# 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.

---

## Defining as Successful Restaurant

We recognize that success can be determined by various categories. In our case we determined that the success of a business based on the yelp dataset could be objectively determined by "is_open", "stars", and "review_count". Our assumption was that successful business most likely is still open, has good star ratings, and a lot of review counts.

---

### 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.

---

## Import Necessary Packages

```
library(tree)
library(dplyr)
library(ggplot2)
library(randomForest)
library(rpart.plot)
library(rpart)
```

---

## #Original Dataset Import & Summary

```
print(summary(restaurantDf))
```

```
##        X             index                             business_id   
##  Min.   :    0   Min.   :     1   __8j8yhsmE98wNWHJNyAgw:    1  
##  1st Qu.:15788   1st Qu.: 39002   __aKnGBedQ51_hEc3D9ARw:    1  
##  Median :31575   Median : 78134   __bqGGnOjtY9eEhrZAUsgA:    1  
##  Mean   :31575   Mean   : 78217   __eb2f_wEBrEl0xCyLqDeQ:    1  
##  3rd Qu.:47362   3rd Qu.:117444   __fMLrmv9M1_W4kBvR2VnQ:    1  
##  Max.   :63150   Max.   :156635   __fyRzU8kL6HkVV3wgxfmQ:    1  
##                                   (Other)               :63145  
##          city         is_open                    neighborhood  
##  Toronto   : 8542   Closed:15091   Unknown           :34236  
##  Las Vegas : 7040   Open  :48060   Ville-Marie       : 1458  
##  Phoenix   : 4361                  Scarborough       : 1180  
##  Montréal  : 3706                  The Strip         : 1058  
##  Charlotte : 2882                  Plateau-Mont-Royal:  976  
##  Pittsburgh: 2642                  Southeast         :  911  
##  (Other)   :33978                  (Other)           :23332  
##   review_count         stars           state      
##  Min.   :   3.00   Min.   :1.00   ON     :15532  
##  1st Qu.:   6.00   1st Qu.:3.00   AZ     :12618  
##  Median :  15.00   Median :3.50   NV     : 8556  
##  Mean   :  50.38   Mean   :3.51   OH     : 5454  
##  3rd Qu.:  45.00   3rd Qu.:4.00   QC     : 5380  
##  Max.   :6979.00   Max.   :5.00   NC     : 4477  
##                                   (Other):11134  
##  BusinessAcceptsCreditCards RestaurantsPriceRange2 RestaurantsTakeOut
##  False  : 3252              1.0    :23893          False  : 4815     
##  True   :52633              2.0    :30064          True   :48694     
##  Unknown: 7266              3.0    : 3551          Unknown: 9642     
##                             4.0    :  631                            
##                             Unknown: 5012                            
##                                                                      
## 
```

```
##    Parking
##   False:31663
##   True :31488
##
```

## Analysis of the Summary

Something that is apparent from the summary is that data regarding regional information have too many different labels. We can also see that we have 63150 observations which is not necessarily a lot of data. Originally we were going to subset the dataset by region (for example by states) and run models on the individual regions. However, since the business dataset with only restaurants is small, we decided to run models on the entire restaurant dataset instead.

```
restaurantDf = read.csv('restaurantInfo.csv', header = TRUE)
print("This is the summary information of the original dataset...")
print(summary(restaurantDf))
nObservations = nrow(restaurantDf)
nFactors = ncol(restaurantDf)
```

---

These are the number of observations in the original dataset

```
print(nObservations)
```

---

These are the number of columns in the original dataset

```
print(nFactors)
```

---

These are the columns in the original dataset

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

---

For reference, we have set the seed to 1.

```
set.seed(1)
```

---

# 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)
```

---

## Setting Ordering to Factors

The columns "BusinessAcceptsCreditCards", "RestaurantsPriceRange2", and "RestaurantsTakeOut" can be considered categorical data types. We extracted the columns above by manually looking through the labels of each columns. However, we examine below that some of the orderings of the factors do not make sense. We arbitrarly ordered the factors from "Unknown" to "negative" to a more "positive" ranking. "Unknown" indicating that we did not have information on the data. "negative" indicating something that we presumed to not be in favor of the customer. "positive" indicating something in favor for the customer. However, this ranking is somewhat subjective. Hence, we explain this part by part for each of the columns.

