Zomato Restaurant Data Analysis

Anchit Gupta, MT19060
Department of CSE
Indraprastha Institue of Technology
New Delhi, India
anchit19060@iiitd.ac.in

***Abstract*—The past century is said to be an industrial age where much revolutionary discovery is done and this century is said to be the data-driven century where the data is new oil for the business. Similarly in case of the Zomato restaurant data. It contains data about mostly the data about the different restaurant data and their features. This data can give us more insight into the type of food habits of the people. In this project, the main aim is to predict the class of rating which is restaurant is probably getting according to some feature of it like cuisines served etc that are served in different Zomato registered restaurant.**

***Index Terms*—Classification, Feature Engineering**

I. Introduction

Today's world is finding a way to connect and making life easy for people using the internet and services provided on that. Growing the coverage of the internet to the common people and increasing demand and making a business to grow in that platform. Zomato is also one of them, as it's one of the topmost nationwide food delivery systems that is growing day by day. So, every restaurant is trying hard to get top recommendation by getting the highest rating. This project is mainly focused on using different feature engineering topics and choosing what are the best features predicting the rating of the restaurant.

II. Dataset

The dataset used in this project stored the information about the different restaurants having feature names as

- url : contains the data URL of the restaurants to Zomato website
- address : address of the restaurant
- name : name of the restaurant
- netorder : online order facility is available or not
- booktable : book table facility is available or not
- rating : rating of the restaurant
- votes : total number of votes given to the user
- phone : phone number of the restaurant
- location : location name of the restaurant
- resttype : restaurant type
- dishliked : dish people liked the most
- cuisines : different cuisines provided by the restaurant
- cost : approx cost meal for two people
- reviews : recent reviews given by the people
- menuitems : list of menu items served to people
- listedin : type of meal is served by the people
- city : city where the restaurant is present

The dataset contains the 51717 rows and 17 columns where only the votes is the only attribute which is of numerical while other are mixed.

III. Preprocessing

The pre-processing is one of the main steps into getting the insight of the data. As in this case, data is coming or generated for a particular usage generated by the author of this given dataset.

A. Data Cleaning

1) Feature Renaming: Before going into further detail, it is observe that the dataset has feature names which are not well readable and ambiguous for example see the listedin and city both look same as both can refer to the context given that restaurant is listed where which sounds same as the city. But the meaning is totally different so the listed is referring that type of meal served by the restaurant. So the data renaming is done on almost all the dataset columns.

Column Name	Percentage Null	Null Values
url	0.000000	0
address	0.000000	0
name	0.000000	0
netorder	0.000000	0
booktable	0.000000	0
rating	15.033741	7775
votes	0.000000	0
phone	2.335789	1208
location	0.040606	21
resttype	0.438927	227
dishliked	54.291626	28078
cuisines	0.087012	45
cost	0.669026	346
reviews	0.000000	0
menuitems	0.000000	0
listedin	0.000000	0
city	0.000000	0

TABLE I: Dataset Null values

2) Eleminating Null Values: Now we check the set of null values from each attribute present in the dataset and try to find out the which data set is getting most null values in Table-I.

So, the table shows the percentage of null values of different attributes in datset. Here it is clearly visible that the dishliked has more than half values are null and even further. Choose to apply some technique to fill this data will act as the noise to the data. But the rating is data is which can be used in future and

some engineering can be done to fill these null values. While the other is the phone number which is also 2% which is very negligible. As these are some observation which are made by using some tools to find out the to get insight of dataset. But as human intuition, we can remove some columns also i.e. the rows which definitely not have any contribution to predicting the class of the rating. That some are given as: url, address, name, phone, location, dishliked and menuitems. The reason for url ,address, name and phone is that only is that they only nominal values which physically don't have any importance in prediction.

3) Filling missing data present initially dataset: As previously explained the in rating data some values are missing but we can use the fact that the rating is calculated by using reviews giving by the users to the restaurant and fortunately we have also some reviews present from which we can use the rating and assign to the rating feature where null is present. In a simple way, the missing values of rating are filled by mean values of reviews rating present in the review feature column. And in worst that it may happen that both the even review is empty which is to be considered as null then we can use some statistical method to give some random value to those rows only if they are occurring in very small number.

