PROJECT REPORT

Predicting Star Ratings for Business Review using Yelp Data

Ghorpade, Mrunal Mahesh
mghorp2@uic.edu

Index

Introduction:

Yelp connects its user to many different local businesses as well as provide crowd-sourced reviews and ratings about the same so that they can evaluate that business or service and make a choice. The aim is to predict the Star Rating for a business from previous information such as the text review, review history, users review (Star) rating as well as businesses statistics. This will help other yelp users to make their convenient choice, especially when a person does not have much time to spend on reading the reviews.

Dataset Exploration and Transformations:

Details about the variables of files used in this project are as below;

File Name	Description	Variable Name and Type
Business	Contains business data including location data, attributes, and categories.	Business_ID: String, Name:String, Address:String, City: String, Zip: string, Categories: string, Business hours: dataframe (key days,values), Business Attributes: dataframe (attributes,values)
Review	Contains all information about the reviews	Review_ID: String, User_ID: String, Business_ID: String , Stars:integer, Review text: String , Date: date

Fig1

The files are in JSON format and had nested and hierarchical columns (Categories and Attribute variables). For e.g., in the 'categories' there are three values of "Burgers", "Fast Food", and "Restaurants" at the same level. This is called 'Array', and it's useful to have multiple values assigned to one entity like "business" in this case. The files are imported using "jsonlite" package.

1. Business Data:

We get reviews on Yelp on various business like Automotive, Doctors, Health and Medicine, Restaurants, Local Businesses and many more. The data contains information about 1,56,693 Businesses in total.

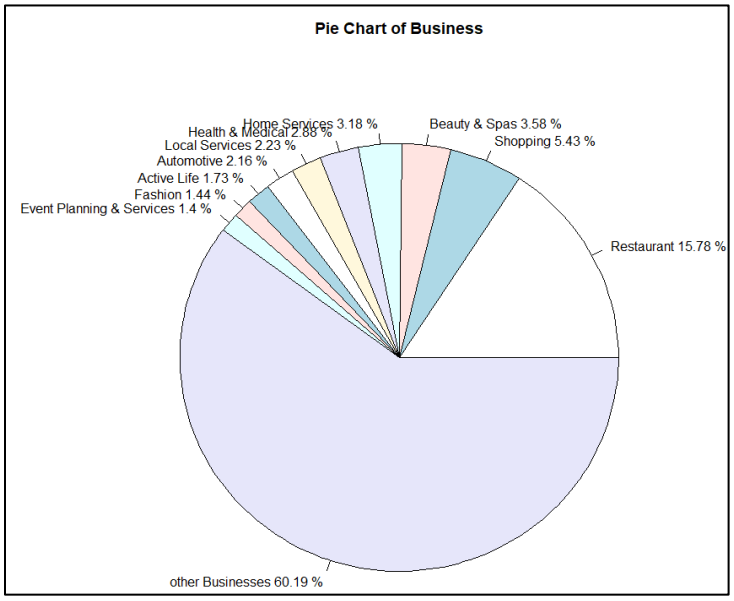
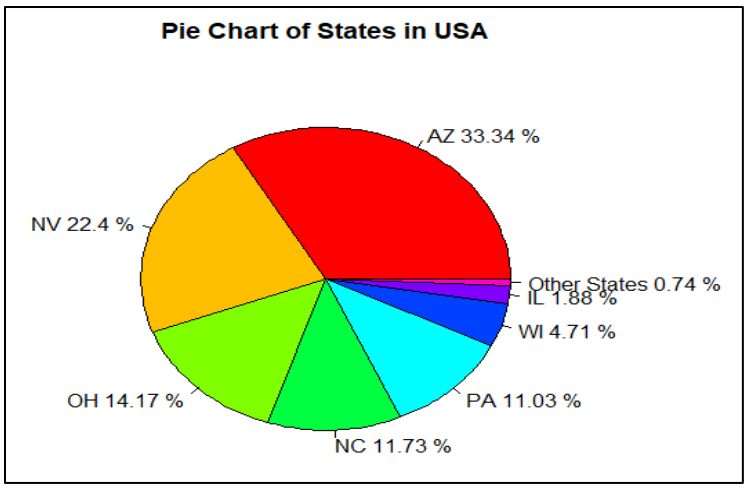