```
print(levels(restaurantDf$RestaurantsPriceRange2 ))
restaurantDf$RestaurantsPriceRange2<- factor(restaurantDf$RestaurantsPriceRange2,
levels = c("Unknown", "1.0", "2.0", "3.0", "4.0"))
print(levels(restaurantDf$RestaurantsPriceRange2))

print(levels(restaurantDf$BusinessAcceptsCreditCards ))
restaurantDf$BusinessAcceptsCreditCards<-
factor(restaurantDf$BusinessAcceptsCreditCards, levels = c("Unknown", "False",
"True"))
print(levels(restaurantDf$BusinessAcceptsCreditCards ))


print(levels(restaurantDf$RestaurantsTakeOut ))
restaurantDf$RestaurantsTakeOut<- factor(restaurantDf$RestaurantsTakeOut, levels =
c("Unknown", "False", "True"))
print(levels(restaurantDf$RestaurantsTakeOut ))
```

---

## 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)
```

---

## Taking the subset of the data

We took 70% of the dataset for the training model and 30% for the test model.

```
restaurant_train = cleanDf %>%
  sample_frac(.7)

restaurant_test = cleanDf %>%
  setdiff(restaurant_train)
```

---

## 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.

---

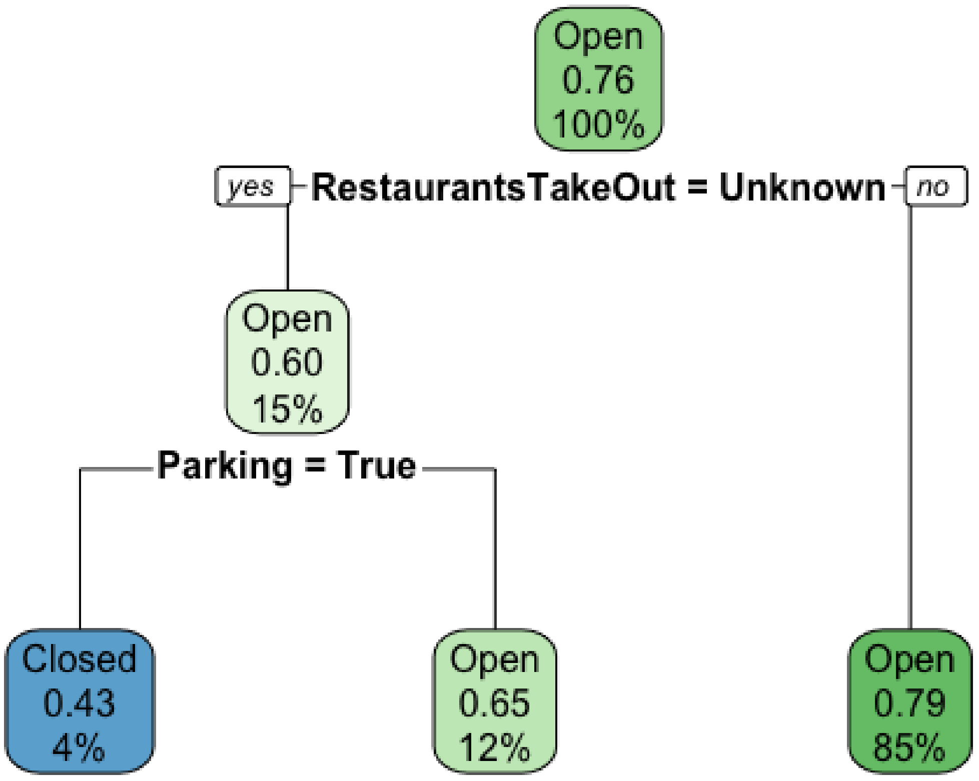
### Predicting if the Restaurant is Open using a Classification Tree

