

CS 4372.501 Computational Methods For Data Scientists

HW 2

Instructor: *Anurag Nagar*

Organized by:
Jeremiah Joseph – jsj180002
Savishwa Gaur – sxx180113

Date: Oct 13, 2022

1. Data Set Introduction	4
2. Decision Tree Model	4
2.1 Base Model	4
2.1.1 Metrics	4
2.1.2 ROC/ Precision-Recall Curve	5
2.1.3 Tree Visualization	6
2.2 Tuned Model	6
2.2.1 Parameters Tuned	6
2.2.2 Parameter Tuning Results	6
2.2.3 Model Metrics	6
2.2.4 ROC/ Precision-Recall Curve	7
3. AdaBoost Model	9
3.1 Base Model	9
3.1.1 Metrics	9
3.1.2 ROC/ Precision-Recall Curve	10
3.1.3 Feature Importance	11
3.2 Tuned Model	11
3.2.1 Parameters Tuned	11
3.2.2 Parameter Tuning Results	11
3.2.3 Model Metrics	12
3.2.4 ROC/ Precision-Recall Curve	13
3.2.5 Overall Thoughts on Model	13
4. Random Forest Model	14
4.1 Base Model	14
4.1.1 Metrics	14
4.1.2 ROC/ Precision-Recall Curve	15
4.1.3 Feature Importance	16
4.2 Tuned Model	16
4.2.1 Parameters Tuned	16
4.2.2 Parameter Tuning Results	16
4.2.3 Metrics	16
4.2.5 Overall Thoughts on Model	18
5. XGBoost Model	19
5.1 Base Model	19
5.1.1 Metrics	19
5.1.2 ROC/ Precision-Recall Curve	21
5.1.3 Feature Importance	22
5.1.4 Tree Visualization	22

5.2 Tuned Model	23
5.2.1 Parameters Tuned	23
5.2.2 Parameter Tuning Results	23
5.2.3 Metrics	23
5.2.4 ROC/ Precision-Recall Curve	24
5.2.5 Overall Thoughts on Model	25
6. Conclusion	25

1. Data Set Introduction

The data set shows information of heart attack cases. The point of our model is to predict whether an individual had a heart attack or not based on certain conditions. There were 13 predictor attributes including ['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh', 'exng', 'oldpeak', 'slp', 'caa', 'thall', 'output']. In our preprocessing we removed chol and trtbps. To see more about this process check our preprocessing file.


2. Decision Tree Model

When running the decision tree model, we first made a model with the default parameters. We then used gridsearchcv to tune the hyperparameters. We evaluated the models with a variety of metrics that we will discuss in the following sections.

2.1 Base Model

2.1.1 Metrics

In the base model, I got an accuracy of 69%. This means our model predicted 69% of the testing data correctly. The precision of the score is the percentage that was correctly guessed as the class. From our model output, we can see that when predicting no heart attack, the model did worse than when predicting yes to heart attack. Both of these scores are fairly similar, however. The recall shows what percentage was right for each class over the total elements of the class. This also reveals fairly similar percentages between the classes with recall for no heart attack being higher than recall for heart attack. The f1 score shows the harmonic mean, and ideally, this number would be high.

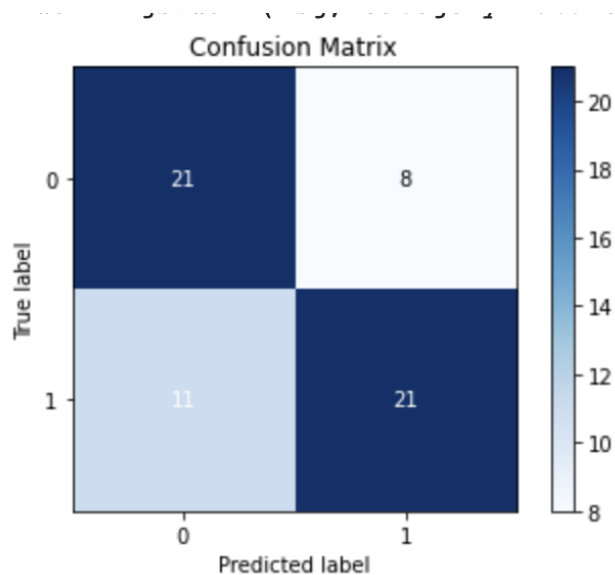


	precision	recall	f1-score	support
0	0.66	0.72	0.69	29
1	0.72	0.66	0.69	32
accuracy			0.69	61
macro avg	0.69	0.69	0.69	61
weighted avg	0.69	0.69	0.69	61

Predicted labels: [1 0 1 0 1 0 1 0 0 1 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0
1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1 0 1 1 0 0 0]

