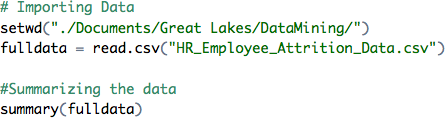
# Introduction

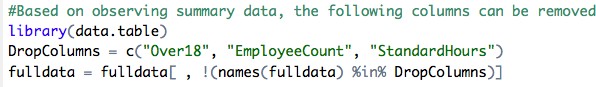
We’ve been given data on attrition. Based on our analysis we try to predict attrition.

# Importing and preparing

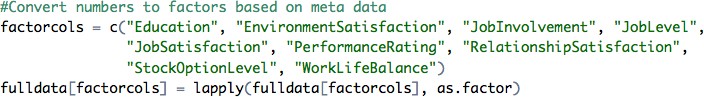
* We use R as a tool to perform analysis and predict our results.
* We import & view the summary of the data



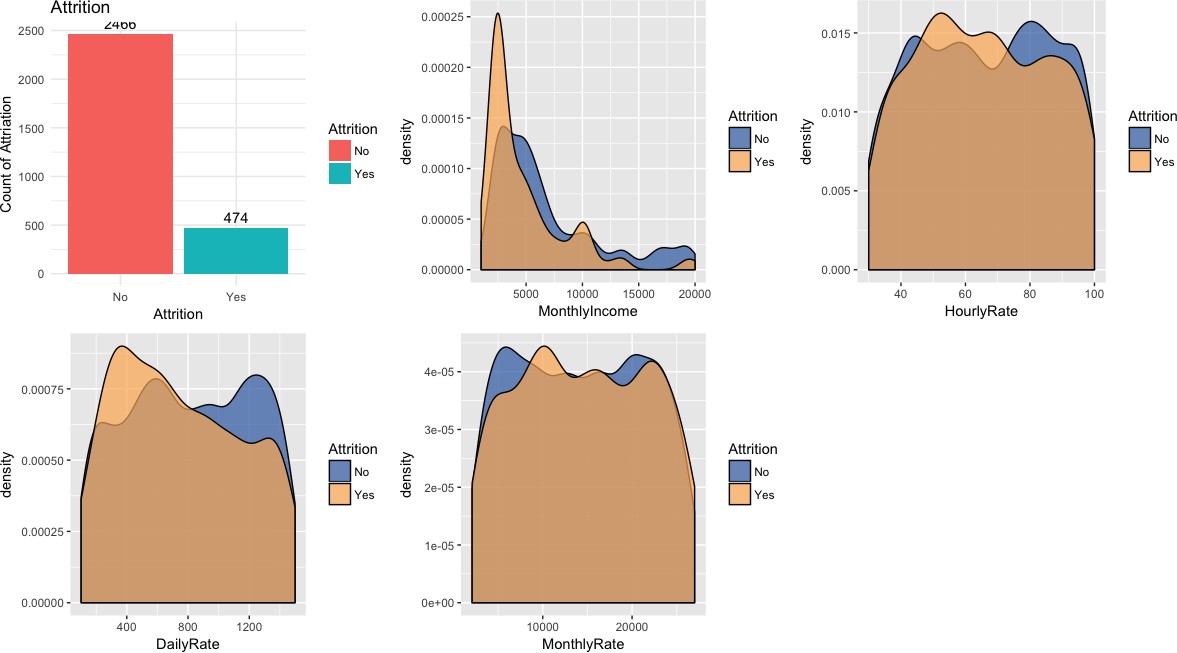
* + Upon viewing the summary, we understand that the following three columns have the same value throughout and is of no use to our analysis.
    - Over18
    - Employee Count
    - Standard Hours
  + Hence, we remove these columns.