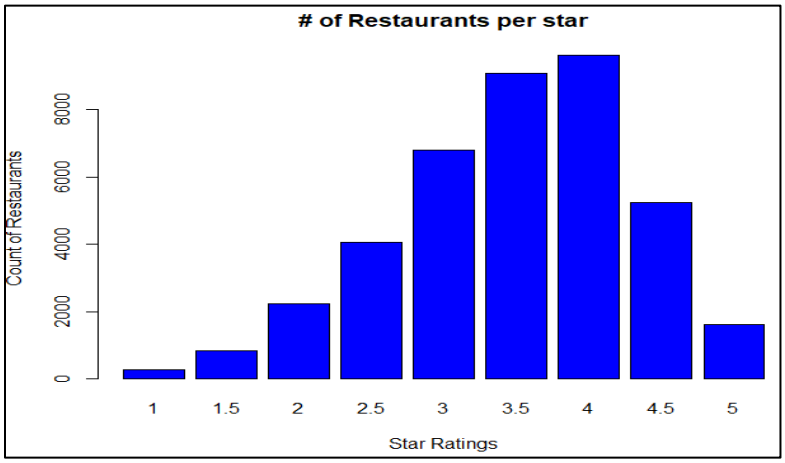
Fig2

It is evident from the Pie chart that Restaurants hold major part amongst all other businesses in the data set. About 15.78% of the total data is Restaurants. Note: the businesses included in the other Businesses section is made combining all the remaining businesses which have less than or equal to 1% of the total businesses. In this project I have focus on the Restaurant Business



The pie chart on the left side shows the distribution of the restaurants present in the given dataset. It is surprising that only 7 states comprise the huge chunk of the data, even though the big states like NY and CA are not present. As the data size is quite large, the data for the state of Illinois will only be used for model building and this model can be replicated across different states

Fig3



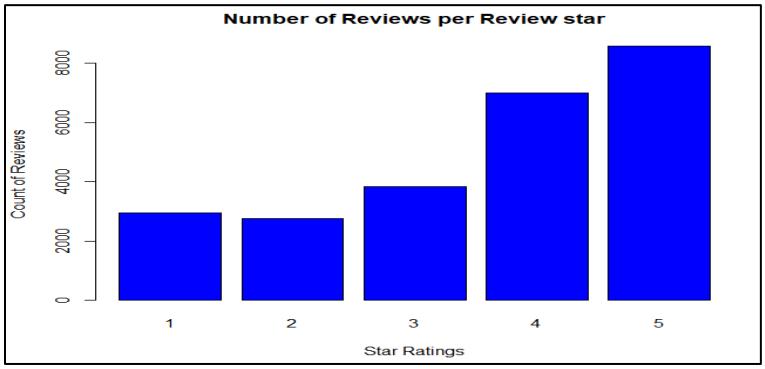
The Bar Plot on the left shows the count of the restaurant per Business Star Ratings. It is a left tailed distribution. there are very few restaurants with less than 2 stars and most of the restaurants are in the range of 3.5 - 4.5 stars ratings.

Fig4

There are about 90 attributes of which only 38 variables had data more than 50%, so these variables were only selected while other variables were discarded. and the null values in these 38 variables where replaced by "unknown" (Note: some of these attributes were binary and have missing values but cannot be directly replaced by False as those attributes values are not known)

2. Reviews Data:

Review details corresponding to the restaurants business are present in this file. There are 4736897 reviews overall. After merging this file with the business data, we get 25110 reviews corresponding to the approximately 800 restaurants in Illinois which has the "text" and "star ratings" of that review.