Accuracy: 0.6885245901639344

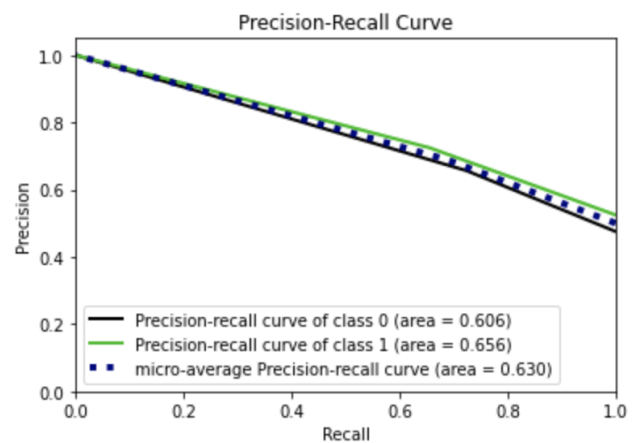
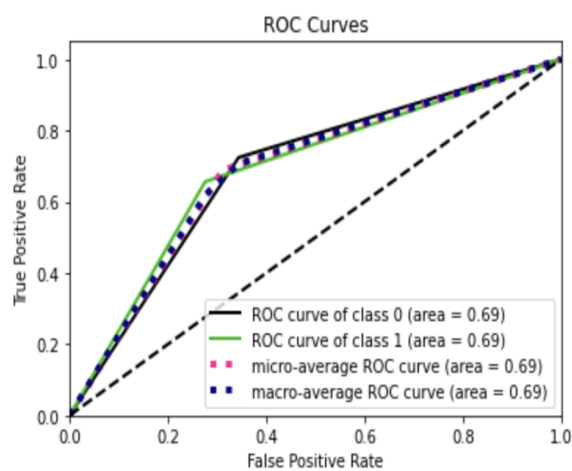
The confusion matrix shows that 21 correctly labeled no heart attack and 21 correctly labeled yes heart attack. There are 11 incorrect no heart attack predictions correctly and 8 incorrectly labeled positive heart attack predictions. As shown from the other data, this is pointing to the data being mostly even in accuracy among the 2 classes.



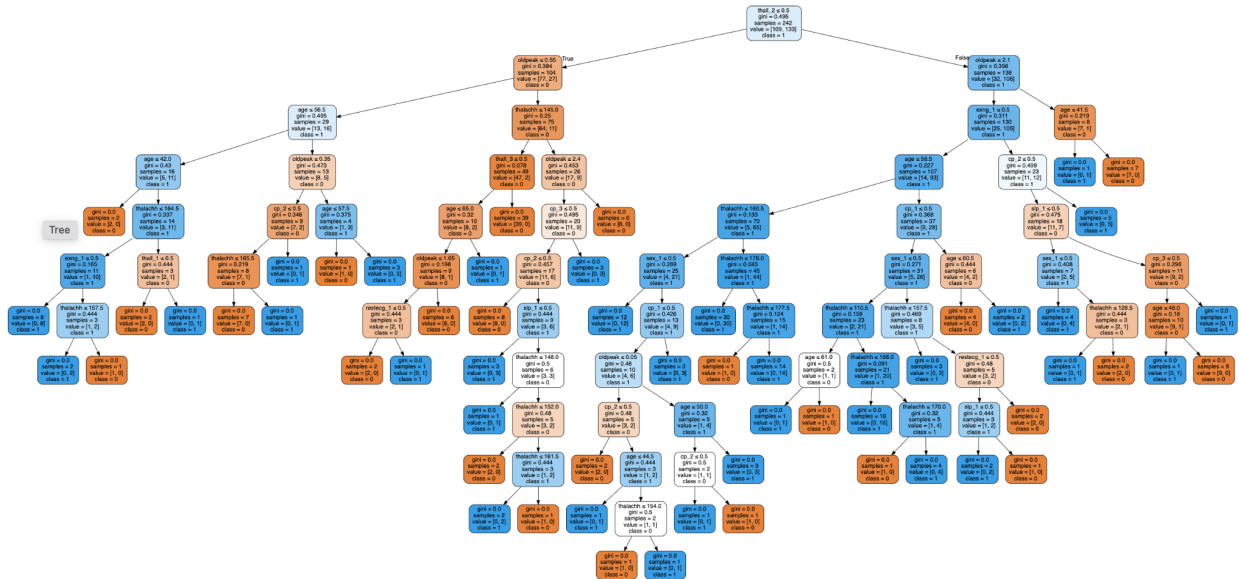
2.1.2 ROC/ Precision-Recall Curve

The ROC curve is a way to show the relationship between sensitivity and specificity. Ideally the curve will have a large area under the curve. The area under the curve in this case is .69 for both classes. This is a relatively good result. The diagonal line shows what would happen with random guessing, so there is a clear difference between that.

The Precision-Recall Curve shows the tradeoff between precision and recall. A high area under the curve would be preferable in this plot. For class 1(has heart attack) the area is .656 and for class 0(does not have heart attack) is .606. These are relatively good values for a simple baseline model.