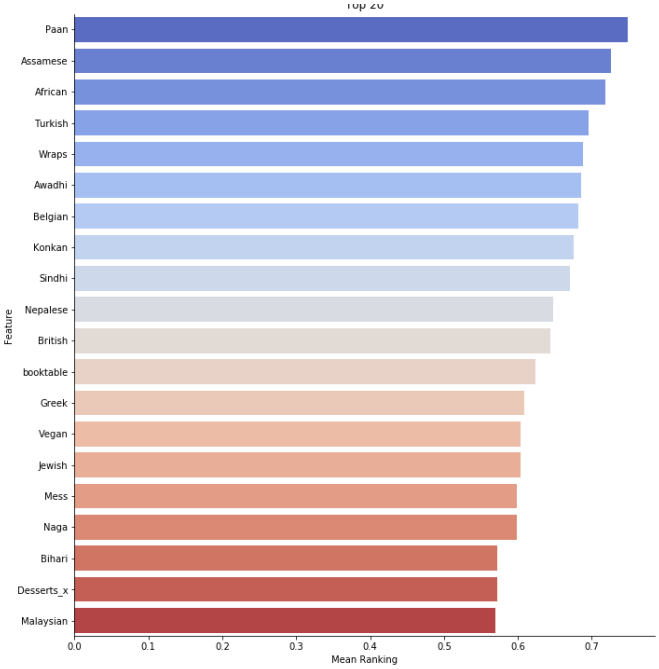

Fig. 1: Top 20 important features according to RFE, Ridge, Lasso, RF

B. Attribute Adjusting

Now the data there are some attributes which are values of the data is present in compound format i.e. the in the cuisines the data is given as (North Indian, Mughlai, Chinese) in one row and they are different for the different rows. Similarily this goes for the columns resttype and listed where values are also present in compound format. So, these columns are removed and replaced by the attributes present in them as asymmetric format. In this way the I got benefit in two ways as I also able to discover more detailed data and also the fact that my nominal values are converted in kind of ordinal values.

Also, in the rating column, there is problem as the rating is present in the out of 5 score formats so I have to extract the rating from it, but the problem is it not directly to remove the rating directly from the rating table initially as there is no proper structure is maintained while creating this dataset.

IV. Feature Engineering

The given dataset after doing preprocessing the number of attributes become around 174 so trying to find the correlation using Pearson correlation matrix is very difficult as it is not visible which are producing the better result as creating the 174 * 174 matrix and trying to visualize the high correlation is very difficult. So, the other methods are used to get the features which are play real importance in the feature importance.Only thing here left is analysis of reviews also but it is not included as this data contains many null values which leads to extinction of 1 out of 4 classes of rating.In this I have used the:

- **Linear Regression**: It works best for the when the data in dataset is quite linear in nature.
- **RFE** (Recursive Feature Elimination): It recursively remove and try to fit the data and train model for the rest attributes.
- **Ridge**: It uses the squared error to impose the penalty on coefficients.
- **Lasso**: It uses statistical methods to increase the prediction accuracy.
- **Random Forest Regression**: It uses the best splitting criteria for the attribute based on Gini index, Entropy and Chi-Squared error.
- **XGBoost Classifier**: It is another tree based classification algorithm specially known for the it's speed and accuracy.

Now using these algorithms we use the feature importance or feature ranking values as they are part of module of python package and taking mean of the corresponding features so that we can get accumulate score (using mean) of the each feature. When mean is obtained the sort them by descending order to get features which are most most important feature for the given data set.

The bottom 20 are shown here in Fig-2 as seen from figure that the mostly have mean score of feature importance $<$ **0.35**, which is threshold for the future use so that we can see if these attributes are removed if there is any improvement in the classification task.

The other algorithm is used to check what are good features used is XGBoost which gives the features based on the f1-score calculated by the result produced by the XGBoost algorithm.