The graph shows the distribution of restaurants with respect to the review stars to be left skewed. The categories with 4-5 star consists of 70% of the reviews and the rest are distributed amongst 1,2 and 3 star ratings

Fig 5

Since the text is the important part of reviews, word cloud is generated for business ratings 4-5 and another for 1-1.5 which can help us know what are the most frequent words used for the best and the worst ratings for a business. Following are the word clouds (Fig 6(a): For Business Star ratings 4-5, Fig 6(b): For Business Star ratings 1-1.5)

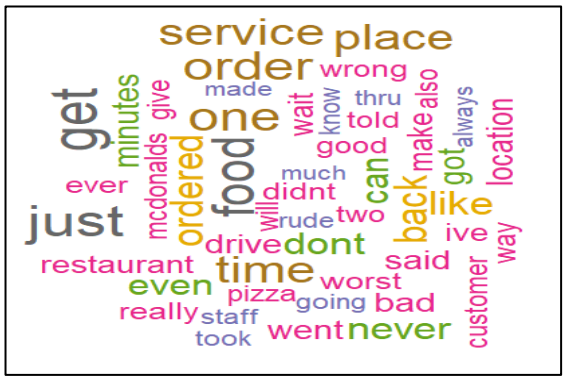


Fig6(a) Fig6(b)

For getting the word clouds, I have calculated frequency of two words at a time. As a single word will not be able to interpret the difference between “good” and “not good”. These two figures show clear tell us why the ratings of particular business were bad or good. For the restaurants with good ratings have the word like “great”, “good”, “amazing”,” best” which gives the positive vibe about that place and the restaurants with low star ratings are represented by words like “worst”, “rude”, “bad”, “never” which gives negative vibe. So, these words can be used to classify positive and negative sentiments of the text.

Sentimental Analysis:

Sentimental Analysis was performed on the reviews columns which contains text. To perform a sentiment analysis, “get_nrc_sentiment” function from “syuzhet” library is used. This function calculates the value of eight different emotions based on the text which is then converted to positive and negative labels which are used as one the predictor variable in the model. The high positive value means review have a positive connotation while negative value suggests negative connotation

Business Rating Prediction:

After merging the reviews data with business data, we get the multiple reviews for the same business. For predicting business ratings different classification models will be used. However as there are multiple rows corresponding to a single business and hence we will get multiple predictions of classes of stars for the same business. So, we choose the star depending upon its probability. The predicted class which has maximum score/probability is considered to as the predicted output of that business. The data is split as 60% training and 40% test for our further analysis.

1. Random Forest Model:

Random forest is basically an algorithm which creates either decision tree or a regression model based on the formula and the data provided to it by the user. Main advantage of random forest is that it avoids overfitting of the training dataset.

Random Forest can be used to find the variables which are significant in model building. To find out the significant variables in the reviews data, Random forest classifier with 500 as number of trees on training data was built. The variables which had importance less than or equal to 1 were removed.

Variables Removed using random forest are as given below:

5	attributes.DietaryRestrictions.vegetarian	1.412860
7	attributes.DietaryRestrictions.soyfree	1.001002
2	attributes.DietaryRestrictions.glutenfree	0.000000
3	attributes.DietaryRestrictions.vegan	0.000000
6	attributes.DietaryRestrictions.kosher	0.000000
8	attributes.DietaryRestrictions.halal	0.000000
9	attributes.DietaryRestrictions.dairyfree	0.000000
14	attributes.HairSpecializesIn.asian	0.000000
27	attributes.AcceptsInsurance	0.000000
45	pos	-1.163180
42	useful	-1.641198

Fig7

The selected variables were then used to generate the random forest model with number of trees =500. The Plot of variable importance using random forest is shown below: We can see that review count, some of the attributes like Restaurant delivery, wi-fi, Alcohol, Parking lots etc., the reviews star, negative sentiment of review and whether the review for that business is cool and funny are important variables in calculating the business star ratings.