2.1.3 Tree Visualization



2.2 Tuned Model

2.2.1 Parameters Tuned

The parameters that we chose to tune the model with are max depth, minimum samples split, and minimum samples leaf. The reason why we choose to try to adjust these parameters is because the original decision tree was so big. We felt this likely caused some overfitting. By tuning the model, we hope to try to eliminate some of his overfitting.

2.2.2 Parameter Tuning Results

The best parameter values were {'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2}. The best score was approximately 0.756. This represents the mean cross-validated score of the best_estimator.

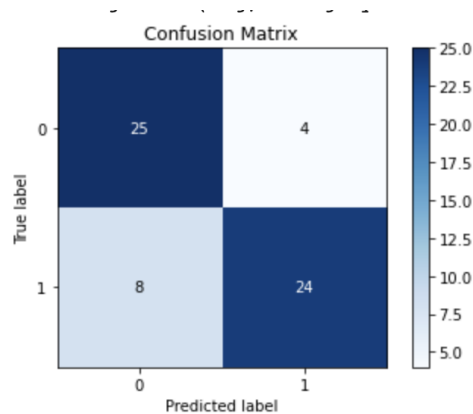
2.2.3 Model Metrics

The average accuracy score is about 80.3%. This represents a huge jump in accuracy from the base model. Similarly the precision, recall, and f1 scores are much higher than the previous model. Another thing to note is that the precision, recall and accuracy are all very similar numbers. This points to the model making balanced choices.

Accuracy Score: 0.8032786885245902

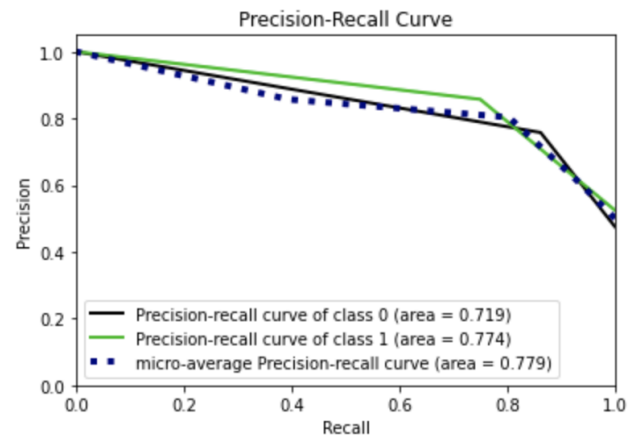
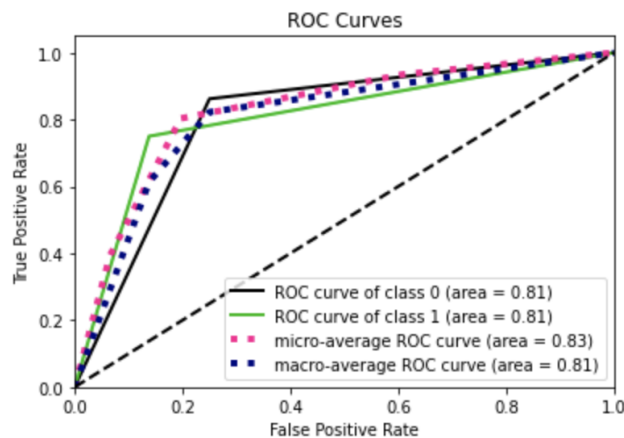
	precision	recall	f1-score	support
0	0.76	0.86	0.81	29
1	0.86	0.75	0.80	32
accuracy			0.80	61
macro avg	0.81	0.81	0.80	61
weighted avg	0.81	0.80	0.80	61

The confusion matrix shows that there were 25 correctly labeled negative heart attack values and 24 correctly labeled positive heart attack values. There are 8 incorrect predicted negative values and 4 incorrect positive values. Again these numbers are very close indicating a balance in the selections of the model.



2.2.4 ROC/ Precision-Recall Curve

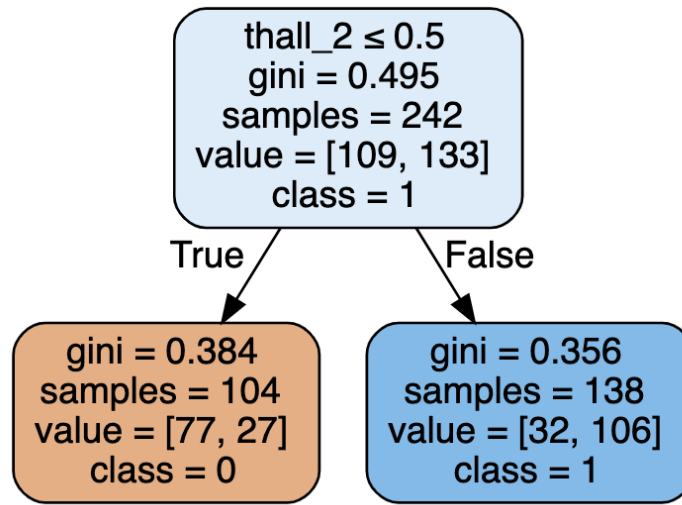
The ROC has an area under the curve of .81 for both classes. This is significantly higher than the baseline model. Similarly the precision-recall curve has a larger area under the curve for both cases than the baseline model.



