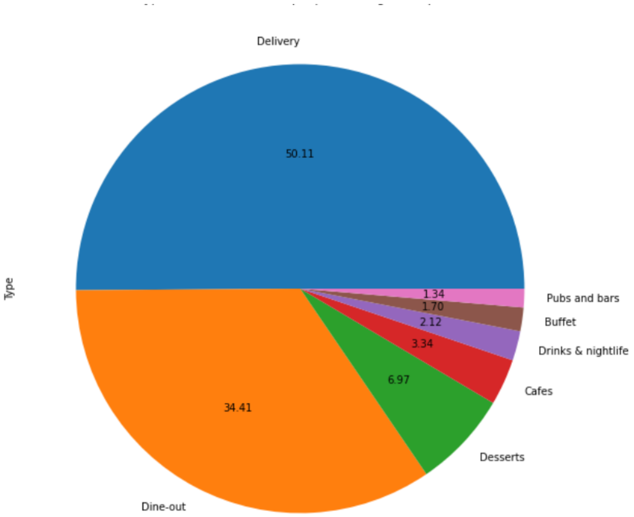


Figure 5: Distribution of the type of restaurants in bangalore

Figure 5 shows the ‘Type’ of restaurant in Bangalore. Majority of the restaurants are of the type- ‘Delivery’. Such restaurants require far less capital investments and gain far more profits as compared to the restaurants of the other type. On the contrary, restaurants of the type- ‘Pubs and Bars’ account the least. This could be because such restaurants cater to a far less %age of the population mainly because these are not considered to be family friendly and have an age restriction.

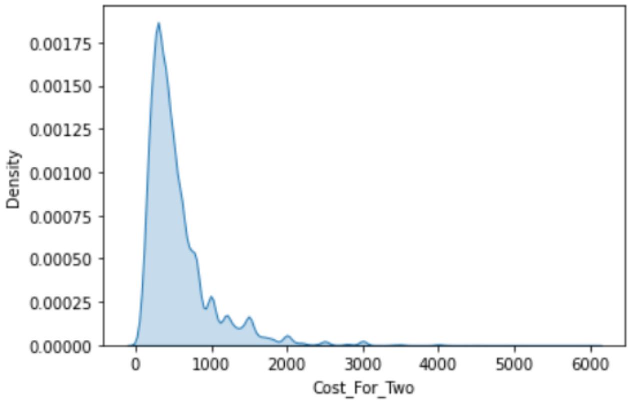


Figure 6: Cost for two at different restaurants

Figure 6 shows that the 'CostForTwo' for most of the restaurants is around Rs. 500 - Rs. 600 which means these are more pocket friendly and so will be preferred especially by the ones running on a budget.

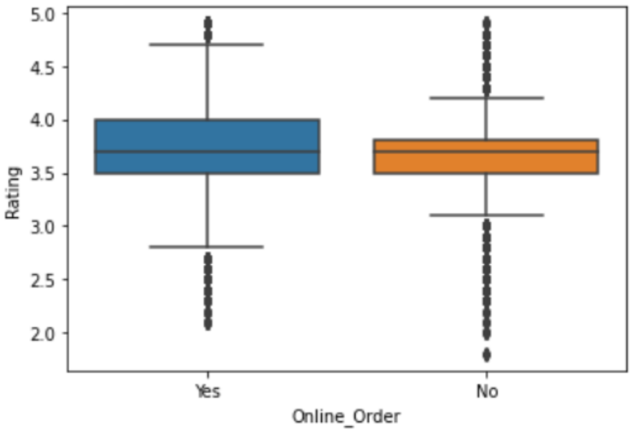


Figure 7: Rating vs restaurants that take/don't take online orders

From Figure 7, it can be inferred that the max rating given to a restaurant taking online orders is higher(4.7) as compared to a restaurant not offering such services (4.3). On the contrary, it is the opposite when the minimum rating is being compared. This could mean that the range of the rating for a restaurant taking online orders is more as it offers more services when compared with the latter thus attracting a wider customer base although the average rating (3.7) for both the categories is the same.

