Evaluating Classifier Performance on the Female Dataset

control = not sick = not affected
case = sick = affected

When looking at the performance of the classifiers on the female data set, we were encouraged by the results. However, we worried that part of these results could be a result of the process used to fill data points that were missing certain measurements. I filled the data points in to try and get a bigger data set to work with, as without the bigger set we couldn't start looking at classification tasks.

The process of filling data points relied on sampling from normal distributions, each of which's mean and variance were set by calculating the sample mean and sample variance of each measurement for all data points in the control and then similarly for the case.

So for each missing measurement in a given data point, depending if the patient was in the control or was an affected patient (case), we select either the control's normal distribution or the case's normal distribution, sample from it, and fill in the missing measurement.

We want to ensure that this data filling process isn't skewing our results… so we will try a different filling process where we don't generate two different normal distributions for each measurement, but rather ignore the affected status and only create one general distribution and sample from it to fill in the missing measurements.

We will compare the results of both filling processes and try to understand whether the original filling process is skewing our results or not.

How does knowing the affection status affect the process of filling missing data points?

Random Forest: (Filling process was run 100 times in both cases)

Knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.719
- Stdev of Accuracy: 0.058
- Median Accuracy: 0.707
- Counts for how many times each feature was filled in:

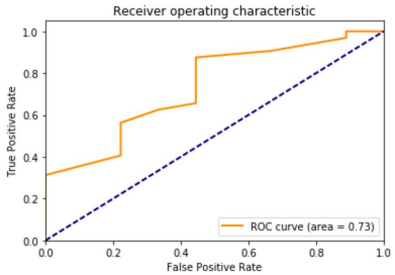
  time: {control : #ofTimesFilledIn, case : #ofTimesFilledIn}

  BL: {0: 0, 1: 0} V4: {0: 5, 1: 8} V6: {0: 5, 1: 6}

  V8: {0: 6, 1: 3} V10: {0: 13, 1: 29} V12: {0: 18, 1: 61}
- Average Feature Importance:

  BL - 0.1 V4 - 0.208 V6 - 0.154 V8 - 0.1 V10 - 0.168 V12 - 0.269

ROC Graph for Median trial



Not knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.541
- Stdev of Accuracy: 0.069
- Median Accuracy: 0.537
- Counts for how many times each feature was filled in:

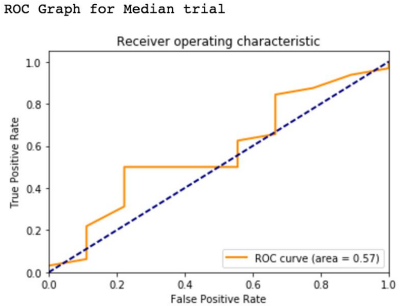
  time: #ofTimesFilledIn

  BL: 0 V4: 13 V6: 11

  V8: 9 V10: 42 V12: 79
- Average Feature Importance:

  BL - 0.124 V4 - 0.221 V6 - 0.201 V8 - 0.135 V10 - 0.149 V12 - 0.17

ROC Graph for Median trial



How does knowing the affection status affect the process of filling missing data points?

Adaboost: (Filling process was run 100 times in both cases)

Knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.747
- Stdev of Accuracy: 0.062
- Median Accuracy: 0.756
- Counts for how many times each feature was filled in:

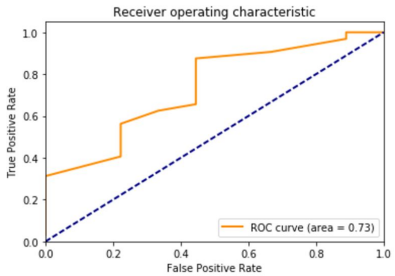
  time: {control : #ofTimesFilledIn, case : #ofTimesFilledIn}

  BL: {0: 0, 1: 0} V4: {0: 5, 1: 8} V6: {0: 5, 1: 6}

  V8: {0: 6, 1: 3} V10: {0: 13, 1: 29} V12: {0: 18, 1: 61}
- Average Feature Importance:

  BL - 0.086 V4 - 0.149 V6 - 0.131 V8 - 0.138 V10 - 0.156 V12 - 0.34

ROC Graph for Median trial



Not knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.544
- Stdev of Accuracy: 0.072
- Median Accuracy: 0.537
- Counts for how many times each feature was filled in:

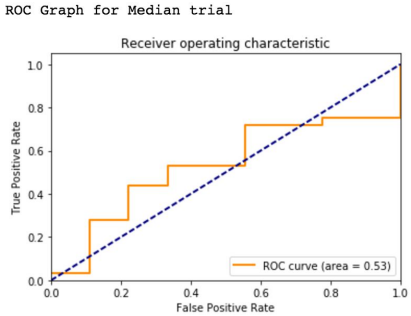
  time: #ofTimesFilledIn

  BL: 0 V4: 13 V6: 11

  V8: 9 V10: 42 V12: 79
- Average Feature Importance:

  BL - 0.092 V4 - 0.191 V6 - 0.165 V8 - 0.173 V10 - 0.18 V12 - 0.199

ROC Graph for Median trial



To ensure the average accuracy of the trials where the affected status was used to fill in missing data points is indeed different than the average accuracy of the trials where the affected status was not used, I ran a Welch t-test. Below are the results:

Random Forest:

statistic=19.709822786717421, pvalue=5.4129533454817911e-48

Adaboost:

statistic=21.257149465849306, pvalue=1.7347694746337791e-52

So yes there is a statistically significant drop in the accuracy of the classifier when not taking into account the affected status of an individual while filling in missing values for data points.

This suggests one of two things:

1. There is actually a difference between case and control and the filling method is just amplifying that difference
2. There is no difference between case and control and the filling method is creating the difference

We now randomize the labeling of patients as case or control and repeat the process. If we see a similar drop when using the affected label or not in the filling process, then there is a strong suggestion that the filling process is creating a difference where one doesn’t exist, as the labels are random.

Otherwise the filling method is just taking advantage of a natural split in the data, which more data would be able to confirm.

After Randomization

Random Forest: (Filling process was run 100 times in both cases)

Knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.661
- Stdev of Accuracy: 0.046
- Median Accuracy: 0.659
- Counts for how many times each feature was filled in:

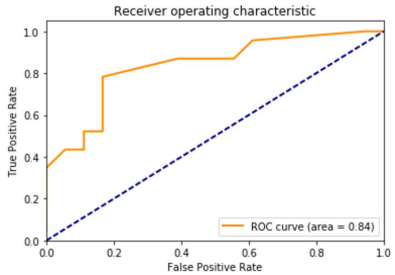
  time: {control : #ofTimesFilledIn, case : #ofTimesFilledIn}

  BL: {0: 0, 1: 0} V4: {0: 10, 1: 3} V6: {0: 6, 1: 5}

  V8: {0: 6, 1: 3} V10: {0: 20, 1: 22} V12: {0: 39, 1: 40}
- Average Feature Importance:

  BL - 0.087 V4 - 0.118 V6 - 0.155 V8 - 0.123 V10 - 0.199 V12 - 0.318

ROC Graph for Median trial



Not knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.415
- Stdev of Accuracy: 0.051
- Median Accuracy: 0.415
- Counts for how many times each feature was filled in:

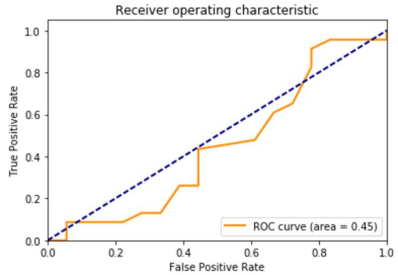
  time: #ofTimesFilledIn

  BL: 0 V4: 13 V6: 11

  V8: 9 V10: 42 V12: 79
- Average Feature Importance:

  BL - 0.126 V4 - 0.166 V6 - 0.185 V8 - 0.155 V10 - 0.185 V12 - 0.184

ROC Graph for Median trial



After Randomization (cont)