We created a classification tree that would predict if the restaurant was still in business or not ("is_open" column) using rpart.

```
restaurantsOpen <- rpart(is_open ~ . , data = restaurant_train, method = "class")
```

```
rpart.plot(restaurantsOpen)
```

---



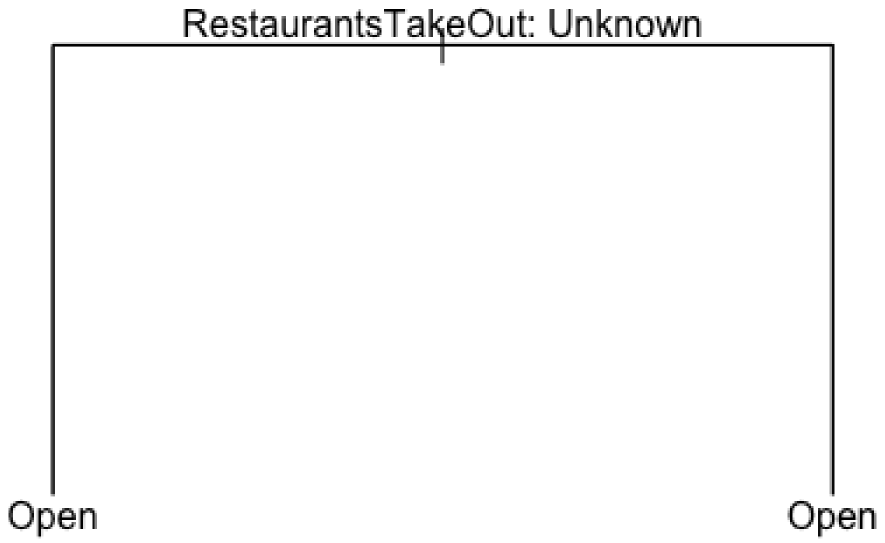
The restaurantsOpen model shows indiciates that if there is data on RestaurantsTakeOut or when "RestaurantsTakeOut = Unknown" is "No", 85% of the time the restaurants are open. This is an interesting find because we did not imagine that the fact that there existed information on take out could be a potential 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))
tree_Open
plot(tree_Open)
```

```
text(tree_Open, pretty = 0)

tree_pred = predict(tree_Open, restaurant_test, type = "class")
table(tree_pred, restaurant_test$is_open)
print((0+1939)/(nrow(restaurant_test)))
```

---

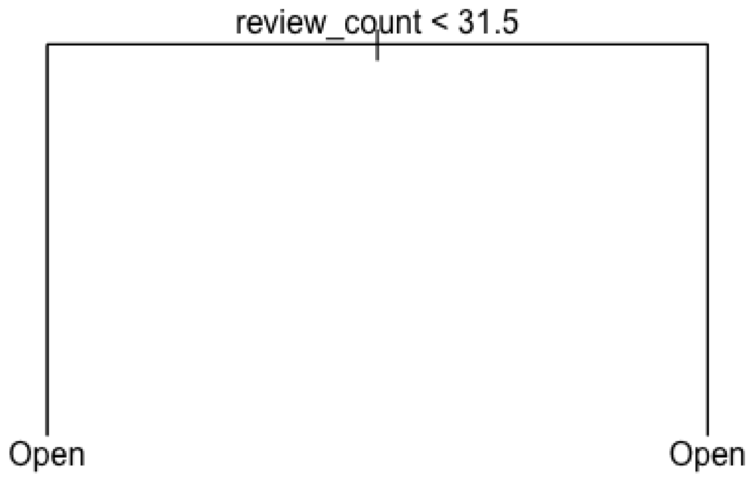


The model created by the tree model seemed odd because it denoted that the restaurant would always be open regardless of the "RestaurantsTakeOut" info. It was a somewhat different model from the rpart model, but this model also decided that "RestaurantsTakeOut" was importing in splitting the data. The training model has a 23.9% error rate and the approach leads to ~67.5% of correct predictions for the test data set. However, we do not think this is significant given the incomprehensible model.