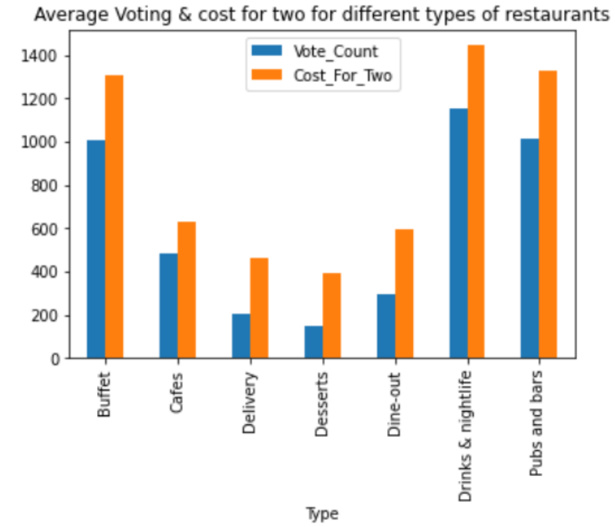


Figure 8: Vote and cost for two for different types of restaurants

Figure 8 shows an analysis between three variables: Type, Cost_For_Two and Vote_Count. From this it can be observed that for the type: ‘Drinks and nightlife’ and ‘Pubs and bars’, the Cost_For_Two is the highest as these restaurants do not charge for the food they serve but for the service and ambience they offer. Also, the overpriced drinks and food help these restaurants to retain higher margins thus fetching them higher profits. Restaurants of the type: ‘Drinks and nightlife’ has the highest Vote_Count as the majority of their customers are ‘Young Adults’ susceptible to digitalization, supported by the quest for social inclusion for which more reviews/votes are expected from them thus, indicating a higher Vote_Count for these restaurant type.

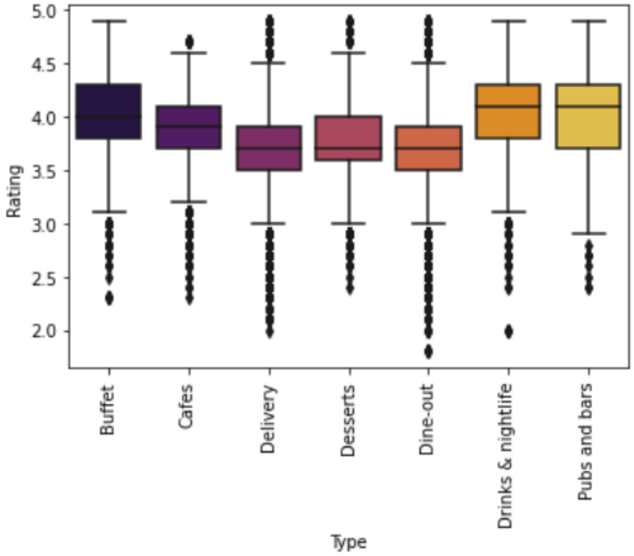


Figure 9: Rating vs type of restaurant

Figure 9 shows the rating for ‘Buffet’, ‘Drinks and nightlife’ and ‘Pubs and bars’ is almost the same. This is because these restaurant type’s major investment is contributed towards providing an experience to its customers from interiors to extensive food and drinks options thus helping them draw better ratings.

2.3 Feature Engineering

For our data, 3 new features namely, Performance, Multiple_Types and Cuisines_Offered were extracted from the existing ones.

2.3.1 Creating new feature: Performance

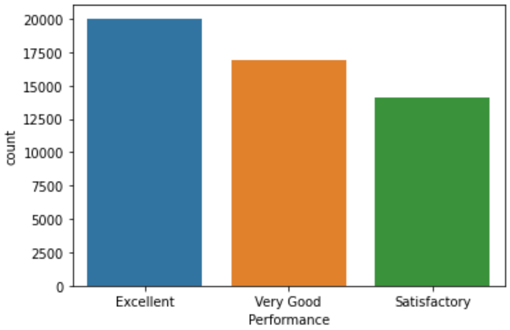


Figure 10: Distribution of target classes

For 'Performance', the existing feature 'Rating' was considered to divide the performance of a restaurant into 3 categories where for restaurants with rating<=3.5 were classified 'Satisfactory' and with rating>=3.8 were classified 'Excellent'. The class 'Very Good' was assigned to restaurants having a rating between 3.5 and 3.8. Figure 10 shows the distribution of the classes so formed.