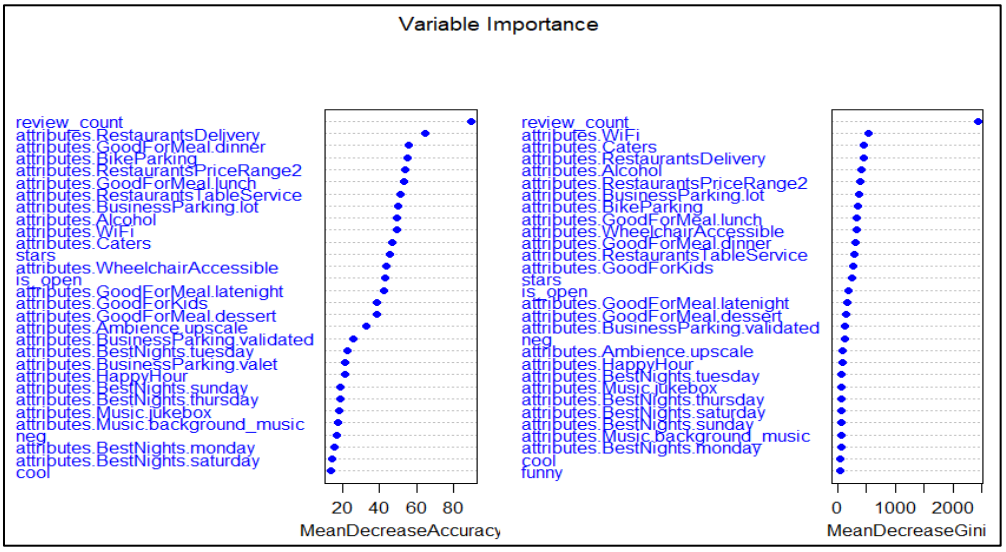


Fig8

The following is the confusion matrix and accuracy on the test data set using random forest.

```
> cm_rf_test
Confusion Matrix and Statistics

          Reference
Prediction   1 1.5   2 2.5   3 3.5   4 4.5   5
       1     7   0   0   0   0   0   0   0   0
       1.5   0  10   0   0   0   0   0   0   0
       2     0   1  46   0   0   1   0   0   0
       2.5   0   1   1  62   2   2   0   0   0
       3     0   1   2   2 134   1   3   0   1
       3.5   0   1   1   3   3 161   3   2   1
       4     0   1   2   4   4   3 139   4   3
       4.5   0   0   0   1   4   2   1  79   6
       5     0   0   0   0   0   0   0   0  16

Overall Statistics

               Accuracy : 0.9071
                 95% CI : (0.8835, 0.9273)
    No Information Rate : 0.2358
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8876
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 1.5 Class: 2 Class: 2.5 Class: 3 Class: 3.5 Class: 4 Class: 4.5 Class: 5
Sensitivity          1.000000    0.66667  0.88462    0.86111   0.9116    0.9471   0.9521    0.9294  0.59259
Specificity          1.000000    1.00000  0.99701    0.99076   0.9826    0.9746   0.9635    0.9780  1.00000
Pos Pred Value       1.000000    1.00000  0.95833    0.91176   0.9306    0.9200   0.8688    0.8495  1.00000
Neg Pred Value       1.000000    0.99297  0.99108    0.98469   0.9775    0.9835   0.9875    0.9904  0.98440
Prevalence           0.009709    0.02080  0.07212    0.09986   0.2039    0.2358   0.2025    0.1179  0.03745
Detection Rate       0.009709    0.01387  0.06380    0.08599   0.1859    0.2233   0.1928    0.1096  0.02219
Detection Prevalence 0.009709    0.01387  0.06657    0.09431   0.1997    0.2427   0.2219    0.1290  0.02219
Balanced Accuracy    1.000000    0.83333  0.94081    0.92593   0.9471    0.9608   0.9578    0.9537  0.79630
```

Fig9

Interpretation: This method gives overall accuracy of 90.71%. We can see from the confusion matrix that the recall rate of class (except star ratings 5 and 1.5) are more than 88%.

2. Support Vector Machine(SVM):

The SVM is one of the machine learning algorithm which solves classification problems using a flexible representation of the class boundaries. By using SVM with kernel functions we can project linearly inseparable data into higher dimensional space where it is linearly separable. It implements automatic complexity control to reduce overfitting. It often has good generalization performance and the same algorithm solves a variety of problems with little tuning.