```
tree_OpenWithoutTakeOut = tree(is_open~ .-RestaurantsTakeOut, data =
restaurant_train)
print(summary(tree_OpenWithoutTakeOut))
tree_OpenWithoutTakeOut
plot(tree_OpenWithoutTakeOut)
text(tree_OpenWithoutTakeOut, pretty = 0)

tree_predWithoutTakeOut = predict(tree_OpenWithoutTakeOut, restaurant_test, type =
"class")
table(tree_predWithoutTakeOut, restaurant_test$is_open)
print((0+1995)/(nrow(restaurant_test)))
```

---



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.

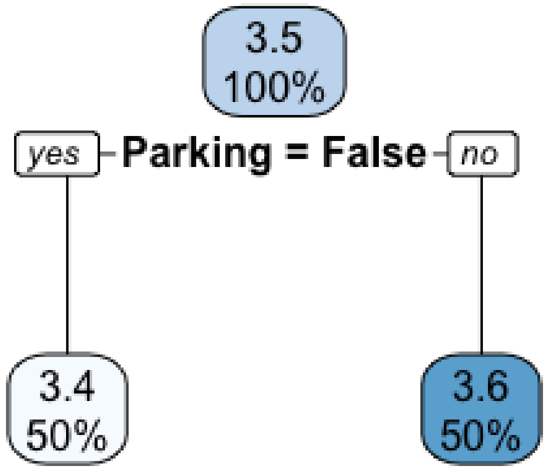
---

## Predicting Star Reviews of Restaurants using a Regression Tree

```
restaurantsStars <- rpart(stars ~ . , data = restaurant_train, method = "anova")
rpart.plot(restaurantsStars)