2.3.2 Creating new feature: Multiple_Types

The restaurant of a type can be coined around multiple names Ex: (Takeaway, Delivery), (Bakery, Dessert Parlour) for which a new feature was introduced 'Multiple_Types' which maintains the count of these names for each restaurant.

2.3.3 Creating new feature: Cuisines_Offered

A restaurant can serve multiple cuisines which is also reflected in the data for which a new feature 'Cuisines_Offered' was introduced which keeps the count of cuisines that every restaurant offers.

2.4 Feature Selection

Out of the current features, Label Encoding was performed on 'Performance' to assign numerical values to the categories created in the previous step where 'Excellent', 'Satisfactory', 'Very Good' are assigned values 0, 1, 2 respectively and stored as a separate feature 'Target'. For features 'Online_Order', 'Book_Table' and 'Type', One Hot Encoding was performed to convert the categorical data to numerical data which adds dummy features to the dataset thus assigning appropriate values to them to make it suitable for model fitting and predictions. In the end, the features 'Restaurant_Name', 'Rating', 'Location', 'Restaurant_Type', 'Cuisines' and 'Performance' were dropped as they were useful to get valuable insights but no longer needed further for predictions.

2.5 Splitting Data into Train and Test

The feature 'Target' is the target variable whose values will be modeled and predicted by other variables and the remaining 15 features are the independent variables which will be used to predict the 'Target'. Stratification in the ratio 3:1 (75:25) was done to ensure proportionate splitting of class labels.

2.6 Feature Scaling

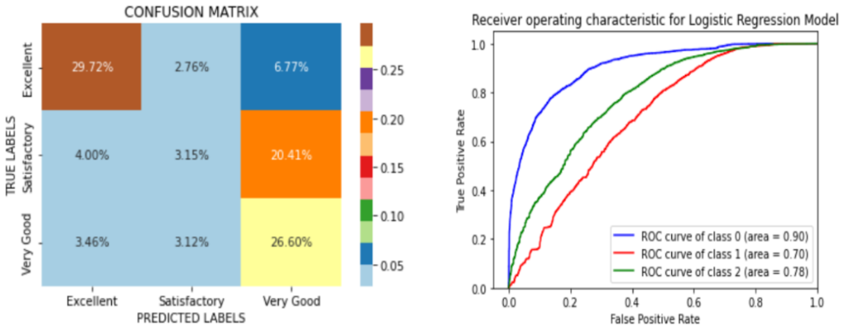
It was observed that of the 15 predictor variables, ‘Vote_Count’ and ‘Cost_For_Two’ had to be normalized and brought to a similar scale with other variables for which feature scaling using StandardScaler was performed.
As a matter of fact, feature scaling does not always result in an improvement in model performance and so models belonging to the class ‘Gradient Descent based’ (Logistic Regression) and ‘Distance based’ (KNN) require feature scaling and models belonging to the class ‘Tree based’ (Decision Tree, Random Forest) do not require feature scaling.[2] The next section discusses the various models used to predict the performance of a restaurant.

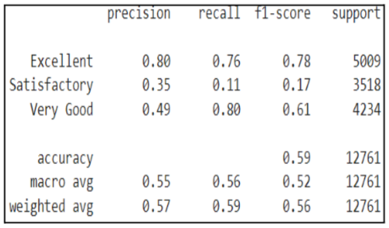
2.7 Modeling

The project stands as a classification problem as it predicts the performance of a new restaurant based on the classes defined earlier. For this various classification models were used starting from the simplest techniques moving on to more complex methods. Also, Hyper- Parameter Tuning using GridSearchCV was performed for every model to gauge the optimum value.

