# Data Mining Technical Report

## Using Different Methods for Metabolite - Predictions

**Cortney Mood and Thomas Mims**

## Introduction

With the basic fundamentals of core machine learning and the many different data mining algorithms that we were taught in CSCI 334 (Data Mining) at the College of Charleston under the leadership of Dr. Paul Anderson, it was now time to take these skills to a different level and apply them to real world situations that people are actually trying to solve. In this report, our goal is to display the effort that we put in to develop an algorithm to predict whether or not a metabolite is present in a spectrum or not. The certain metabolite that we were required to work with is Tyrosine. Working with this one metabolite simplified our work but still didn't make it less difficult. The main concern was to try and figure out the best method that will give you the most accuracy of Tyrosine present. The details to this project can be found on Github at https://github.com/Anderson-Lab/Metabolite-Prediction.

## Methods

There were many different ways that we could tackle this problem. So, we decided to divvy up the tasks and see the possible results that we could derive. Cortney thought that the best methods to use were those of random forest, k-neighbor, and linear discriminant analysis.

Linear discriminant analysis is something that was very easy to understand when it was taught by Dr. Anderson. Just to give background information as to what that is, the principle insight of linear discriminant analysis (LDA) is that the covariance matrix can tell about the scatter that is within a dataset, which is the amount of spread that there is within the data. The way to find this scatter is to multiply the covariance by the pc, the probability of the class (which is the number of data points there are in the class divided by the total number). Adding the values of this for all of the classes gives us a measure of the within-class scatter ($S_W$), and we would want this value to be small. However, to be able to separate the data we would also want the distance between the classes to be large, which is known as between-class scatter ($S_B$). So, you would have to find a weight vector that would help maximize $S_B/S_W$ in order to separate the classes more efficiently and accurately. This was considered to be the best method due to the fact that we could separate Tyrosine into specific regions. And from our knowledge, we could take the two maximum points and calculate the distance in between them to be considered the between-class scatter. However,

it was kind of difficult when putting everything together to try and figure out the correct parameters.

K-Nearest Neighbor seemed to be the method that made the most sense and was sort of easier to comprehend. The purpose of the algorithm is to find a way in which the data points are separated into several separate classes to predict the classification of a new sample point. Starting with a particular data set, which would be a specific data set under Tyrosine for this report. We wanted to be able to predict the classification of new data points based off of the known classifications of the observations in the given data set. This would make sense because one possible way that you could figure out if there is Tyrosine present is by comparing it to another graph that does show that Tyrosine is present. In order to do that, we would still have to decide which observations from the data set are similar enough to our new observation for us to take the classification into account. One way to solve that is by considering all of the data points that are within a certain radius of the new sample point. Or, we could take a certain amount of what are considered to be the nearest points.

The final method that Cortney thought would be useful is random forest and is also kind of similar to the nearest neighbor prediction. It is also an ensemble approach. The main principle behind ensemble methods is by having lots of learners that each gets slightly different results on a particular dataset (where some learn other things well and others are considered to be weak) and putting them together, the results that are generated will be significantly better. The random forest starts with a standard machine learning technique called a "decision tree" which, in ensemble terms, corresponds to our weak learner. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets.