tree_stars = tree(stars~., data = restaurant_train)
print(summary(tree_stars))
tree_stars
plot(tree_stars)
text(tree_stars,  pretty = 0)
```

---



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.

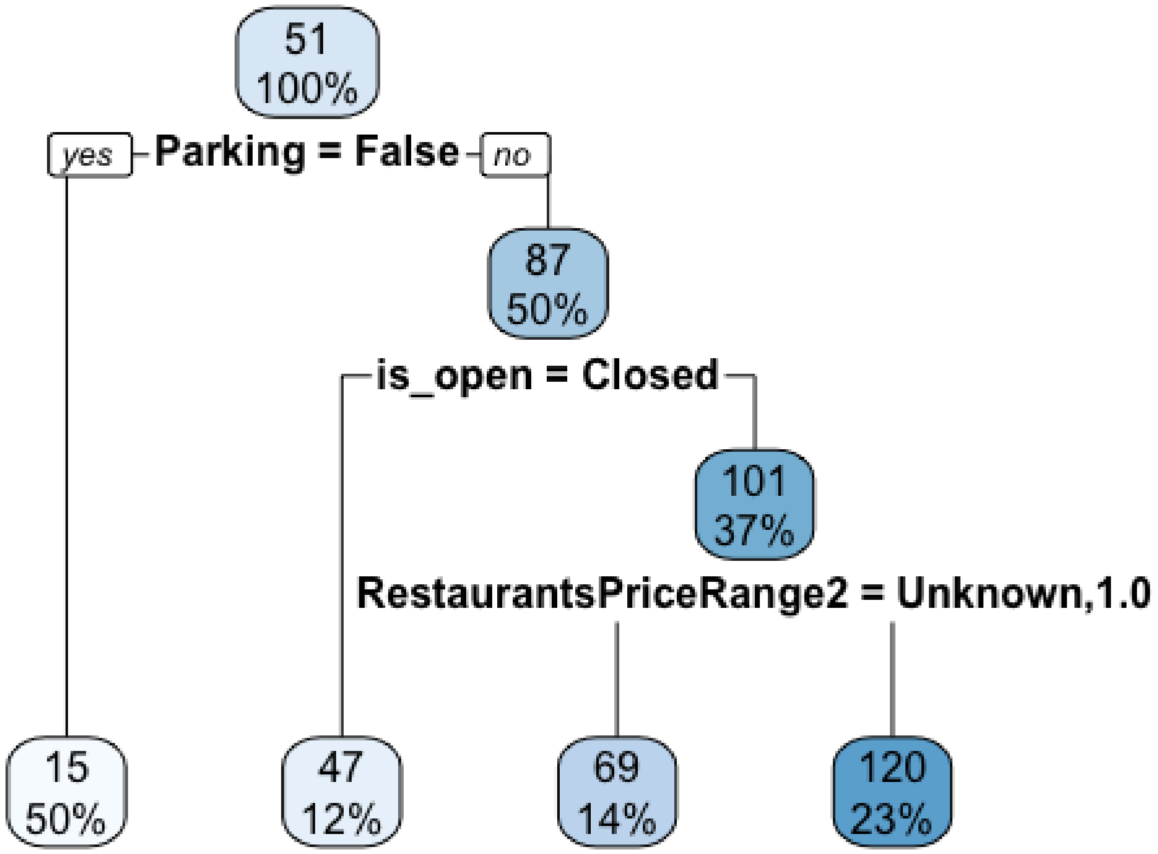
---

## Predicting ReviewCounts of Restaurants using a Regression Tree

```
restaurantReviews <- rpart(review_count ~ . , data = restaurant_train, method =
"anova")
rpart.plot(restaurantReviews)