2.7.1 Logistic Regression

In Logistic Regression, the training and testing accuracies 60% turned out to be almost the same which showed that the model neither faced overfitting nor underfitting. But this alone cannot be considered to comment about the performance of a classifier as it doesn’t perform well in case of imbalanced data.





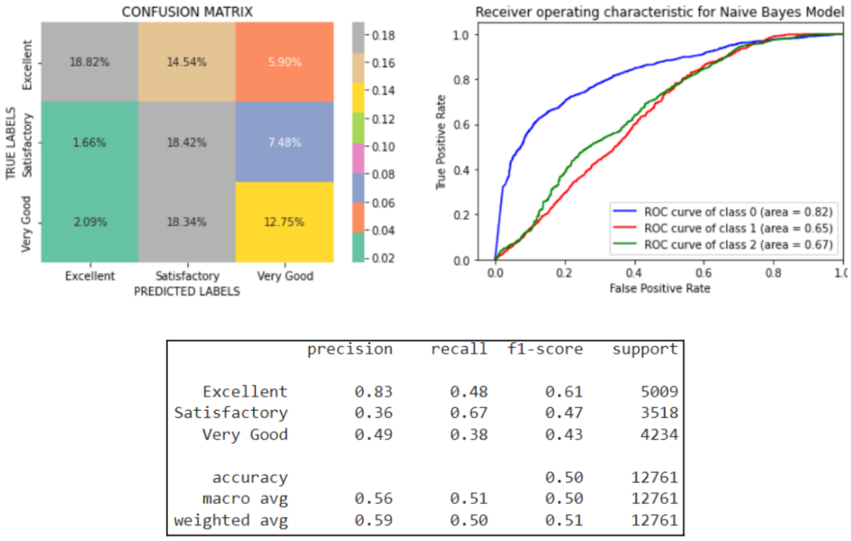
```
              precision    recall  f1-score   support

   Excellent       0.80      0.76      0.78      5009
Satisfactory       0.35      0.11      0.17      3518
   Very Good       0.49      0.80      0.61      4234

    accuracy                           0.59     12761
   macro avg       0.55      0.56      0.52     12761
weighted avg       0.57      0.59      0.56     12761
```

Figure 11: Performance of logistic regression

From the %age of correctly predicted labels of the Confusion Matrix (Figure 11), it was observed that the model was not able to identify much for class ‘Satisfactory’. The ROC curve and so the score shows that the model performs well for class ‘Excellent’ i.e class 0 but not for the other two classes. Also, the accuracy, precision, recall and f1-scores are not that significant, hence we moved to the next classifier for better performance metrics.

2.7.2 Naive Bayes

The Naive Bayes classifier works on the principle of Bayes’ theorem and its main task is to compute the likelihood of an event ‘X’ happening given another event ‘Y’ happens. The training and the testing accuracies are almost 50%.



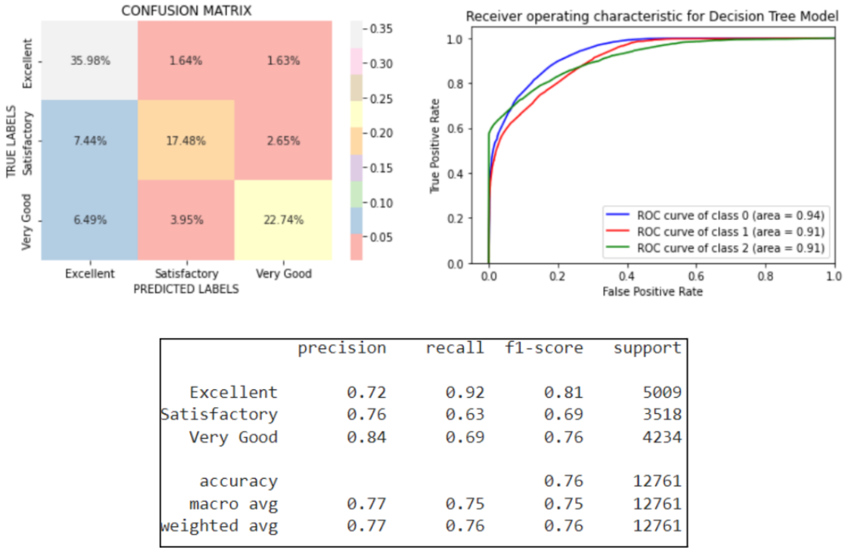
```
              precision    recall  f1-score   support

   Excellent       0.83      0.48      0.61      5009
Satisfactory       0.36      0.67      0.47      3518
   Very Good       0.49      0.38      0.43      4234

    accuracy                           0.50     12761
   macro avg       0.56      0.51      0.50     12761
weighted avg       0.59      0.50      0.51     12761
```