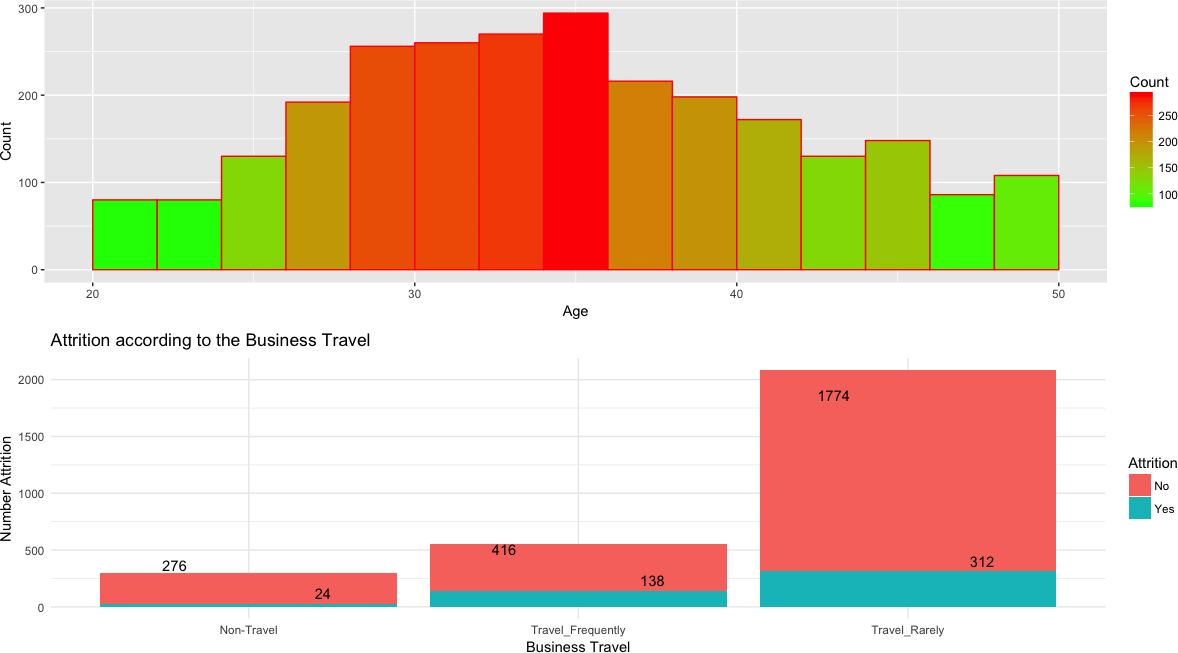
* + Based on meta data, we now have to convert the necessary columns from numbers to factors.