Adaboost: (Filling process was run 100 times in both cases)

Knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.608
- Stdev of Accuracy: 0.053
- Median Accuracy: 0.61
- Counts for how many times each feature was filled in:

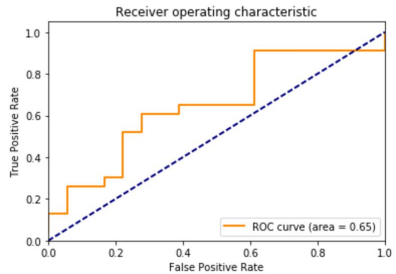
  time: {control : #ofTimesFilledIn, case : #ofTimesFilledIn}

  BL: {0: 0, 1: 0} V4: {0: 10, 1: 3} V6: {0: 6, 1: 5}

  V8: {0: 6, 1: 3} V10: {0: 20, 1: 22} V12: {0: 39, 1: 40}
- Average Feature Importance:

  BL - 0.165 V4 - 0.099 V6 - 0.191 V8 - 0.143 V10 - 0.103 V12 - 0.299

ROC Graph for Median trial



Not knowing Status when filling

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 82
- Number of Testing Data Points: 41
- Mean Accuracy: 0.439
- Stdev of Accuracy: 0.054
- Median Accuracy: 0.439
- Counts for how many times each feature was filled in:

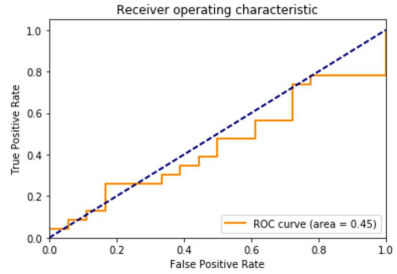
  time: #ofTimesFilledIn

  BL: 0 V4: 13 V6: 11

  V8: 9 V10: 42 V12: 79
- Average Feature Importance:

  BL - 0.152 V4 - 0.131 V6 - 0.126 V8 - 0.157 V10 - 0.176 V12 - 0.258

ROC Graph for Median trial



Just to ensure these differences are real, we perform a Welch t-test on the average accuracy.

Random Forest:

statistic=35.554600055280844, pvalue=2.1685590616235594e-87

Adaboost:

statistic=22.38127702418134, pvalue=4.0000520338910029e-56

The differences are significant, implying that the filling method is creating a difference where one doesn't exist.

This is further backed up by the fall in significance (in terms of feature importance) of V12 and V10 (the two data points that are getting filled in the most) when not taking into account the affected status. The continued importance of V12 even when not taking into account the affected status, is probably due to the fact that the majority of data points need their V12 value filled in, and the classifier is seeing a potentially larger variability in V12 values and trying to fit to that variability… not realizing that the variability comes from a noisy source and should be ignored.

This doesn't imply that measurements at V10 and V12 aren't important, we just need more fully filled out female data points to understand whether they are important or not.

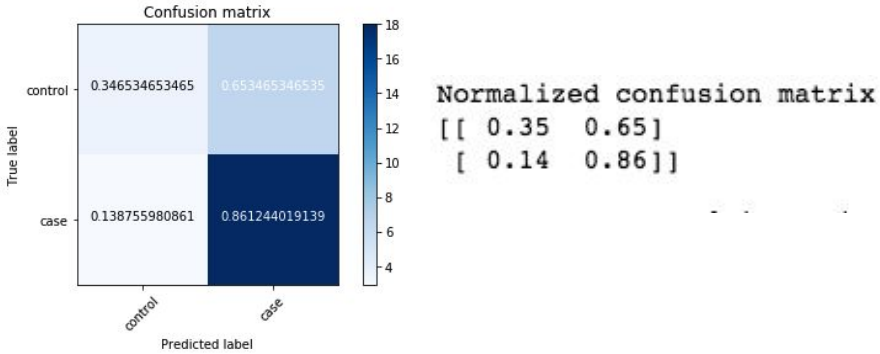
Ignore the slides from this point onwards… or you can read on, but be aware of the below points. The slides before this aren't affected by this realization by the way, I ran the processes again using more variability in the creation of the test-train split and everything still held.