tree_rc = tree(review_count~., data = restaurant_train)
print(summary(tree_rc))
tree_rc
plot(tree_rc)
text(tree_rc,  pretty = 0)
```

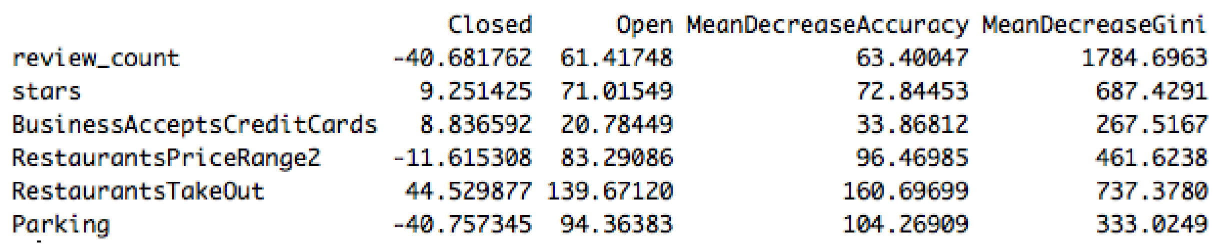
---



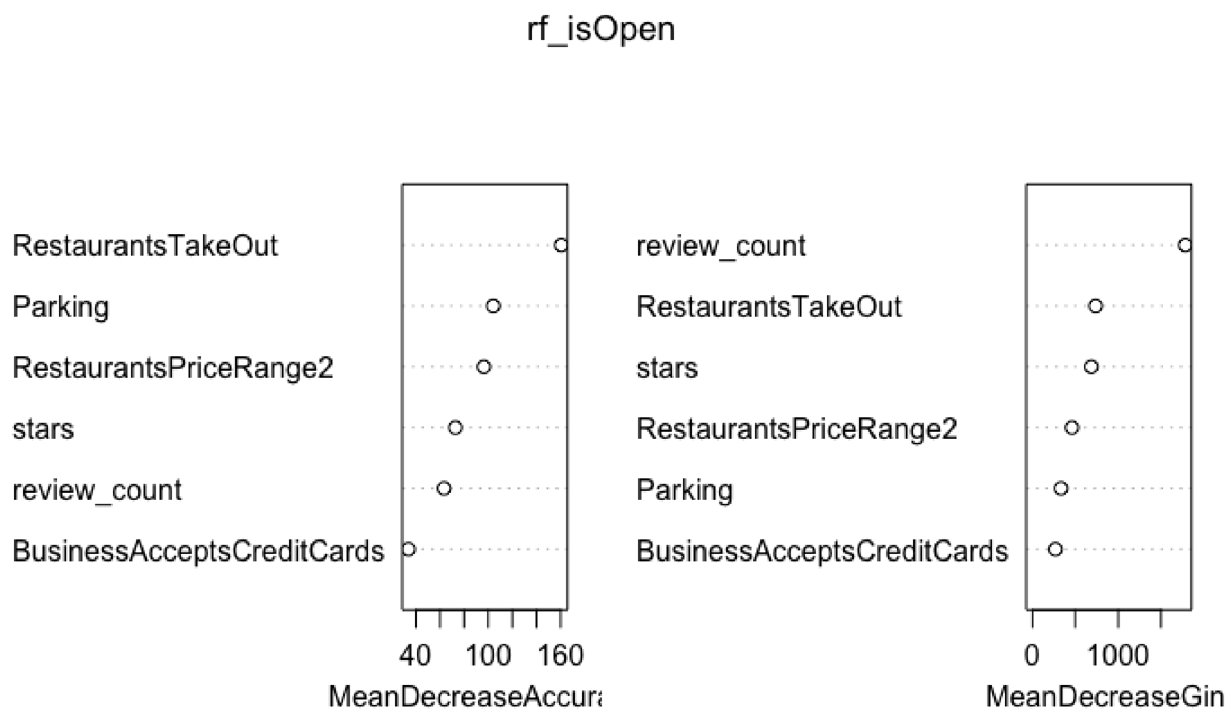
The rpart model in this case is the same as the tree model as well. The model has four terminal nodes and three splitting categories: "Parking", "is_open", and "RestaurantsPriceRange2". The splitting categories intuitively make sense as to why they would determine review counts. If there was no parking, 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.

# Random Forest: Restaurant is_open

```
rf_isOpen = randomForest(is_open~.,
                         data = cleanDf,
                         mtry = 3,
                         importance = TRUE)
print(importance(rf_isOpen))
print(varImpPlot(rf_isOpen))
```