Figure 12: Performance of naive bayes

The Confusion matrix (Figure 12) tells us that the model is not identifying much for class ‘Very Good’.A dip in Accuracy while Precision, Recall and F1-score quite remain the same with all at almost 50% and a good ROC score for only class ‘Excellent’ i.e class 0 was observed, which means the model is not good for the rest classes. This model turns to be discouraging in the sense that it failed to correctly predict the target variables successfully in more than half of the cases. Hence, we move to the next classifier for better performance metrics.

2.7.3 Decision Tree

In the Decision tree classifier, it is observed that the training and testing accuracies are almost the same, hence the model is neither overfitting nor underfitting.



```
              precision    recall  f1-score   support

   Excellent       0.72      0.92      0.81      5009
Satisfactory       0.76      0.63      0.69      3518
   Very Good       0.84      0.69      0.76      4234

    accuracy                           0.76     12761
   macro avg       0.77      0.75      0.75     12761
weighted avg       0.77      0.76      0.76     12761
```

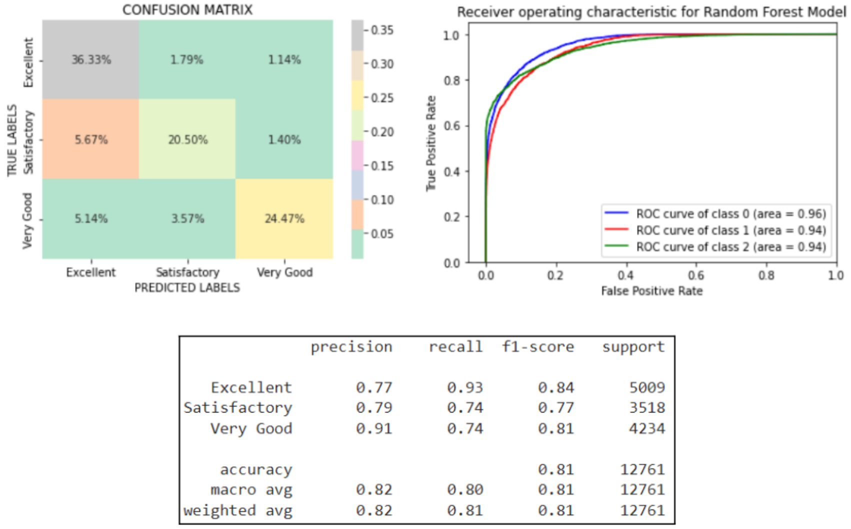
Figure 13: Performance of decision tree

Confusion matrix (in Figure 13) shows that the model identifies each class much better than the above models, however, correct predictions for the classes ‘Satisfactory’ and ‘Very Good’ are still behind as compared to ‘Excellent’. This model is working well for all 3 classes but slightly better for class

0, that could also be because our data is imbalanced having more data points for class 0. Accuracy, Precision, Recall and F1-score are all around 75% and really good ROC scores for all 3 classes were achieved. Hence the model is performing better than the above models. Moving further to check another classifier for better performance metrics.

