# restaurantAnalysis

*Ji Won Chung*

*12/5/2017*

## Purpose of Exploring Restaurants of the Business Dataset

Our final project does a two part analysis: one on the user dataset and its corresponding reviews and another on the business dataset. Our final goal is to find if there is any overlap in what features businesses consider important and users consider important. We focused on restaurants information from the business dataset because we believed that Yelp reviews were generally written for restaurants and thought it was an interesting subset to look at. In addition, we hypothesized the user reviews would mainly be about restaurants and food. We try to identify what components define a "successful" restaurant business by exploring the business dataset which this file demonstrates. Our goal for exploring the restaurant business data is to use two types of models: decision trees and random forests to see if we can identify what are some potential factors that define a "successful" business. We planned to do this by identifying potential patterns that appear from the decision tree and random forest models. For example, we hypothesized that some factors may be indicated as factors of high importance for both the decision tree and random forests. If so, we further reasoned that those factors could potentially define what makes a restaurant business successful.

### Explanation of Successful Business Criteria

"is_open" demonstrates whether the business is still in business or not. "is_open" has two labels: 'Opened' and 'Closed'. 'Opened' indicates that the business is still open and 'Closed' indicates otherwise. "stars" is the star ratings of the business data. "review_count" is the number of reviews written for the business.

## Original Dataset Import & Summary

These are the number of observations in the original dataset

```
print(nObservations)
```

```
## [1] 63151
```

These are the columns in the original dataset

```
print(colnames(restaurantDf))
```

```
##  [1] "X"                         "index"
##  [3] "business_id"               "city"
##  [5] "is_open"                   "neighborhood"
##  [7] "review_count"              "stars"
##  [9] "state"                     "BusinessAcceptsCreditCards"
## [11] "RestaurantsPriceRange2"    "RestaurantsTakeOut"
## [13] "Parking"
```

# Changing Data Types & Reordering Factors

## Changing Factors to Numeric Data

"stars" and "review_count" are factor types in R initially. However, they are numbers on an ordinal scale and have a numeric ordering. Thus, we convert the data types of those columns to numeric.

```
restaurantDf$stars <- as.numeric(restaurantDf$stars)
restaurantDf$review_count <- as.numeric(restaurantDf$review_count)
```

# Removing Columns

We removed unnecessary columns "X" and "index" which are remnants of the preprocessing data in python. We also removed "business_id", "city", "neighborhood", and "state" because we were not interested in such fine grain information nor regional information given that our observation set was small.

```
cleanDf <- subset(restaurantDf, select = -c(X, index, business_id, city, neighborhood, state))
head(cleanDf)
```

```
##   is_open review_count stars BusinessAcceptsCreditCards
## 1  Closed            4   4.5                      False
## 2  Closed           10   4.5                       True
## 3    Open           21   2.0                       True
## 4    Open            3   3.0                    Unknown
## 5    Open           15   3.0                       True
## 6    Open            6   2.5                       True
##   RestaurantsPriceRange2 RestaurantsTakeOut Parking
## 1                    2.0               True   False
## 2                    1.0               True   False
## 3                    2.0               True   False
## 4                Unknown            Unknown   False
## 5                    1.0               True    True
## 6                    2.0               True   False
```

# Creating a Decision Tree

For each of the decision tree models we made two plots. One using rpart.plot and the other using the tree library. The initial reason was the fact that the text() function would not work. However, we found out that it does work if we run the programs in chunks. However, we left both models because it was interesting to see slightly different models in some of the cases and some gave more statistics than others.