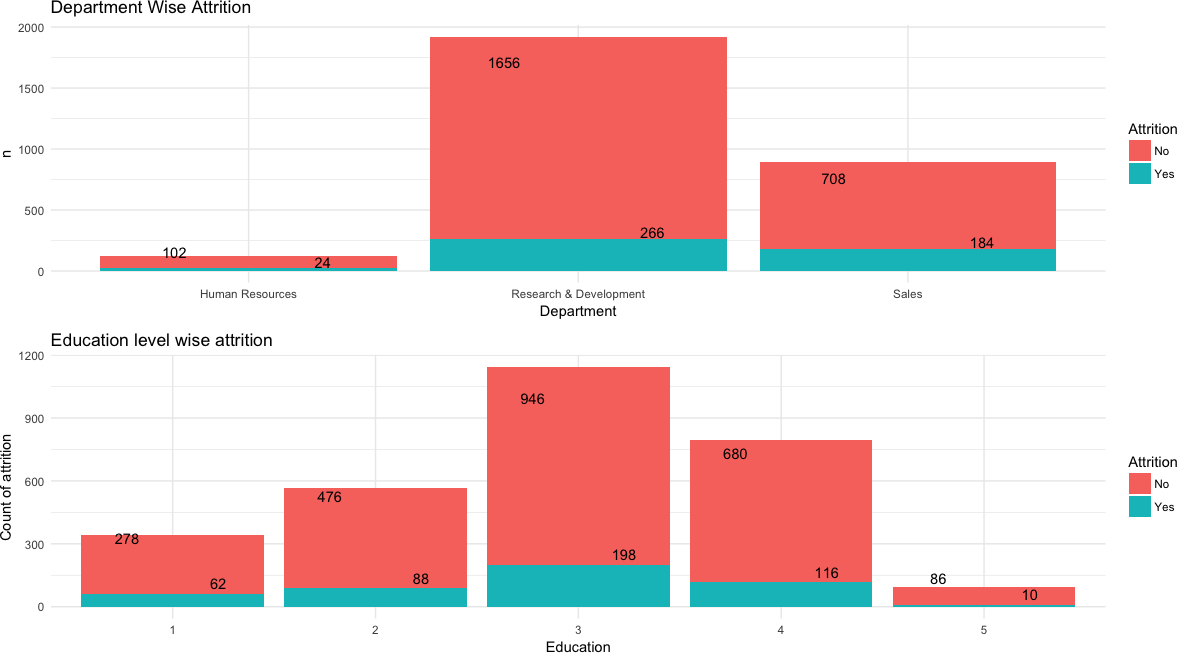
# Visualizing the data



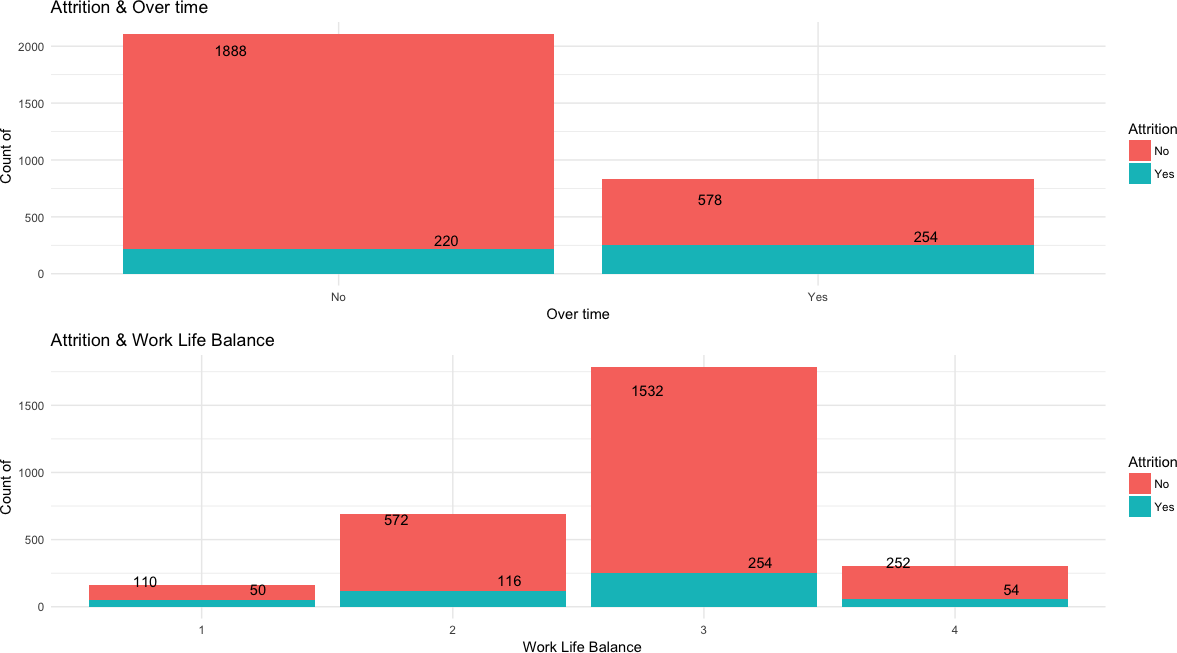
* + Based on our observation only 474 / 2466 = 19.22% is “Yes” in attrition. If we build a model directly on using this data, then the model would be more biased towards “No”. This bias can be treated with techniques like resampling or using more data etc.,
  + There is a high amount of attrition amongst low income employees.



* + High attrition can be observed with people between 27 – 36 years of age.
  + We can note that people who travel frequently for business are more likely to leave the job than people who do not travel.



* + We can see that sales department has a higher attrition rate when compared to other departments.



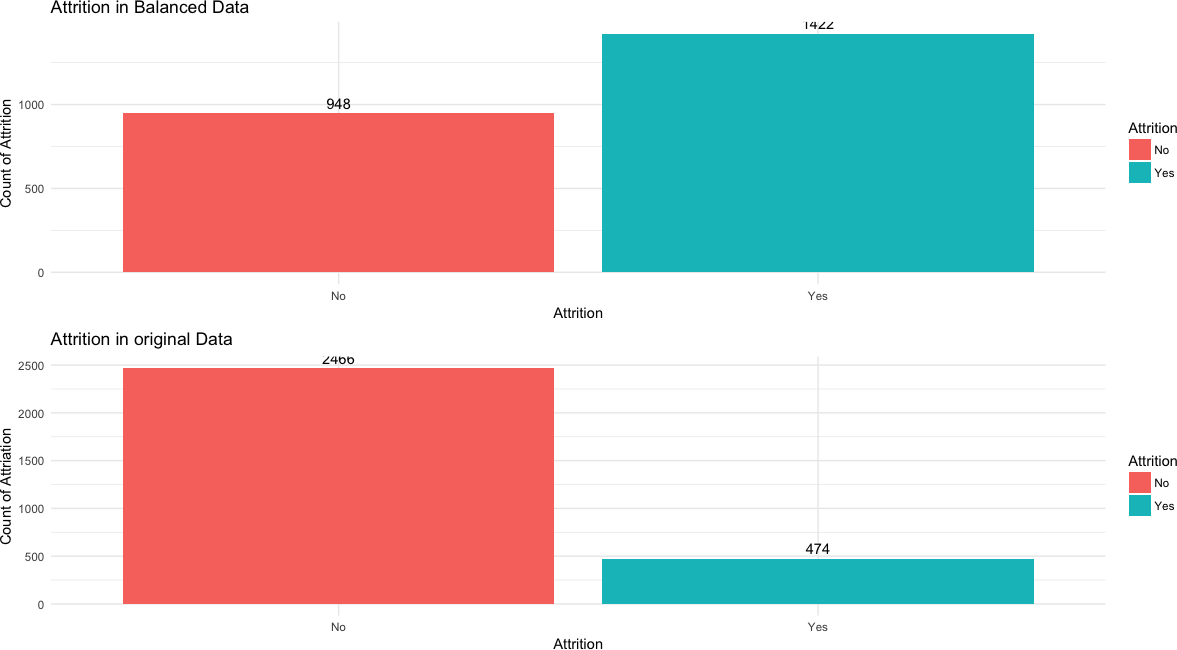
* + We can see that people who tend to do overtime are more likely leave the organization
  + Likewise, people who rated to having good work life balance are less likely to leave the organization.