2.2.5 Overall Thoughts on Model

The tuned model was clearly better performing than the baseline, default model. This is clear from all the metrics mentioned above as well as the ROC and Precision-Recall Curve. This further fuels the suspicion that the previous models were overfitting.

2.2.6 Tree Visualization



3. AdaBoost Model

When running the AdaBoost model, we first made a model with the default parameters. We then used `gridsearchcv` to tune the hyperparameters. We evaluated the models with a variety of metrics that we will discuss in the following sections.

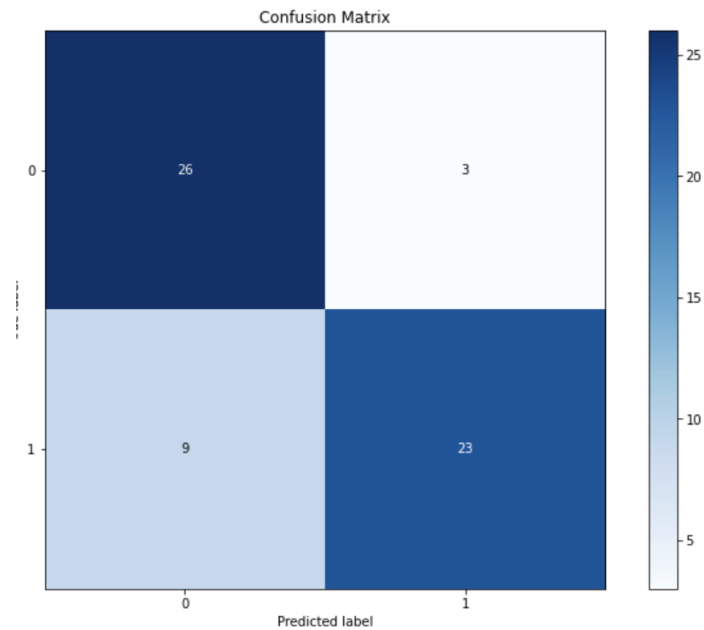
3.1 Base Model

3.1.1 Metrics

In the base model, I got an accuracy of approximately 80%. This means our model predicted 80% of the testing data correctly. The precision of the score is the percentage that was correctly guessed as the class. From our model output, we can see that when predicting not heart attack the model did worse than when predicting yes to heart attack. The recall shows what percentage was right for each class over the total elements of the class. The class of no heart attack did much better in this metric than the ones with heart attack. The f1 score shows the harmonic mean, and ideally, this number would be high.

Accuracy Score: 0.8032786885245902					
	precision	recall	f1-score	support	
0	0.74	0.90	0.81	29	
1	0.88	0.72	0.79	32	
accuracy			0.80	61	
macro avg	0.81	0.81	0.80	61	
weighted avg	0.82	0.80	0.80	61	

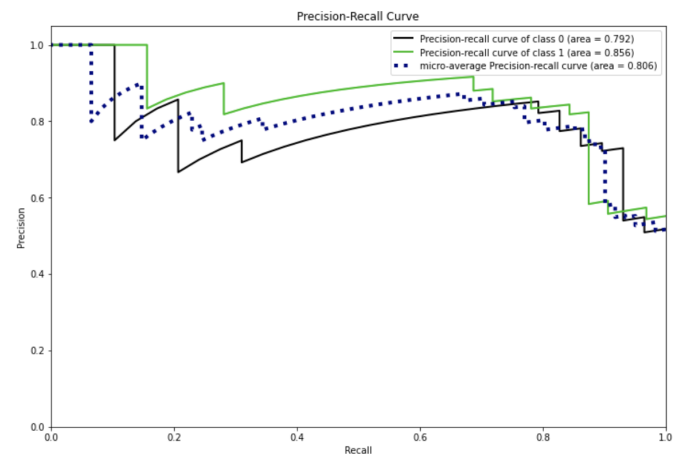
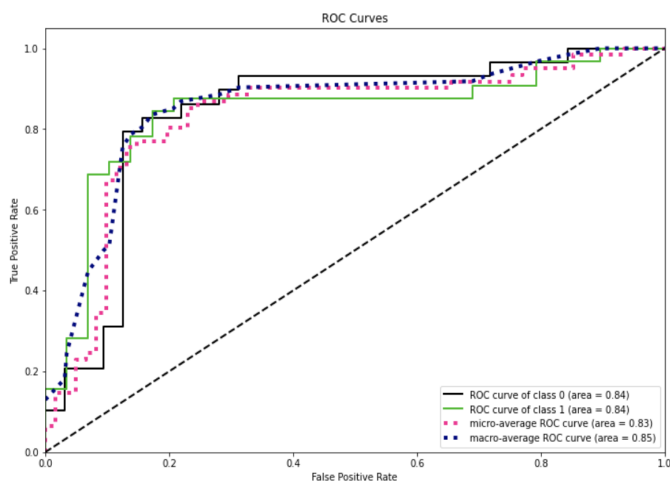
The confusion matrix shows that there were 26 correctly labeled negative heart attack values and 23 correctly labeled positive heart attack values. There are 9 incorrect predicted negative values and 3 incorrect positive values. This shows there seems to be more incorrect guesses that the model does not think is heart attack than there are incorrect guesses where the model thinks the person has cancer.