Following is the summary of SVM model applied to the training data:

```
> summary(train_svm)

Call:
svm(formula = business_stars ~ ., data = review_train, probability = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.02040816

Number of Support Vectors:  9720

 ( 2935 352 2442 1063 2055 70 621 33 149 )


Number of Classes:  9

Levels:
 1 1.5 2 2.5 3 3.5 4 4.5 5
```

Fig 10

The following is the confusion matrix and accuracy on the test data set using SVM.

```
> cm_svm_test
Confusion Matrix and Statistics

          Reference
Prediction   1 1.5   2 2.5   3 3.5   4 4.5   5
       1     0   0   0   0   0   0   0   0   0
       1.5   0   3   0   2   0   0   0   0   0
       2     6   3  23   8   3   1   0   2   0
       2.5   0   3   6  34   4   4   2   0   0
       3     1   1   9  12 102  16  15   3   3
       3.5   0   2   3   5  17 126   8   8   0
       4     0   3   9  11  16  16 112  20  13
       4.5   0   0   2   0   5   7   9  52   6
       5     0   0   0   0   0   0   0   0   5

Overall Statistics

               Accuracy : 0.6338
                 95% CI : (0.5975, 0.6691)
    No Information Rate : 0.2358
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5519
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 1.5 Class: 2 Class: 2.5 Class: 3 Class: 3.5 Class: 4 Class: 4.5 Class: 5
Sensitivity          0.000000   0.200000  0.44231    0.47222   0.6939     0.7412   0.7671     0.61176 0.185185
Specificity          1.000000   0.997167  0.96562    0.97072   0.8955     0.9220   0.8470     0.95440 1.000000
Pos Pred Value            NaN   0.600000  0.50000    0.64151   0.6296     0.7456   0.5600     0.64198 1.000000
Neg Pred Value       0.990291   0.983240  0.95704    0.94311   0.9195     0.9203   0.9347     0.94844 0.969274
Prevalence           0.009709   0.020804  0.07212    0.09986   0.2039     0.2358   0.2025     0.11789 0.037448
Detection Rate       0.000000   0.004161  0.03190    0.04716   0.1415     0.1748   0.1553     0.07212 0.006935
Detection Prevalence 0.000000   0.006935  0.06380    0.07351   0.2247     0.2344   0.2774     0.11234 0.006935
Balanced Accuracy    0.500000   0.598584  0.70396    0.72147   0.7947     0.8316   0.8070     0.78308 0.592593
```

Fig 11

Interpretation: This method gives overall accuracy of 63.38%. The confusion matrix can tell us how many values are correctly and incorrectly predicted. The diagonal elements in the matrix are the correct predictions i.e. For actual 2 star ratings 25 of the business are predicted as 2 star and rest are predicted in the range of 2.5-4.5 which is incorrect. We can calculate overall Accuracy and the error rate of the model using confusion matrix. We can see from the confusion matrix that the recall rate for stars 3-4.5 are above 60%.

3. Extreme Gradient Boosting (XGboost)

XGboost is similar to gradient boosting. (Gradient boosting produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function).
XGboost has both linear model solver and tree learning algorithms. So, what makes it fast is its capacity to do parallel computation on a single machine. This makes xgboost at least **10 times faster** than existing gradient boosting implementations.

Parameters used for performing XGBoost models are:

Parameters	Value
Booster	gbtree
maximum depth	6
Learning Rate	1
Maximum number of iteration	200