# Models & Predictions

### Preparing the data for prediction

***SMOTE***

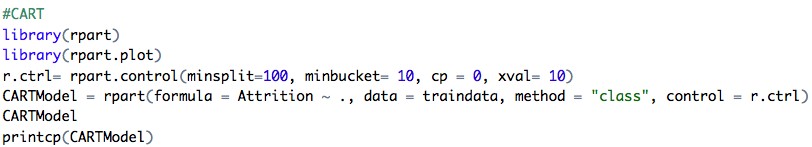
We earlier noted that the data is not balanced. Hence, we perform SMOTE to balance the data out.

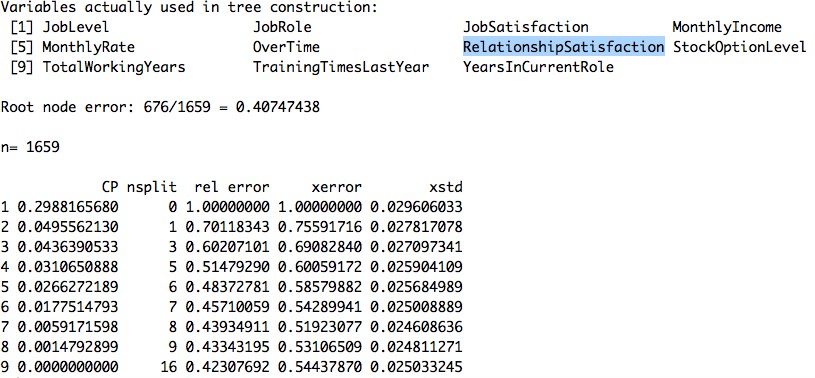


We need to split the data into train & test. As recommended in the assignment instruction, the data is split with 70:30 ratio. 70% for training and 30% for testing.

### CART

We build the CART model after setting up the basic control settings.



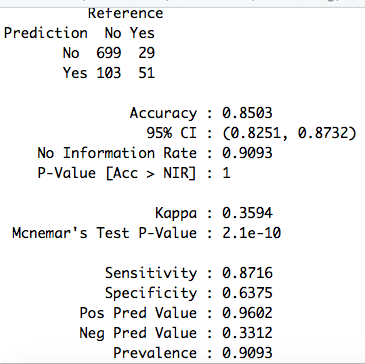


Now we can predict.

