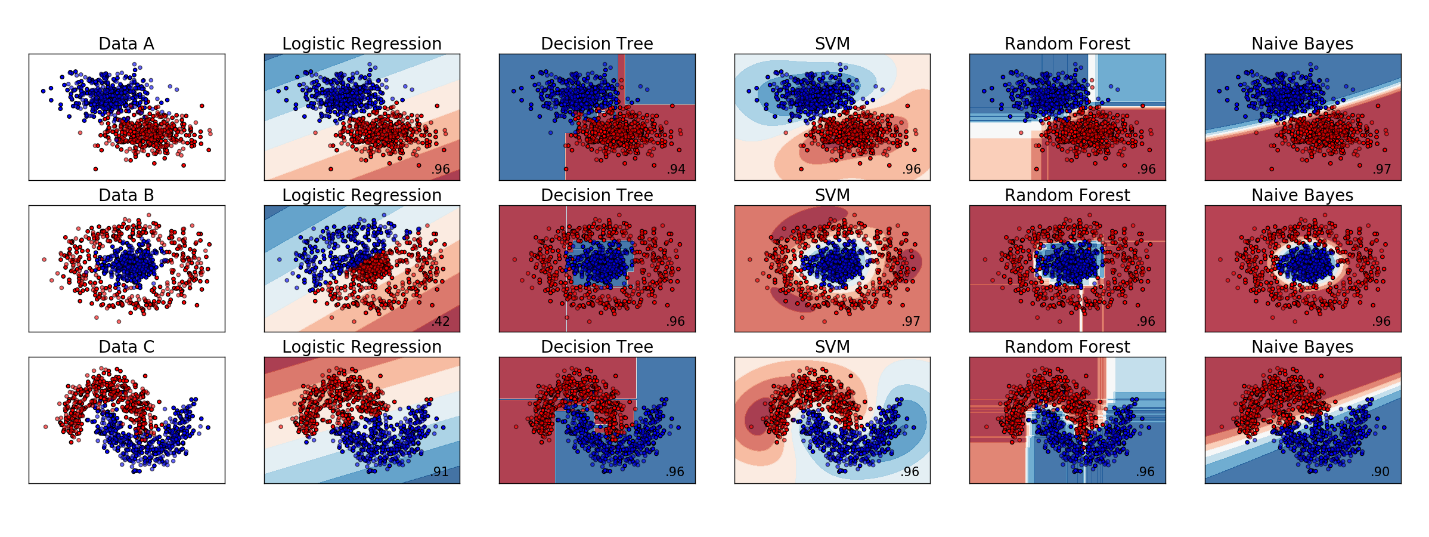
Luna McBride

CSCI3202

**Assignment 3 Report**

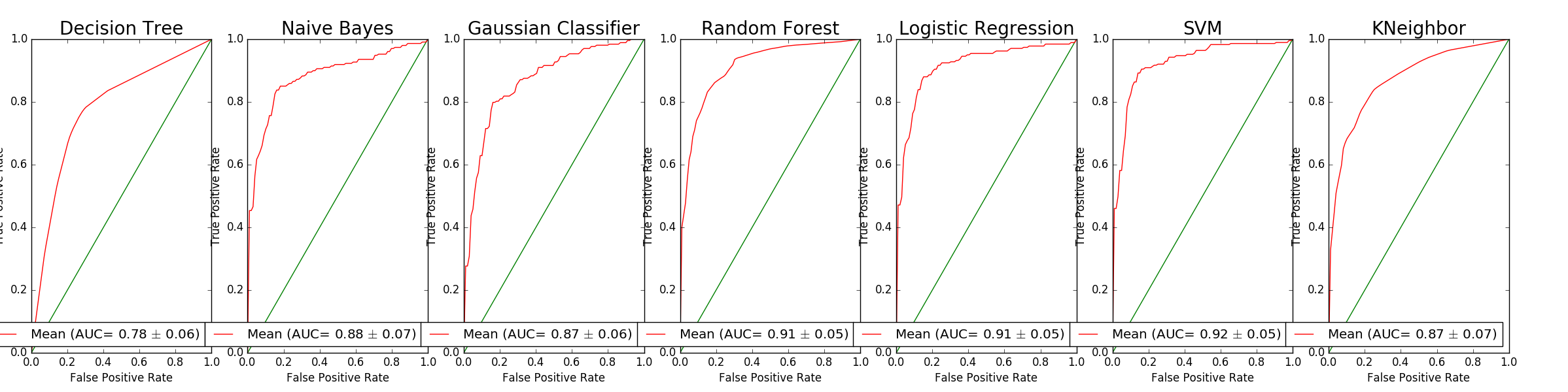
**Part I:**



1. Accuracy:
   1. Logistic Regression: This approach worked quite well when you could draw a strong line between groups. Mixing and outliers were hardly considered here, leading to the mixed points misclassified. Due to this strong split tendency, the one with one encasing the other got a horrible score of 0.44, meaning it misclassified more than half.
   2. Naïve Bayes: This one was quite nice for areas with clear-cut sections of the screen devoted to the group. This included one group encasing another, however, its range was pretty large. In addition, by staying linear, it had a bit of trouble when points looped back down at the side, giving it issue on data set number 3.
   3. SMV: This approach gave clean-cut details with areas of strongest relation to the group having darker colors. All datasets were well accounted for with a little issue for one datapoint encased in the overall other group (which is always a hard thing to account for). Despite all of this, because it encapsulates the data so well, adding a new point may upset many parts of this set and cause a shift to be needed to make it better again. If we are not changing the data, this is a very good representation, but if we are in a constantly-changing dataset, there could be better.
   4. Decision Tree: Like the SMV, this approach had all datasets well accounted for. Of course, there were some merged areas that were hard to account for, but that is par for the course at this point. The boundaries, however, were very specific, some even being simply a very thin offshoot from the main area. This leaves almost no room for additional data points without restructuring the whole group distinction entirely. It had no grey area like some of the others had, which would likely fall apart if a red hit into one of those thin blue line areas, for example.
   5. Random Forest: This one was another one that had all datasets well accounted for. Unlike the previous two with very strict boundaries, however, it leaves some room for indecision. There are white areas with no data and lighter areas where, based on the data, the area is more likely to contain members of that group than others. This not only allows for some ambiguity; it allows further points to come in with little need to change up the entire graph. The forest seems to be the best for the trade off of flexibility and accuracy, but model choice really depends on the need of the specific dataset.
2. Conclusions:
   1. Logistic Regression: This one is very situational. This one worked best when the data clearly had a line separating the data and a relatively large decision boundary. The one with one group encasing the other met none of those properties, and as such, was very poorly represented, for example.
   2. Naïve Bayes: This one did not do as well as some of the others. It had a bit of issue with crossover like with the spiral of graph number 3, however, its decision boundary made it a fair choice for datasets that will be adding more data. Data can be added without much recalculation with this one (unless there is a huge curveball, but that can be said about all of them), which makes it a strong model for change-based data sets.
   3. SMV: This method fit the models really well, with an overall closeness that it had the highest match values of all of the models. Despite this, the decision boundaries were super close to the data. This gives almost no variability if more random points are added. This makes it the best for non-changing datasets, but most sets are not like that.