```
                              Closed      Open MeanDecreaseAccuracy MeanDecreaseGini
review_count              -40.681762  61.41748             63.40047        1784.6963
stars                       9.251425  71.01549             72.84453         687.4291
BusinessAcceptsCreditCards  8.836592  20.78449             33.86812         267.5167
RestaurantsPriceRange2    -11.615308  83.29086             96.46985         461.6238
RestaurantsTakeOut         44.529877 139.67120            160.69699         737.3780
Parking                   -40.757345  94.36383            104.26909         333.0249
```

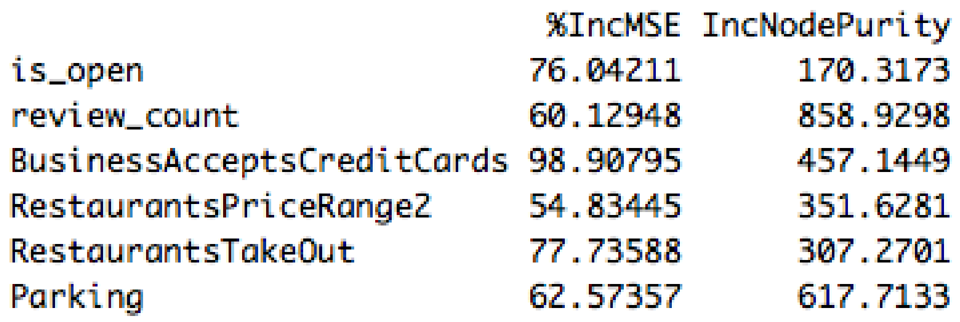


We used sqrt(p) variables or approximately 3 mtry's because this was a random forest of classification trees. We tested it on the entire dataset, cleanDf. The results indicate that across all of the trees considered in the random forest, information on RestaurantsTakeOut is by far the most important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

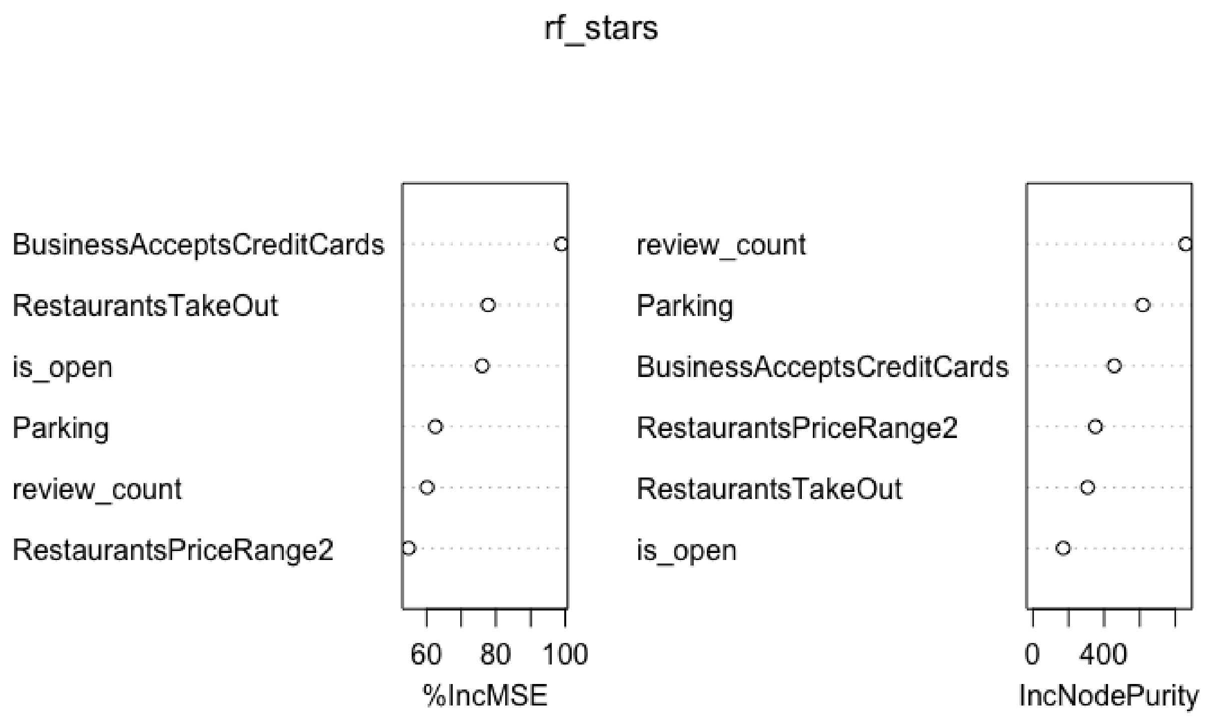
## Random Forest: Restaurant stars

We used p/3 variables or approximately 2 mtry's for the following random forests because this was a random forest of regression trees. We tested it on the entire dataset, cleanDf for both cases.