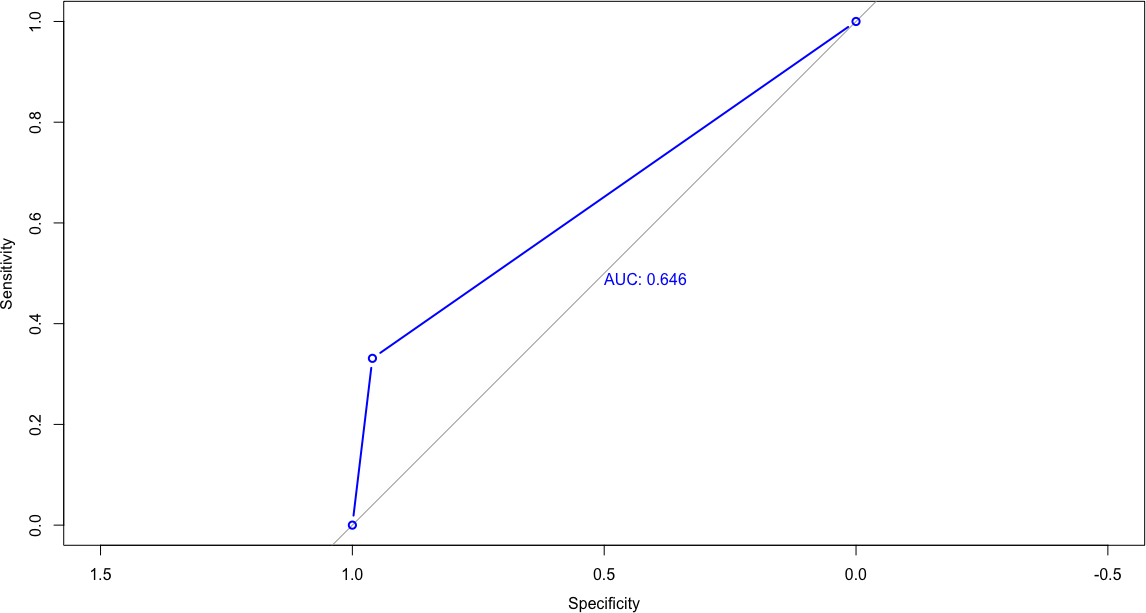


***Without Balancing:***

The prediction results are as follows. The model accuracy is 85.03% which is pretty decent.

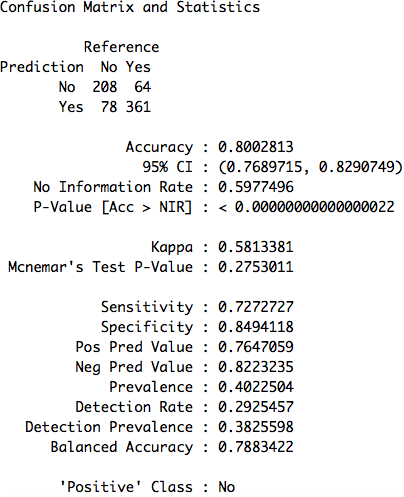


We now check the AUC Score.

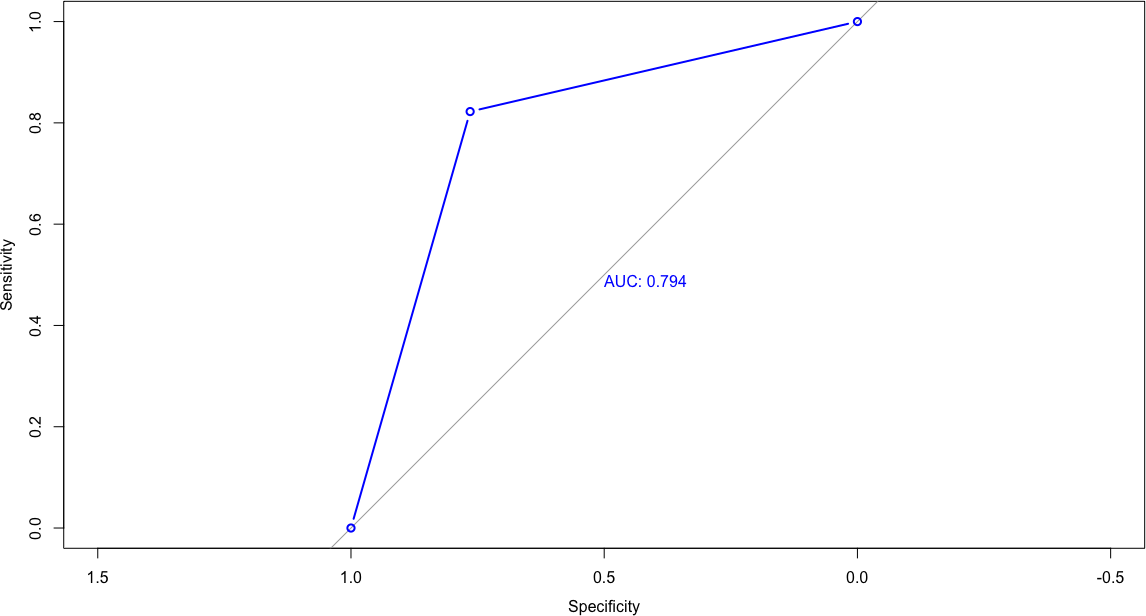


AUC: 0.646. The score is moderate.

