# Milestone Report 2

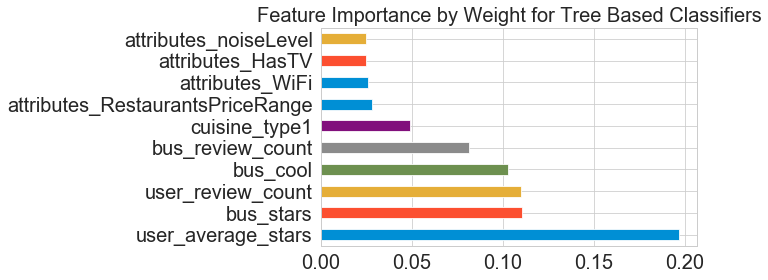
## Machine Learning:

This project was aimed at providing a recommendation system utilizing the Yelp Dataset. This would provide insight into how likely a user is to like a restaurant (rating greater than 3) and if they would give a rating greater than the business’ average. This also could provide some additional insight to be used in other functions of the business.

To gain this insight, I chose to utilize a classification-based set of prediction models including Logistic, KNN Classifier, ADAboost, SVM, Random Forest, and Decision Tree. Comparing the results of the various models in terms of accuracy would allow me to identify which of the prediction models was best to take a deeper exploration into.

**The first question to answer**: Would a user like a restaurant? (Give a rating greater than 3)

Once choosing the means to complete this prediction system, it was a matter of determining which features would provide the greatest impact. I completed the feature reduction for Tree Based Classifiers and the results are as follow:



This allowed me to identify the 10 most important features to utilize in my prediction systems.

The next step was to create and test the various models and being sure to fine tune the parameters for each model. Below is a summary of the results:



Consistently, the ADAboost Classifier was the best performing based upon its accuracy score determining how often it predicted correctly whether a user would give a rating greater than 3.

A more in depth exploration of the ADAboost model can be seen below:

**Confusion Matrix:**

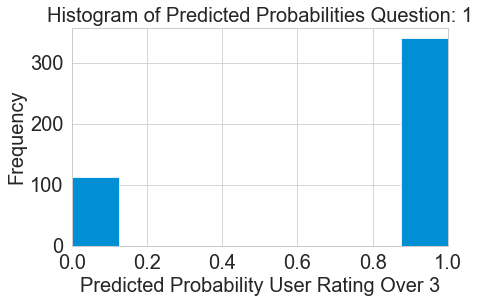
[[ 77 75]

[ 36 265]]

|  |  |
| --- | --- |
| 77 True Negatives | 75 False Positives |
| 36 False Negatives | 265 True Positives |

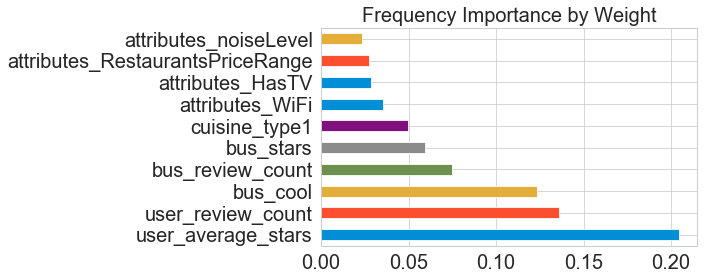
|  |  |  |
| --- | --- | --- |
| Attribute | Result | Description |
| Accuracy | .755 | How often classifier is correct |
| Error (misclassification rate) | .245 | How often classifier is incorrect |
| Sensitivity | .8804 | When actual value is positive, how often is classifier correct |
| Specificity | .5066 | When actual value is negative, how often is classifier correct |

Our classifier is highly sensitive and not very specific. Meaning that it does a great job of predicting when the actual value is positive, but struggles with when the actual value is negative. This greater sensitivity may also be due to the fact that majority of users give higher ratings, if you recall the plot showing user average star distribution from above.



**Question 2:** Would a user give a rating greater than the businesses average?

Let’s explore our second question in greater depth. It began with identifying the features that would provide the greatest impact on our prediction model. This can be seen below:

After understanding which features were most important, I built the same set of models with their individually tuned parameters. The results are below:



We see here that the ADAboost Classifier was the most accurate prediction model again. Let’s take a deeper exploration into this model to understand even more.

**Confusion Matrix:**

[[ 136 93]

[ 68 156]]

|  |  |
| --- | --- |
| 136 True Negatives | 93False Positives |
| 68 False Negatives | 156True Positives |

|  |  |  |
| --- | --- | --- |
| Attribute | Result | Description |
| Accuracy | .645 | How often classifier is correct |
| Error (misclassification rate) | .355 | How often classifier is incorrect |
| Sensitivity | .696 | When actual value is positive, how often is classifier correct |
| Specificity | .594 | When actual value is negative, how often is classifier correct |

Our classifier is just fairly sensitive and specific when predicting if a user would give a rating greater than the business’ average. The second question we targeted to predict wasn’t as accurate as our first question, and this may still be due to the user high rating bias we saw earlier.