```
rf_stars = randomForest(stars~.,
                         data = cleanDf,
                          mtry = 2,
                         importance = TRUE)
print(importance(rf_stars))
print(varImpPlot(rf_stars))
```



```
                            %IncMSE IncNodePurity
is_open                    76.04211      170.3173
review_count               60.12948      858.9298
BusinessAcceptsCreditCards 98.90795      457.1449
RestaurantsPriceRange2     54.83445      351.6281
RestaurantsTakeOut         77.73588      307.2701
Parking                    62.57357      617.7133
```



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.

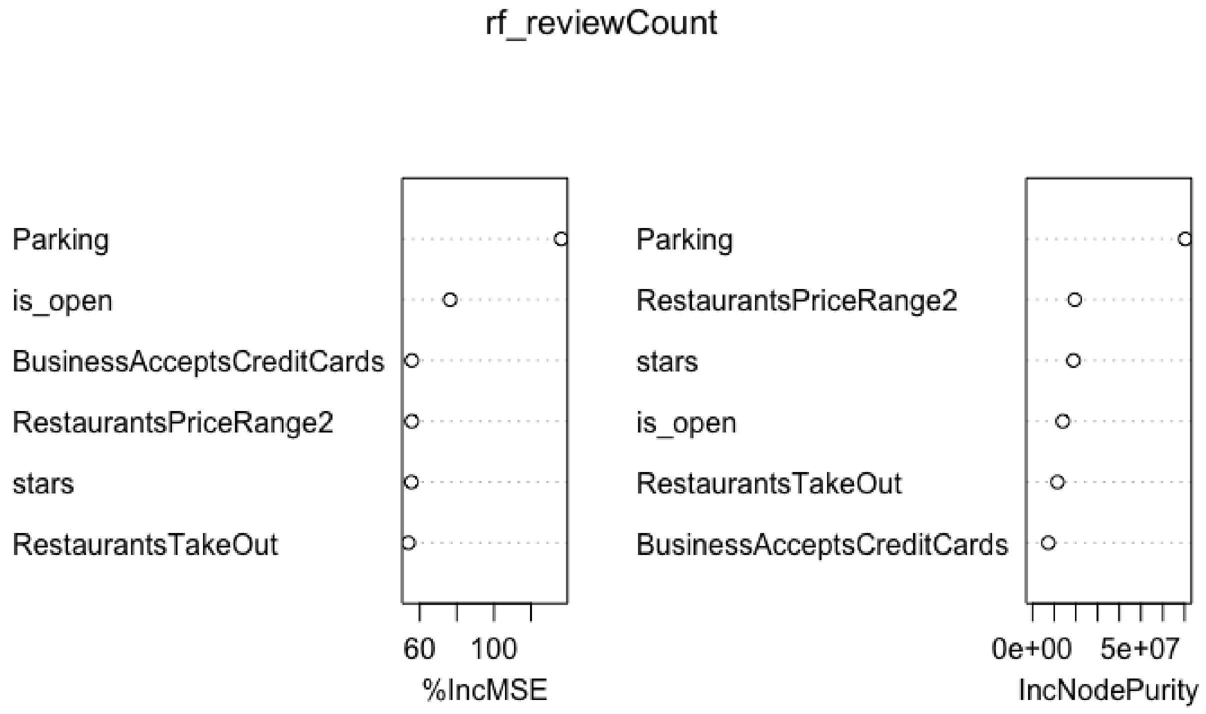
---

## Random Forest: Restaurant review_count

```
rf_reviewCount = randomForest(review_count~.,
                          data = cleanDf,
                          mtry = 2,
                          importance = TRUE)
print(importance(rf_reviewCount))
print(varImpPlot(rf_reviewCount))
```

---

```
                            %IncMSE IncNodePurity
is_open                     76.36389      13984326
stars                       55.46820      18873379
BusinessAcceptsCreditCards  55.68179       7303602
RestaurantsPriceRange2      55.67413      19492750
RestaurantsTakeOut          53.88925      11523896
Parking                    136.39255      70434751
```



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.