## With Balancing:



ROC:



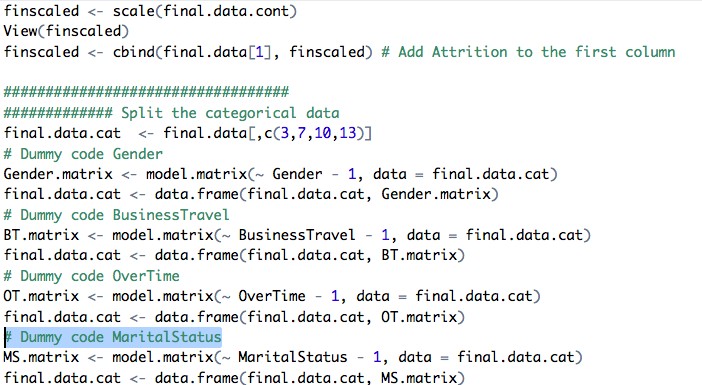
**AUC: 0.794.**

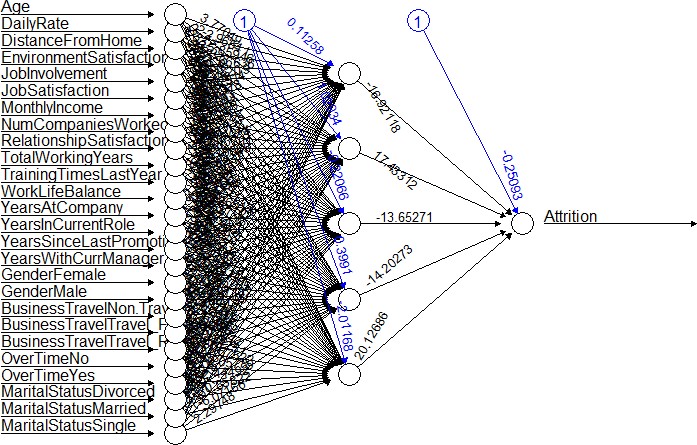
**Note that after balancing the AUC score improved significantly.**

**Neural Network**

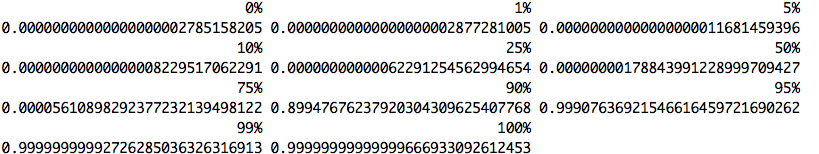
We use the same balanced data set to predict using neural nets as well.

First, we scale the numeric variables in the data so that the gradation may happen properly. Next for using qualitative data in neural nets we use dummy variable coding method.



Now, we build the Model.

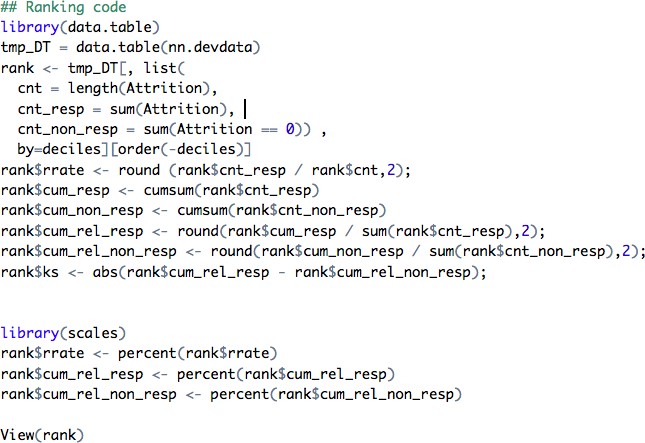
We observe that the predicted probabilities are gradating properly.