3.1.2 ROC/ Precision-Recall Curve

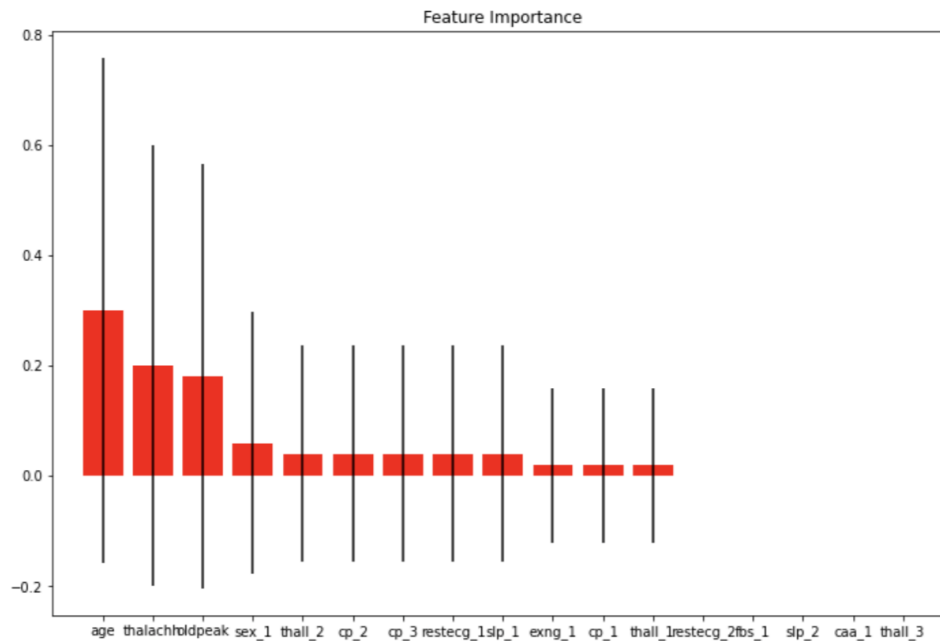
The ROC curve is a way to show the relationship between sensitivity and specificity. Ideally the curve will have a large area under the curve. The area under the curve in this case is .84 for both classes. This is a relatively good result. The diagonal line shows what would happen with random guessing, so there is a clear difference between that.

The Precision-Recall Curve shows the tradeoff between precision and recall. A high area under the curve would be preferable in this plot. For class 1(has heart attack) the area is .856 and for class 0(does not have heart attack) is .792. These are relatively good values for a simple baseline model.



3.1.3 Feature Importance

When looking at the features importance it looks like the most impactful according this model is age, and talach.



3.2 Tuned Model

3.2.1 Parameters Tuned

The parameters that we chose to tune the model with are n_estimator, learning rate, and algorithm. The reason why we choose to try to adjust these parameters is because we felt these would have large impacts on how well the model ran. We felt these values would provide a reasonable set of changes to compare with the base model.

3.2.2 Parameter Tuning Results

The best parameter values were {'algorithm': 'SAMME', 'learning_rate': 1.0, 'n_estimators': 50}. The best score was approximately 0.801. This represents the mean cross-validated score of the best_estimator.

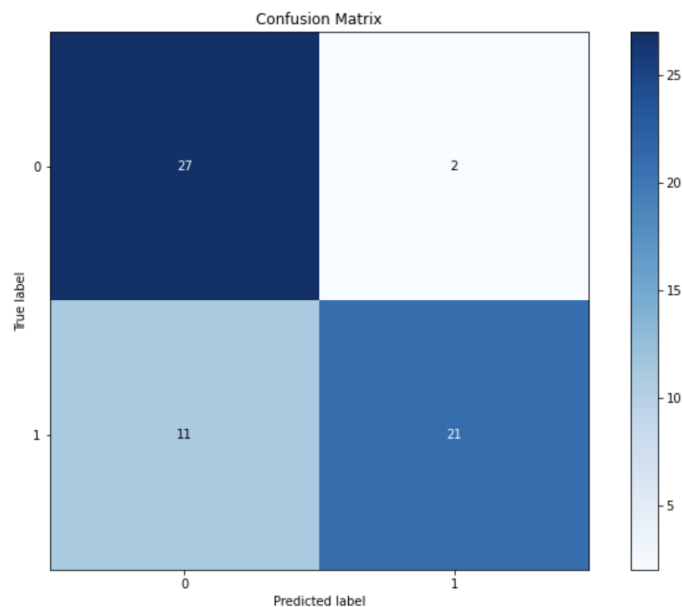
3.2.3 Model Metrics

The average accuracy score is about 78.7%. This represents a small decline in accuracy from the base model. Similarly the precision of class 0(no heart attack) and the precision class 1(heart attack) are also lower . Another thing to note is that the precision, recall and f-score have significant differences in the values.