2.7.4 Random Forest

In Random Forest classifier, the training accuracy is 86% while testing accuracy is 81% which is the best accuracy achieved of all models. The model performed well as the accuracy increased significantly and so it worked significantly well on the unseen test data.



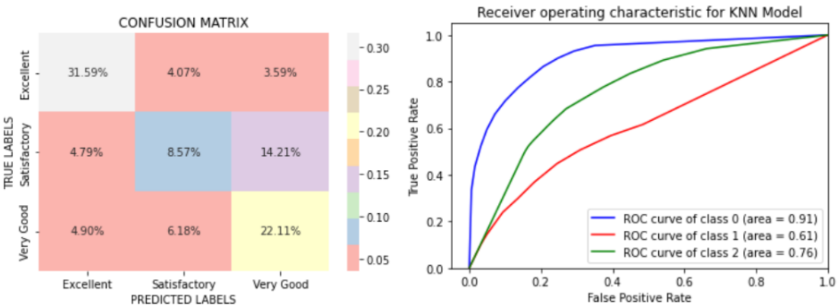
```
              precision    recall  f1-score   support

   Excellent       0.77      0.93      0.84      5009
Satisfactory       0.79      0.74      0.77      3518
   Very Good       0.91      0.74      0.81      4234

    accuracy                           0.81     12761
   macro avg       0.82      0.80      0.81     12761
weighted avg       0.82      0.81      0.81     12761
```

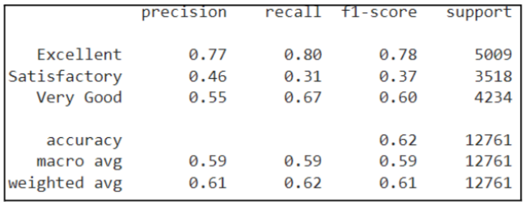
Figure 14: Performance of random forest

The Confusion matrix in (Figure 14) tells us that the model identifies each class much better than previous models. This model is working well for all 3 classes. The performance metric scores so attained were the best of all models i.e. above 80%. Also, the ROC scores for all the 3 classes calculated were the best for this model. With an overall best model, we move to build the next classifier for even better performance metrics.

2.7.5 K-Nearest Neighbour

For this model, the training accuracy is 79% while the testing accuracy is 62% which is a clear case of overfitting on the training set and thus not predicting well on the unseen test data. This could be due to the fact that KNN is sensitive to outliers and doesn't perform well on complex, large datasets.





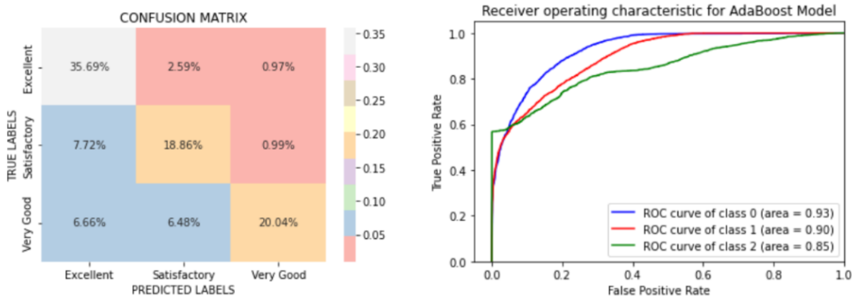
```
              precision    recall  f1-score   support

   Excellent       0.77      0.80      0.78      5009
Satisfactory       0.46      0.31      0.37      3518
   Very Good       0.55      0.67      0.60      4234

    accuracy                           0.62     12761
   macro avg       0.59      0.59      0.59     12761
weighted avg       0.61      0.62      0.61     12761
```

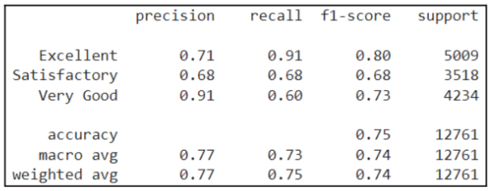
Figure 15: Performance of knn

Confusion matrix (in Figure 15) tells us that the model was not able to sufficiently identify the class ‘Satisfactory’ as it could for the other 2 classes. (Figure 5.3) The ROC-AUC plot for this model tells us that only class ‘Excellent’ i.e class 0 was identified well with class ‘Satisfactory’ i.e. class 1 showing least ROC score. The model does not work well for all the 3 classes. Since, the performance metric scores are also not that significant i.e around 60%, we can say that the model does not work well and so we move further to build the next classifier for better performance metrics.