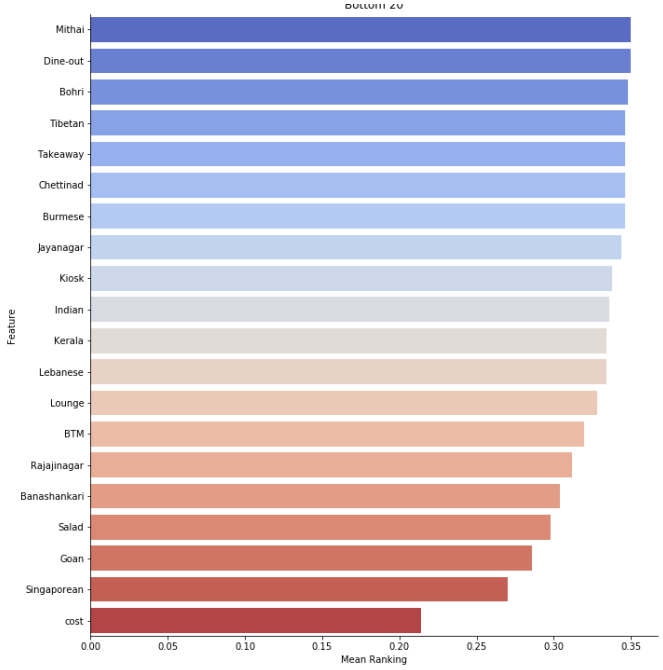

Fig. 2: Worst 20 unimportant features according to RFE, Ridge, Lasso, RF

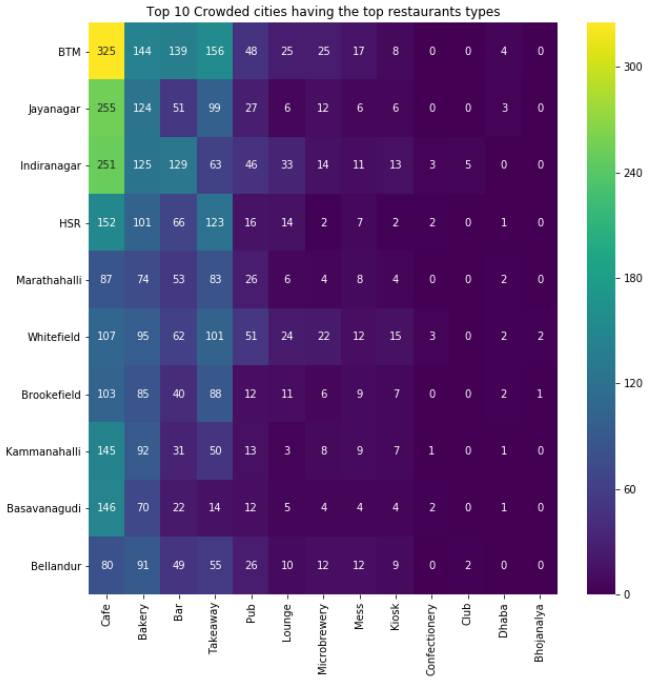

Fig. 3: Top 10 Cities having which Top 10 type of Restaurants

V. Performance of Classifiers

A. Models Used

For this problem I have choosen 4 models to start with:

- Gaussian NB
- Random Forest Classifier
- MLP Classifier
- XGB Classifier [1]

The reason for choosing these models are as initially we want to have to test our data. So, starting with initial simpler model i.e. Naive Bayes. But accuracy from that is not very good it gives us hint that the may be our dataset in non-linear for the comparison other model.So, running the Random Forest Classifier and MLPC Classifier, the surprising result was the Random Forest Classifier which is gives the best accuracy i.e. near 80%-90%. This result gives us the motivation of using the better tree algorithm so XGBoost Classifier is one the most preferred algorithm.

B. Data Used for Models

So, basically we are aiming to compare the three different data's on this which are:

- First one is obtained from original data but removing the less important features which are obtained in by using mean scores described in section Feature Engineering.
- Second is using the original data as it is.
- Third is using the fact that the original dataset is quite imbalance so taking nearly a balanced dataset and testing on it, if it's working better than the others and is class imbalance is reducing our accuracy scores.Nearly, 20 attributes are removed from the data.

Data	Model Name	Accuracy	Precision Score	F1-Score
Original	GaussianNB	0.172043	0.586869	0.207198
Drop. Attribute	GaussianNB	0.165396	0.563887	0.204077
Balanced Class	GaussianNB	0.390330	0.448291	0.355038
Original	RandomForest	0.905670	0.901067	0.899168
Drop. Attribute	RandomForest	0.885239	0.874724	0.876160
Balanced Class	RandomForest	0.712264	0.715472	0.711753
Original	MLP	0.824927	0.803801	0.790369
Drop. Attribute	MLP	0.813001	0.796172	0.780100
Balanced Class	MLP	0.630896	0.639100	0.630410
Original	XGBClassifier	0.813783	0.816041	0.77385
Drop. Attribute	XGBClassifier	0.809286	0.810961	0.769676
Balanced Class	XGBClassifier	0.662736	0.659695	0.651491

TABLE II: Scores of different models on different datasets

VI. Results

So, the data which is obtained with the various models is shown in Table-II. As previously mentioned the models were run in the 3 datasets with 4 models producing accuracy score, precision score and f1-score.

VII. Conclusion

So, the models gives us the inference that the original data is working best on all of the three datasets. Also, the trend that we observe back in feature engineering is that the tree based classifier is working best on the dataset. And the linear model working worst in all scenario as assumed that data is non-linear and should work not well on linear models. While MLP Classifier is good having even more precision on Drop attribute data set than the XGBooster which is our tree model.

REFERENCES

[1] Tianqi Chen,Carlos Guestrin,"XGBoost: A Scalable Tree Boosting System", University of Washington, Published:2006.