Accuracy Score: 0.7868852459016393

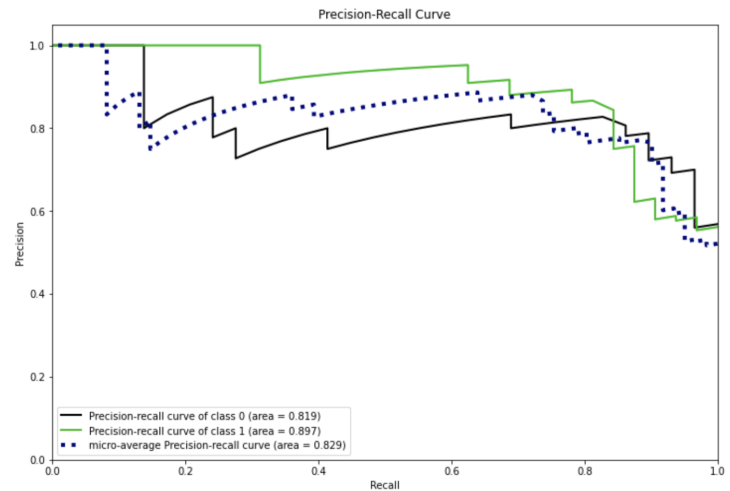
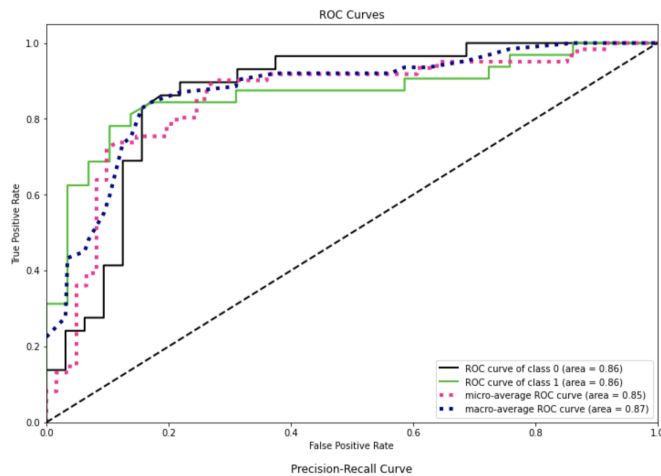
	precision	recall	f1-score	support
0	0.71	0.93	0.81	29
1	0.91	0.66	0.76	32
accuracy			0.79	61
macro avg	0.81	0.79	0.78	61
weighted avg	0.82	0.79	0.78	61

Looking at the confusion matrix we can see that the biggest problem in the model is predicting no heart attack when in reality there is a heart attack. In a real world use of this problem domain, this would be unacceptable, and probably the worst result. For this reason this points to this model having serious weaknesses.



3.2.4 ROC/ Precision-Recall Curve

The ROC has an area under the curve of .86 for both classes. This is slightly higher than the baseline model. Similarly the precision-recall curve has a larger area under the curve for both cases than the baseline model.



3.2.5 Overall Thoughts on Model

The tuned model seemed to perform worse than the baseline model. Though it had high area under the ROC area under the precision recall curve; the overall accuracy, precision, and recall were better in the default parameter model. Furthermore, the high value of positive heart attack cases that the model missed goes against a major priority of the problem domain.

4. Random Forest Model

We then moved to building a Random Forest Model. We looked at accuracy score, precision, recall, F1 score, ROC curves and feature importance plots to measure the performance of the model.