2.7.6 AdaBoost

For Adaboost Classifier, default weak learners Decision trees are used to build the final strong classifier. The training and testing accuracies for the Adaboost model are nearly 76%, so no problem of overfitting or underfitting was observed.





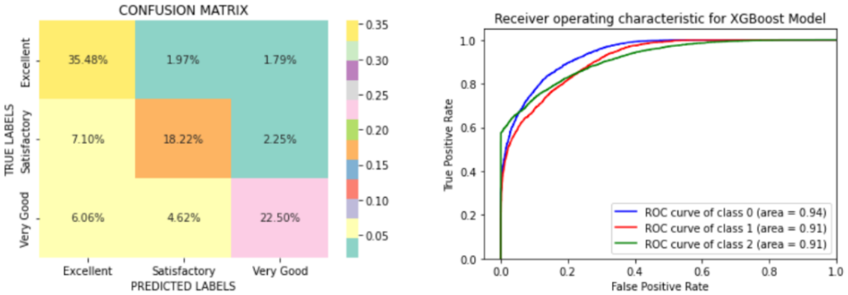
```
              precision    recall  f1-score   support

   Excellent       0.71      0.91      0.80      5009
Satisfactory       0.68      0.68      0.68      3518
   Very Good       0.91      0.60      0.73      4234

    accuracy                           0.75     12761
   macro avg       0.77      0.73      0.74     12761
weighted avg       0.77      0.75      0.74     12761
```

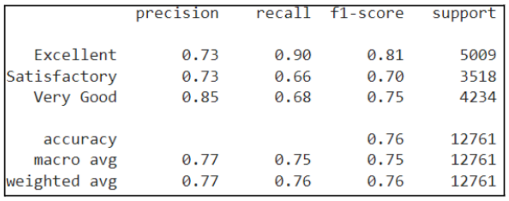
Figure 16: Performance of adaboost

The Confusion matrix (Figure 16) shows that the model is able to identify each class to some extent. All the values of predictive metrics vary between 75%-77%, which isn’t an improvement to the Random Forest classification model. Although the high ROC scores (Figure 5.8) indicates that the model is able to identify all the classes well. Overall, due to lower accuracies and other metrics we move further to build the next classifier for better performance metrics.

2.7.7 XGBoost

XGBoost classifier is the name given to Decision trees that are gradient boosted which can classify the data much faster and performs better than other ensemble methods. The training and testing accuracies for the XGBoost model are nearly 77%, thus indicating no overfitting or underfitting problem. But, in comparison to the previously fitted model (Random Forest), lower accuracies were received which discourages the use of this model.





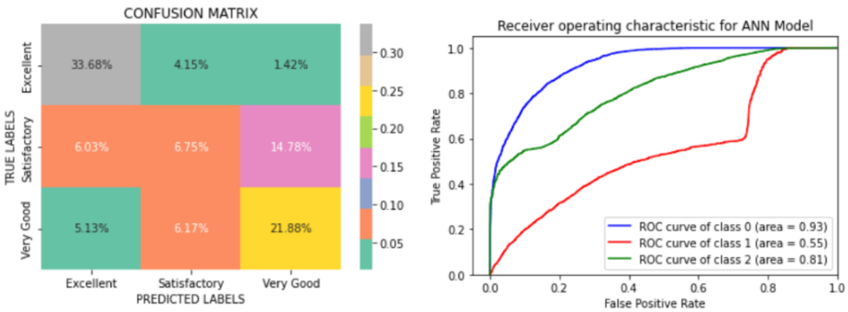
```
              precision    recall  f1-score   support

   Excellent       0.73      0.90      0.81      5009
Satisfactory       0.73      0.66      0.70      3518
   Very Good       0.85      0.68      0.75      4234

    accuracy                           0.76     12761
   macro avg       0.77      0.75      0.75     12761
weighted avg       0.77      0.76      0.76     12761
```