I have realized that there is a big enough class imbalance in the dataset, and need to correct for that… the screenshot I sent you was a result of this. Essentially I didn't realize that in the test-train split created, the test split contained only 4 control examples (negative) and 27 case examples (positive), so the ROC graph, which measures True Positive Rate, and False Positive Rate, looked really good and hence incorrectly led me to believe that the classifier was performing excellently.

This class imbalance (roughly ⅓ control, ⅔ case) is causing the leaves of the random forest classifier to be filled with examples of case patients, but not control patients… and hence it tends to mark most things as 1. The adaboost classifier does do a better job of handling the imbalance though. All hope is not lost though as both classifiers definitely seem to be picking up on something.

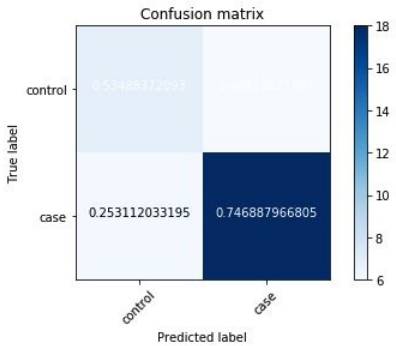
In a "hacky" attempt to ensure the classifiers are doing something I created 10 different train-test sets from the original data, and saw how the classifier did on each of these 10 train-test sets... Here are the averaged out confusion matrices for each classifier:

Random Forest:




```
Normalized confusion matrix
[[ 0.35  0.65]
 [ 0.14  0.86]]
```

Adaboost




```
Normalized confusion matrix
[[ 0.53  0.47]
 [ 0.25  0.75]]
```

Let’s see how the classifier does without using V10 and V12 values.

When we include the first four previous measurements, BL, V4, V6, V8

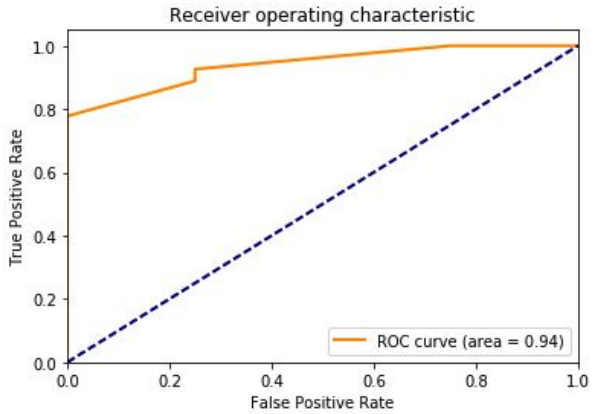
Note no filling process is occurring

Random Forest

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.806
- Stdev of Accuracy: 3.35*10^-16
- Median Accuracy: 0.806
- Average Feature Importance:

  BL - 0.177 V4 - 0.294 V6 - 0.275 V8 - 0.254

ROC Graph for Median trial

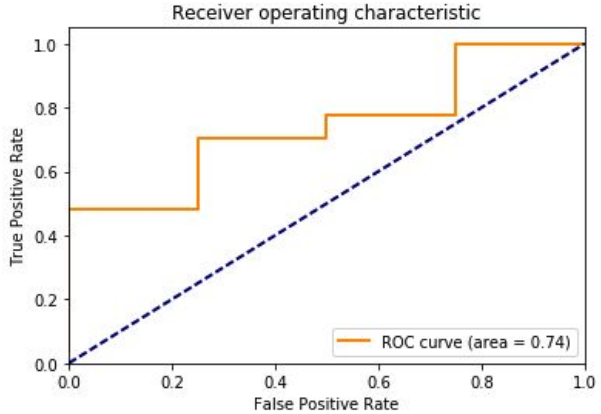


Adaboost

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.677
- Stdev of Accuracy: 4.46*10^-16
- Median Accuracy: 0.677
- Average Feature Importance:

  BL - 0.28 V4 - 0.21 V6 - 0.21 V8 - 0.3

ROC Graph for Median trial



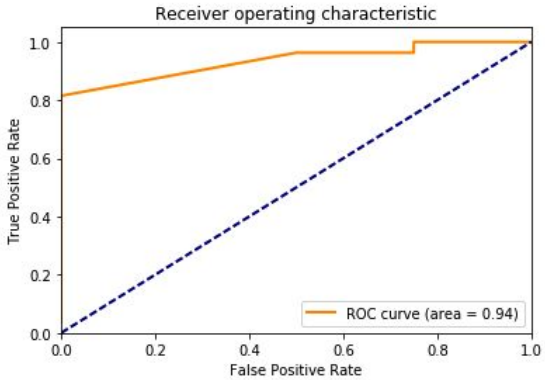
If we throw out the BL measurement, both classifiers improve!

Random Forest

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.839
- Stdev of Accuracy: 1.12*10^-16
- Median Accuracy: 0.839
- Average Feature Importance:

  V4 - 0.318 V6 - 0.378 V8 - 0.304

ROC Graph for Median trial

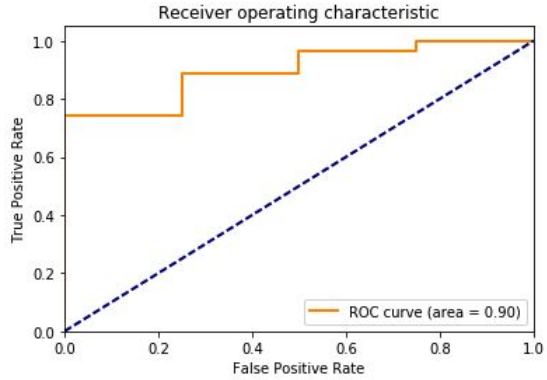


Adaboost

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.839
- Stdev of Accuracy: 1.12*10^-16
- Median Accuracy: 0.839
- Average Feature Importance:

  V4 - 0.301 V6 - 0.306 V8 - 0.383

ROC Graph for Median trial

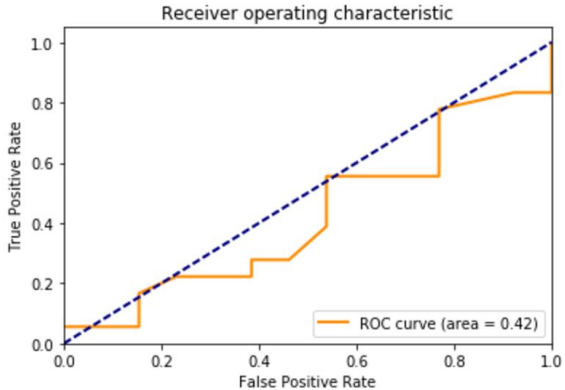


Finally let's test the classifiers out only using V4, V8, V10 on the data set where the labels were assigned randomly, to ensure the classifiers are indeed doing better than chance

Random Forest

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.419
- Stdev of Accuracy: 2.28*10^-16
- Median Accuracy: 0.419

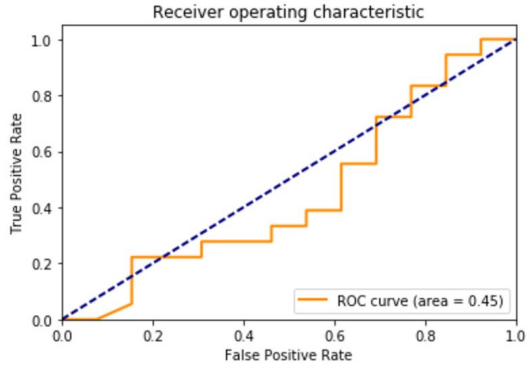
ROC Graph for Median trial



Adaboost

- Total Data Points Before Trying to Fill: 216
- Number of Training Data Points: 62
- Number of Testing Data Points: 31
- Mean Accuracy: 0.403
- Stdev of Accuracy: 0.016
- Median Accuracy: 0.387

ROC Graph for Median trial



It turns out the classifiers are indeed picking up on some significant signal