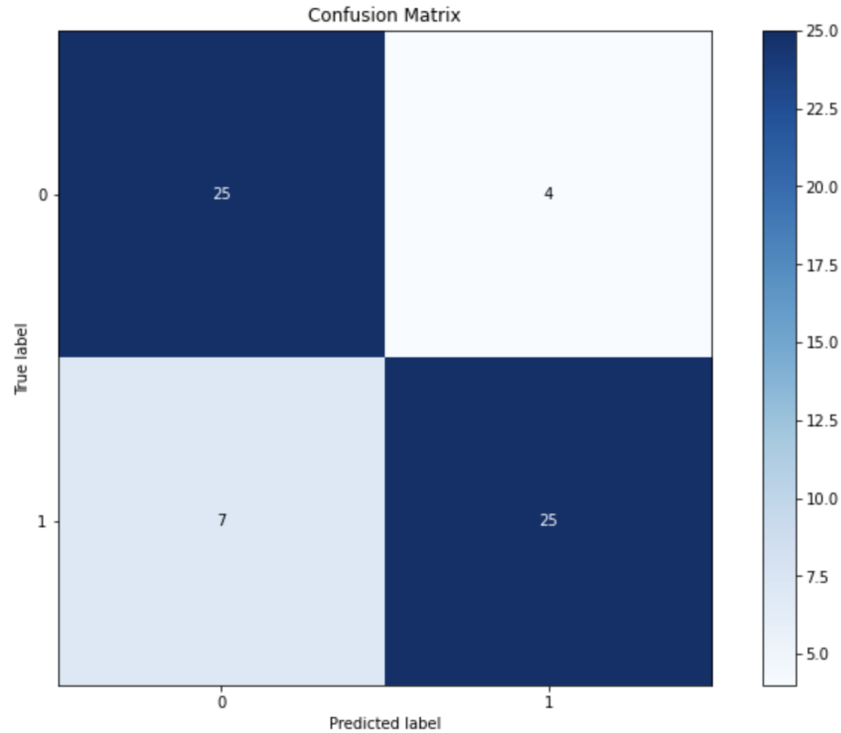
4.1 Base Model

4.1.1 Metrics

Looking at the classification report below, we can see that the accuracy of the base model is about 82%, which is good. Also we see that the base model has high precision for class 1, meaning that it more correctly classifies heart attacks than non-heart-attacks. More specifically, the model is correct about heart attack classifications 86% of the time. Looking at the recall, we see that the model has a rate of 78% for class 1, meaning out of all of the cases that have heart attacks the model is correct 78% of the time. The f1-score is .82 which is the harmonic mean of precision and recall, ideally, a higher f1-score is desirable. The support measures the occurrences of the class in our dataset, seeing that the numbers are very close we can conclude that our dataset is balanced.

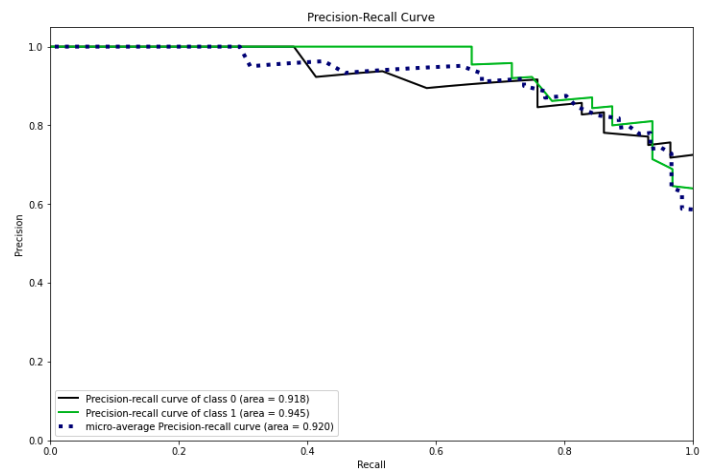
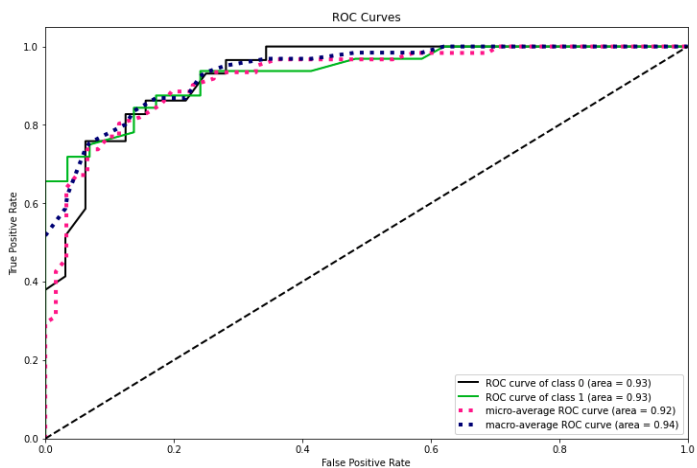
Accuracy Score: 0.819672131147541				
	precision	recall	f1-score	support
0	0.78	0.86	0.82	29
1	0.86	0.78	0.82	32
accuracy			0.82	61
macro avg	0.82	0.82	0.82	61
weighted avg	0.82	0.82	0.82	61

Looking at the confusion matrix below, we can see that the model had a high true negative and true positive rate, with a count of 25 for both. Also, the model only had 4 false positives and 7 false negatives.