The following is the confusion matrix and accuracy on the test data set using Gradient Boosting Algorithm.

```
> cm_gbm_test
Confusion Matrix and Statistics

          Reference
Prediction   1 1.5   2 2.5   3 3.5   4 4.5   5
       1     7   0   0   0   0   0   0   0   0
       1.5   0  13   1   0   1   0   0   0   0
       2     0   0  47   1   0   1   0   0   0
       2.5   0   0   0  64   0   1   0   0   0
       3     0   1   2   1 142   1   0   1   1
       3.5   0   0   0   1   0 162   1   0   0
       4     0   0   1   3   2   2 144   3   0
       4.5   0   1   1   1   0   3   1  81   2
       5     0   0   0   1   2   0   0   0  24

Overall Statistics

               Accuracy : 0.9487
                 95% CI : (0.93, 0.9636)
    No Information Rate : 0.2358
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9382
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 1.5 Class: 2 Class: 2.5 Class: 3 Class: 3.5 Class: 4 Class: 4.5 Class: 5
Sensitivity          1.000000    0.86667  0.90385    0.88889   0.9660     0.9529   0.9863     0.9529  0.88889
Specificity          1.000000    0.99717  0.99701    0.99846   0.9878     0.9964   0.9809     0.9858  0.99568
Pos Pred Value       1.000000    0.86667  0.95918    0.98462   0.9530     0.9878   0.9290     0.9000  0.88889
Neg Pred Value       1.000000    0.99717  0.99256    0.98780   0.9913     0.9856   0.9965     0.9937  0.99568
Prevalence           0.009709    0.02080  0.07212    0.09986   0.2039     0.2358   0.2025     0.1179  0.03745
Detection Rate       0.009709    0.01803  0.06519    0.08877   0.1969     0.2247   0.1997     0.1123  0.03329
Detection Prevalence 0.009709    0.02080  0.06796    0.09015   0.2067     0.2275   0.2150     0.1248  0.03745
Balanced Accuracy    1.000000    0.93192  0.95043    0.94367   0.9769     0.9747   0.9836     0.9694  0.94228
```

Fig 12

Interpretation: This method gives overall accuracy of 94.87%. The confusion matrix can tell us how many values are correctly and incorrectly predicted. The diagonal elements in the matrix are the correct predictions i.e. For actual 2 star ratings 47 of the business are predicted as 2 star and rest are predicted in the range of 3-4.5 which is incorrect. We can calculate overall Accuracy and the error rate of the model using confusion matrix. We can see from the confusion matrix that the recall rate for all the stars is above 85%.

Insights:

- XGBoost gives the maximum accuracy and recall rate than Random Forest and SVM for predicting business star ratings for this dataset.
- Naïve Bayes was performed on the data set but it did not give good results and hence is not included in the Project Report. (Please Refer to code Review data).

Future Work:

Due to the limited time, I had no chance to try to develop more models about text mining, such as TF-IDF and Predicting Review stars for each review and using that as an input variable for predicting the business stars and to see if this increases the accuracy of the results.

Conclusion:

One of the most important learnings, derived out of this exercise is that raw data is always going to contain quality issues like missing values, outliers etc. Hence, cleaning the data becomes very significant as models will depend upon how we handle these data redundancies. If the data issues are not handled properly, then there will be inaccuracies in the model and this could lead to incorrect analysis of the data.

After preprocessing the data, some text mining techniques such as sentiment analysis to build important features. Different classification techniques were applied to identify the business star ratings of the restaurants in Illinois. This can be applied to the whole datasets containing various other business. These results help us in predicting the future values if required by extrapolating the model and gives us more insights into identifying why certain restaurants have good ratings or to know what measures should be taken to improve their star rating.

Appendix:

No	Figure Number	Description
1	Fig1	Data Description
2	Fig2	Pie Chart of Businesses
3	Fig3	Restaurant Distribution in Yelp Dataset – State wise
4	Fig4	Star Rating Wise Distribution - Restaurants
5	Fig5	Frequency Distribution of Review stars
6	Fig6	Word Cloud for Review data (a) for 4-5 star ratings (b) for 1-1.5 star ratings
7	Fig7	Variables Removed using Random Forest Variable Importance
8	Fig8	Variable Importance for Random Forest
9	Fig9	Confusion Matrix for Random Forest
10	Fig10	Summary Results for Support Vector Machine
11	Fig11	Confusion Matrix and Accuracy Support Vector Machine
12	Fig12	Confusion Matrix and Accuracy XGBoost