   4. Decision Trees: This fits the models pretty well as well, but falls into similar problems in specificity. This one has decision boundaries so specific that some of them are random thin lines shooting out from the main blob, as well as assuming further points are most definitely part of the group its point lands in (as shown by a bold color instead of having ranges of lighter/no color). This is another one that works best in static datasets, but it does not necessarily have to do as much crazy changing as the SMV when data changes.
   5. Random Forest: This is another one that fits the models well. The forest, however, works in ranges with white and lighter colored areas for a rough idea where further points will go, but without saying for sure. This means it has better decision boundaries for change, as more points that come out of left field have a space without warping all the boundaries. Of course, it is still strong for static datasets, but it has better tools for handling change in data.

**Part II:**

**Note: 1=2a, 2=2b, etc. This was just easier in Word**

1. 
   1. Above
   2. I would say the SVM was the best overall. The standard deviation was one of the lowest along with the highest mean, making it a strong choice. It is important to note, however, that both Logistic Regression and Random Forest had the same standard deviation, but the mean was under SVM by 0.01. Those are also great choices, but the SMV mean was a little higher.
   3. -KNeighbor: This method looks around points for their k nearest neighbors, judging group possibility based off neighboring points. Basically, this is assuming you belong to what is around you, which is not the smartest choice in this situation, but it was better than the decision tree, so…

-Gaussian Classifier: This uses Gaussian (Normal) distribution-like probabilities to classify different objects. This was probably not the best here, as our dataset was large and not really able to classify easily as “Normal”, but I just thought it would be interesting to see how it competed.

1. Hyperparameters:
   1. The Grid Search method was used, which essentially brute forced all the combinations of given numbers from a grid to find the best combination to create a better result. (Code commented out after testing because it takes quite a while)
   2. -max\_depth: forests are full of a bunch of trees. The max\_depth dictates how big a tree is allowed to be, changing how much, and thus, which points are in a specific tree, changing the forest.

-n\_estimators: this is the number of trees in the forest. The more n\_estimators grows, the bigger the forest.

-C: this is our allowed margin for error. It essentially puts some penalty into the equation for misbehavior, killing the potential for issue the higher it gets.

-gamma: this is a term used as a multiplier. Depending on what we let the svc use (be it sigmoid, polynomial, etc), this multiplier alters that equation, which can then change the classification.

c. Despite the amount of toying with this function, the SVC mean and standard deviation did not change at all. The random forest did change, so it can be said that the problem was not with the method. Changing the kernel changed the results, of course, but the change in gamma and c did not change anything.

**I have run out of time. This is the end of what I could get done.**