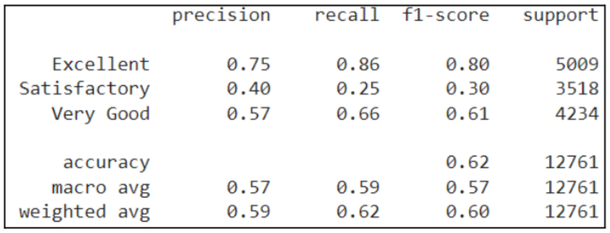
Figure 17: Performance of xgboost

The Confusion matrix(in Figure 17) is indicative of each class where class ‘Satisfactory’ was the least identified. The ROC scores from the ROC-AUC plot shows that it is able to identify all the 3 classes well. All values of the predictive metrics vary between 75%-77%, which also is not improvement to our last best model, Random Forest, thus, moving on to the next model.

2.7.8 Artificial Neural Network

Multilayer Perceptron classifier is the sklearn version of the Artificial neural network. The training accuracy for MLP is 75% while testing accuracy is 62% which is a clear case of overfitting which discourages us from using this model. Since MLP doesn’t perform well on large complex datasets and is sensitive to outliers, overfitting can occur.



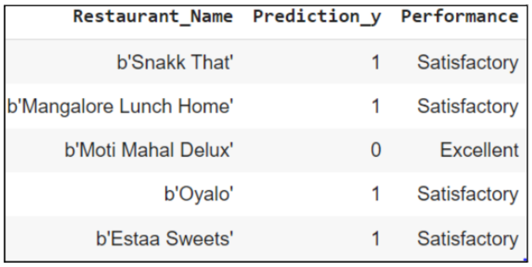


```
              precision    recall  f1-score   support

   Excellent       0.75      0.86      0.80      5009
Satisfactory       0.40      0.25      0.30      3518
   Very Good       0.57      0.66      0.61      4234

    accuracy                           0.62     12761
   macro avg       0.57      0.59      0.57     12761
weighted avg       0.59      0.62      0.60     12761
```

Figure 18: Performance of artificial neural network

The Confusion matrix (Figure 18) tells us that the model is not able to clearly identify class ‘Satisfactory’ i.e. class 1 as compared to other classes. From the ROC- AUC plot it is clear that the model is not able to identify class 1 and so it has the least ROC score (55%). All values of the predictive metrics even for this model fall far behind from those of Random Forest.

3 Results



Restaurant_Name	Prediction_y	Performance
b'Snakk That'	1	Satisfactory
b'Mangalore Lunch Home'	1	Satisfactory
b'Moti Mahal Delux'	0	Excellent
b'Oyalo'	1	Satisfactory
b'Estaa Sweets'	1	Satisfactory

Figure 19: Performance prediction using random forest

As Random Forest model came out to be the best classifier, we further used it to predict the performance of some restaurants that were new to the system and got the results as shown in Figure 19.

4 Conclusion

A total of 8 models were trained out of which, the performance metrics of Random Forest stood tall and so it was finalized for predicting the performance of a new restaurant. With this analysis and implementation it was possible to deliver useful information for Food Business areas and determine the factors affecting them.

References

[1] https://rafaelsilva.com/files/teaching/inf-553-fall-2018/017-predicting-success-upcoming.pdf

[2] https://towardsdatascience.com/what-is-feature-scaling-why-is-it-important-in-machine-learning

[3] https://www.posist.com/restaurant-times/resources/complete-guide-opening-restaurant.html

[4] https://towardsdatascience.com/multiclass-classification-evaluation-with-roc-curves-and-roc-auc

[5] https://machinelearninggeek.com/multi-layer-perceptron-neural-network-using-python