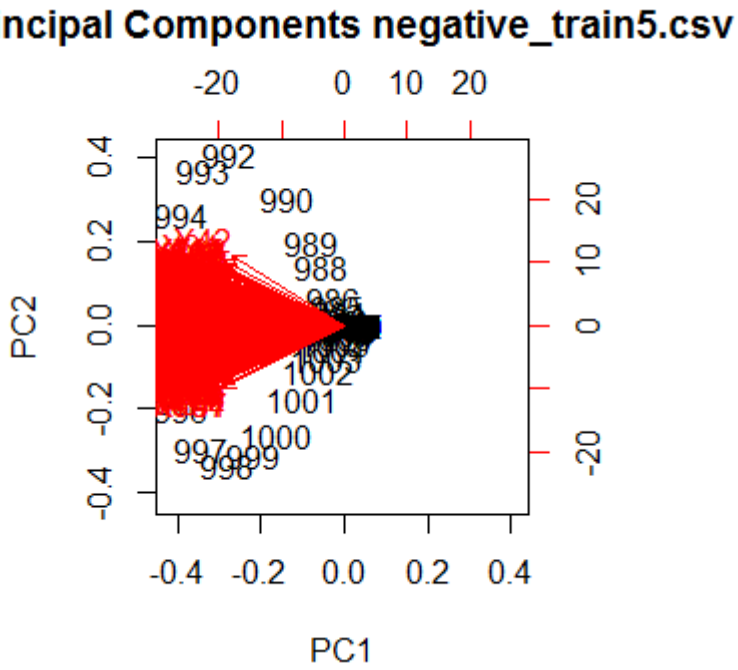
From here we decided to implement the Rattle package, which is an interactive GUI offered through the CRAN packages. Rattle implements many graphical packages as well, including RGtk2 and rggobi. Rattle was also a great learning tool for us, considering neither of us have much experienced with data mining beyond Dr. Anderson's course. Rattle did not require any comprehension of code, just comprehension of methods and how they interact with the data sets that we feed into it.

Rattle, although easy to interact with, was not as easy to install on our personal machine. We started by using Thomas' R Studio online that was provided to us through Learn 2 Mine, however due to the lack of operating system, Rattle was unable to properly load. More specifically, the RGtk2 package which Rattle called produced a "namespace error". We took steps to install desktop R Studio on Thomas' workstation, and installation of all packages proved successful through this version.

Rattle gave us a large variety of workspace. Our first idea was to create a graph of each data set individually (all positive and negative sets from easiest to hardest). This gave us a visual representation of the size, which we previously had not been able to effectively view in R Studio due to how large the sets are. From there we were able to also check the mean, median, highest

value, and lowest value of each set. We wanted to be sure that we did not produce any outputs that would completely contradict the range of numbers given in each set.

Our next idea was to run the sets through the option of principal component analysis, or PCA. Rattle provides the user with a graphing PCA option, which was very useful for understanding the complexity of our datasets. The graph for the negative_train5.csv is displayed below, for example. From this graph, we can see that the variance lies across the horizontal like of 0.0. This is not shocking, considering that there is almost no other line other than a horizontal that can produce this type of variance. It was also not a surprise that all of our other sets, including the positive training sets, came to produce a similar output.




**Results**

Our final results were inconclusive. We were able to come through with very skewed results using random forest (for many of the data sets we would get results ranging from 0.23% accuracy to close to 80% accuracy without changing our methods). We had hoped to eliminate this range using the Rattle package, however we found that although Rattle is visually extremely useful, it does not run comparison on more than one data set. Rattle is more suited to within set comparison, such as male heart attacks vs. female heart attacks or weather sets.

## Discussion

For future research and projects, we believe that it is best if we take better time in trying to come up with better hypotheses. In addition to that, it would also be beneficial to practice the R language some more on our own time while we are not in school or even possibly consider taking the beginning course of data mining that is offered at the College of Charleston. It is difficult for students that have not been exposed to data mining to come up with different hypotheses for things that are happening in the real world. We feel that it would be beneficial to have a pre-requisite class devoted to just programming in R, this way we would be more prepared for the data mining course.

## Conclusion

Overall, this was a great learning experience being that we are both beginners of data mining. Even though there was not always a high number of accuracy within the different methods shown, the main purpose was that we were able to figure out a way to make our methods more efficient. In addition to that, this does not mean that we were not successful. It just means that we would have to try and continue to establish better methods in order to get a higher accuracy. Many of them did not work due to the fact that there were inaccurate calculations. And that could be because we did not look at the regions correctly or were not able to come up with logical, efficient hypothesis in order to calculate the accuracies correctly. However, we would recommend using