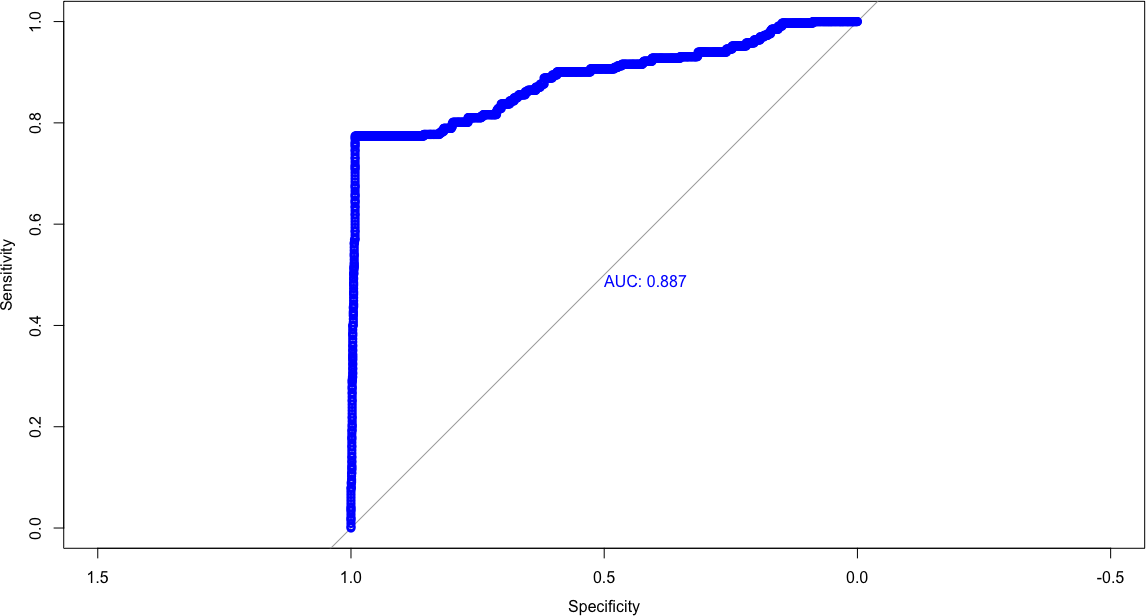


The Gini index is calculated:

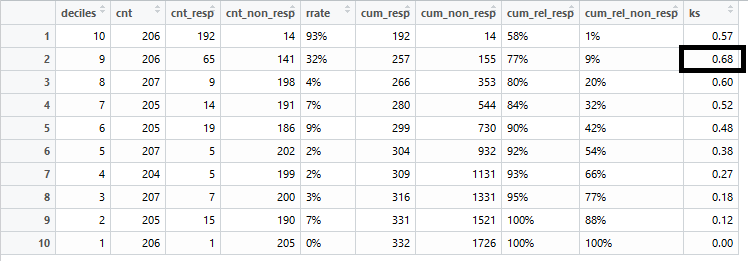


The Gini index has a healthy score.

Measuring the ROC:

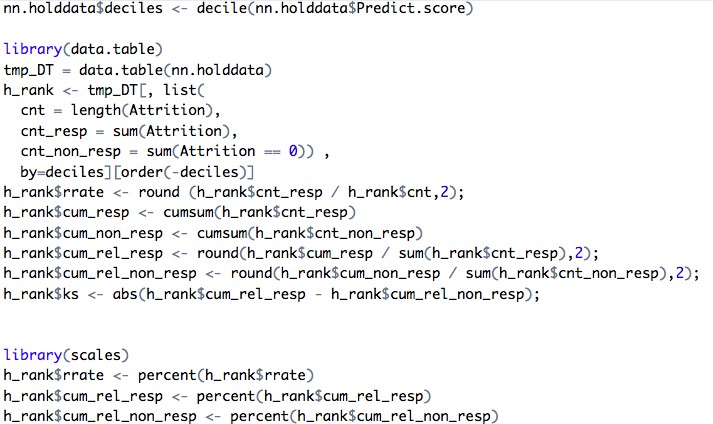


The ROC measure indicates that the model is good. We now decile the data & arrive the KS factor:

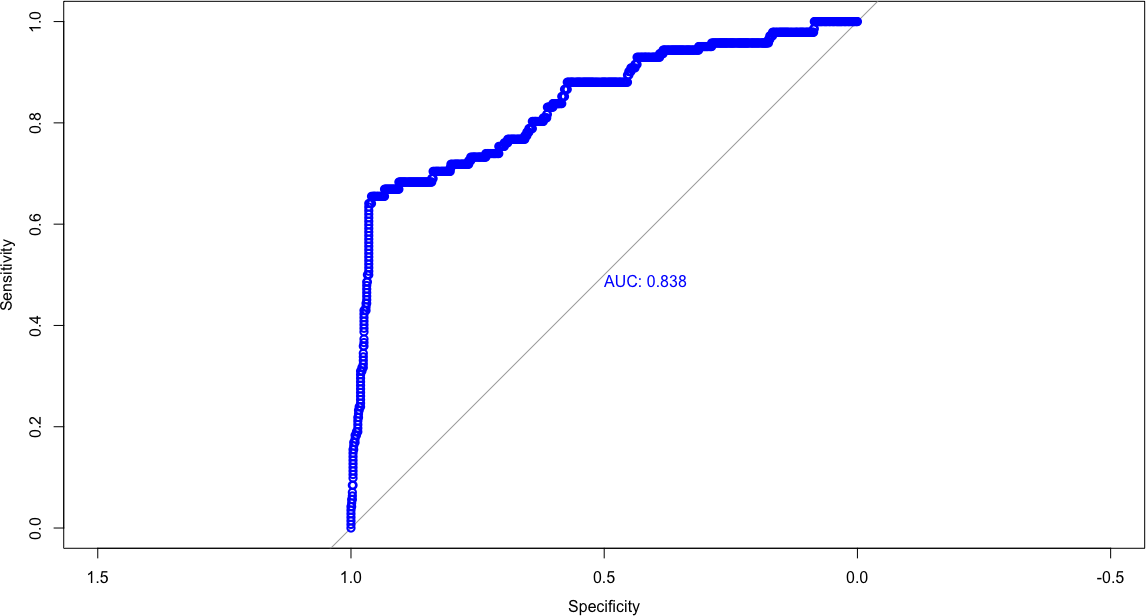


KS Factor for Train: 0.68

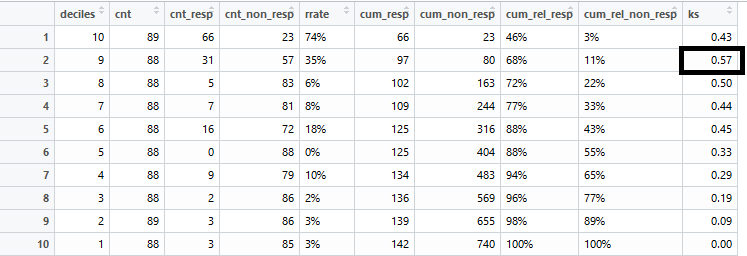
Now run the neural network model on test data.



ROC for Test Data:



0.838, which indicates that he model is quite good at predicting.



**Comparing CART & NN**

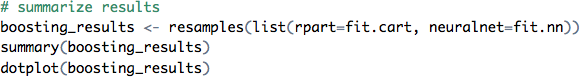
Based on ROC scores & Accuracy, **Neural networks seem to be the better model suited for predicting this data.**

### Ensemble (Boosting)

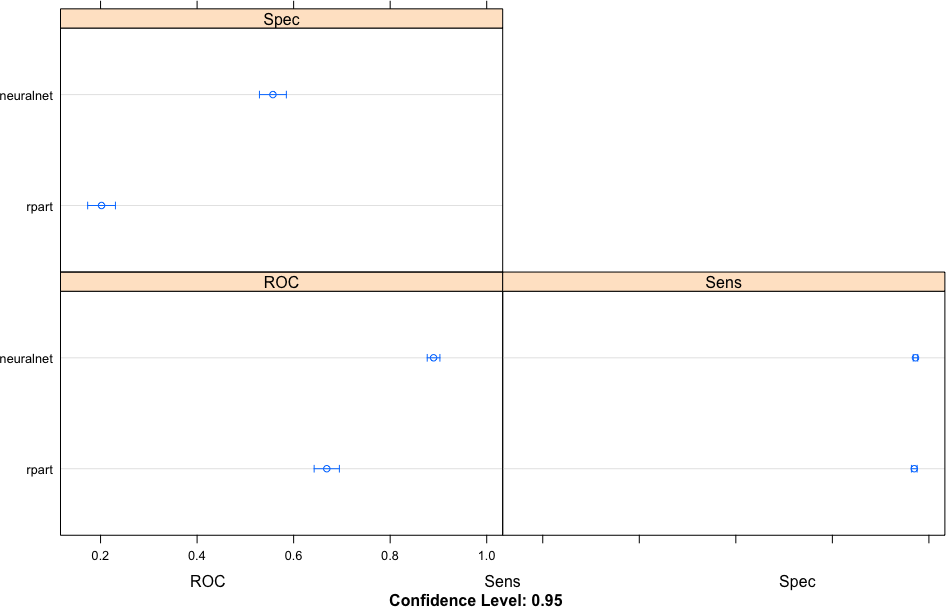
For the ensemble, we go with boosting technique.

Let us prepare the models. We use caret Ensemble library.

As part of the boosting methods, “rpart” and “Model Averaged Neural Network” is used. Measure is set to ROC. A repeated cv method is used.



The results of the ensemble model are as follows.



The mean ROC Score for neural is 0.88 and for CART is 0.66.

# Conclusion

Based on ROC score we can conclude that the ensemble model would produces the best result for this data.

### Sources:

<https://machinelearningmastery.com/machine-learning-ensembles-with-r/> <https://datascienceplus.com/fitting-neural-network-in-r/>