The restaurantsOpen model shows indiciates that if there is data on RestaurantsTakeOut or when "RestaurantsTakeOut = Unknown" is "No", 85% of the time the restaurants are open. This is an interesting find because we did not imagine that the fact that there existed information on take out could be a potential predictor for whether the restaurant was open or not. If the restaurant had no information on take out and parking was either unknown or unavailable, then 12% of the time the restaurants seemed to be open. Else, they were closed. Perhaps this indicated that if parking in fact had no affect on a successful restaurant, because 12% of the restaurants would be open regardless of whether they had parking or not. Perhaps the 4% of the restaurants that were closed because they had parking occurred because they spent too much money on parking and failed to keep their finances afloat. Using the rpart model we were unable to determine what the training error rate was so we decided to do a tree model.

```
tree_Open = tree(is_open~ ., data = restaurant_train)
print(summary(tree_Open))
```

```
##
## Classification tree:
## tree(formula = is_open ~ ., data = restaurant_train)
## Variables actually used in tree construction:
## [1] "RestaurantsTakeOut"
## Number of terminal nodes:  2
## Residual mean deviance:  1.076 = 47560 / 44200
## Misclassification error rate: 0.239 = 10566 / 44206
```

```
tree_Open
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 44206 48620 Open ( 0.2390 0.7610 )
##   2) RestaurantsTakeOut: Unknown 6800  9162 Open ( 0.4016 0.5984 ) *
##   3) RestaurantsTakeOut: False,True 37406 38400 Open ( 0.2095 0.7905 ) *
```

```
plot(tree_Open)
text(tree_Open, pretty = 0)
```

RestaurantsTakeOut: Unknown
Open
Open

```
tree_pred = predict(tree_Open, restaurant_test, type = "class")
table(tree_pred, restaurant_test$is_open)
```

```
##
## tree_pred Closed Open
##    Closed      0    0
##    Open      891 1939
```

```
print((0+1939)/(nrow(restaurant_test)))
```

```
## [1] 0.685159
```

We wondered if we could get a better model by taking out the “RestaurantsTakeOut” as a predictor. The tree now has a review_count as the splitter, but this too is a weird model because it predicts that the restaurants are open regardless of which direciton the tree splits. The training error is ~24% the approach leads to ~69% of correct predictions for the test data set. However, we do not think this is significant given the incomprehensible model.

The rpart model in this case is the same as the tree model. This model only has one node which raises doubts as to whether this is a good model. However, it seems to indicate that the availability of parking determines the star ratings. It seems that if there is no parking 50% of the data have a star rating of less than 3.5. If there is no information on Parking or if Parking exists, then half the data seems to have a rating of higher than 3.6. This is still interesting because it reflects that no Parking leads to low star ratings. This makes sense becaue a lot of people probably will not go to a restaurant if it does not have parking available.

The rpart model in this case is the same as the tree model as well. The model has four terminal nodes and three splitting categories: "Parking", "is_open", and "RestaurantsPriceRange2". The splitting categories intuitively make sense as to why they would determine review counts. If there was no parking, then half the restaurants had less than 51 reviews (or an average of 15). Else, if there were no information or there was parking, whether the restaurant was open or closed determined the number of reviews. If the restaurant was closed than 12% of the restaurants had less than 87 reviews (or an average of 47). Else, if the restaurant was Open, then the the Restaurants had around 101 reviews or more than 87 reviews. If the RestaurantsPriceRange2 was Unknown or on the low 1.0 range, then 14% of the restaurants had less than 101 reviews. If it was more than 1.0 price range then 23% of the time the restaurants had more than 101 reviews (or an average of 120 reviews). This seems to show that restaurants that potentially have parking, are open, and have a "high" or above 1.0 price range that they have more reviews. This intuitively makes sense because restaurants that are successful probably have parking information, are still open, and have a "pricey" range.

We used sqrt(p) variables or approximately 3 mtry's because this was a random forest of classification trees. We tested it on the entire dataset, cleanDf. The results indicate that across all of the trees considered in the random forest, information on RestaurantsTakeOut is by far the most important variables. This reflects what we saw in our rpart model for the decision tree.

The results indicate that across all of the trees considered in the random forest, information on BusinessAcceptsCreditCards is one of the most important variables. It actually is unclear what variable was important. There does not seem to be a variable that is strikingly important which seems to also reflect the bad model seen from our decision tree earlier. However, it does seem that Parking is never the bottom 2 importance variables.

The results indicate that across all of the trees considered in the random forest, information on Parking and RestaurntsPriceRange2 are by far the most important variables. This reflects what we saw in our decision tree earlier, so it strengthens our case that Parking and Price Range is important for determining the number of review counts of a business. The "is_open" category is also never the bottom two, so perhaps that is significant as well.