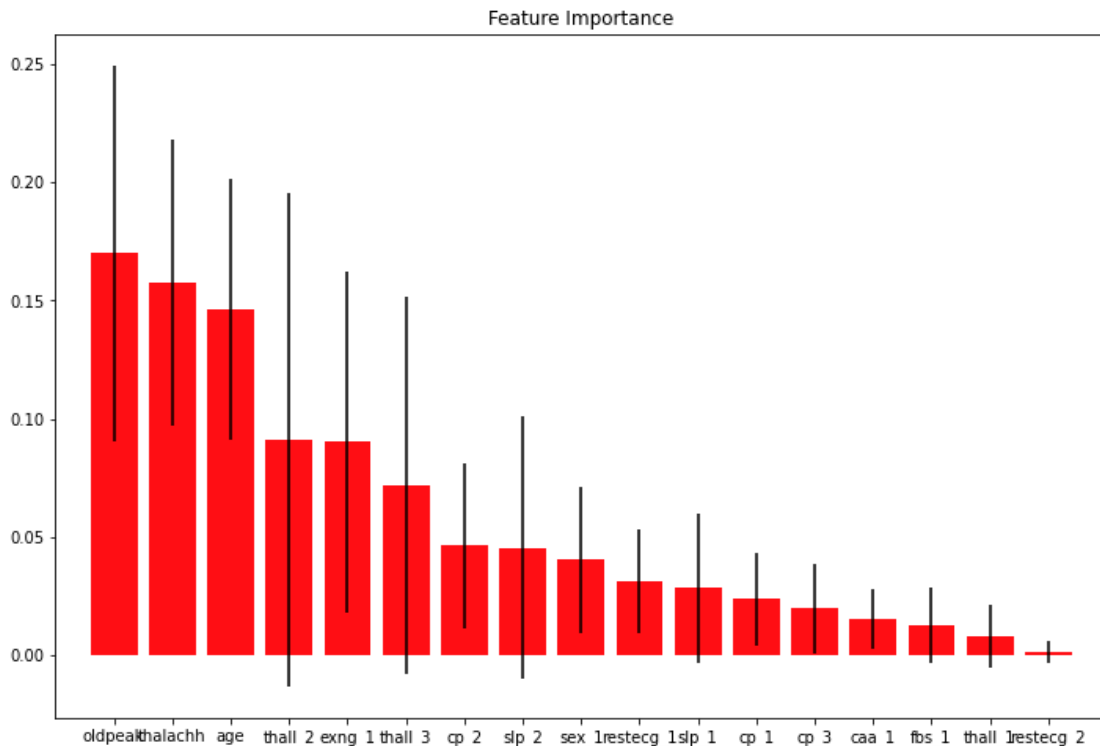
4.1.2 ROC/ Precision-Recall Curve

The Precision-Recall Curve shows the tradeoff between precision and recall. A high area under the curve would be preferable in this plot. For class 1 the area is .93 and for class 0 is .93. These are relatively good values for a simple baseline model. This shows a good balance of class values in our data set because the precision-recall curves for both classes are relatively similar



4.1.3 Feature Importance

Looking at the feature importance plot below, old peak is the most important feature and restecg_2 is the least important feature.



4.2 Tuned Model

4.2.1 Parameters Tuned

The parameters that we chose to tune the model with are max depth, and n_estimators. The reason why we choose to try to adjust these parameters is because the original random forest model was predicting too many false negatives, which is risky for our problem domain and suspected some overfitting occurring.

4.2.2 Parameter Tuning Results

The best parameter values were {'max_depth': 10, 'n_estimators': 600}. The best score was approximately 0.785. This represents the mean cross-validated score of the best_estimator.

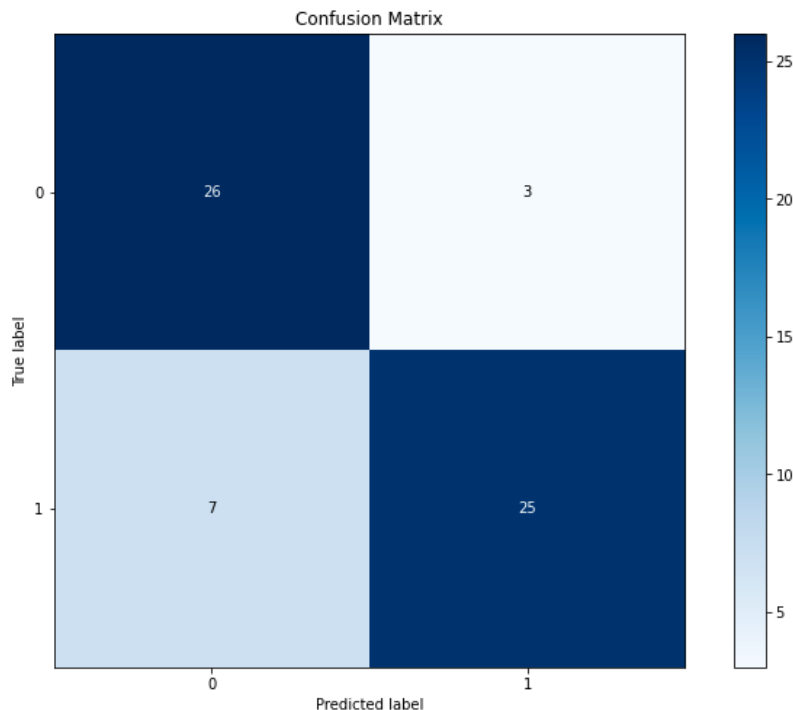
4.2.3 Metrics

Looking at the classification report below, we can see that the accuracy of the tuned model is about 83.6%, which is an improvement from the base model. Also we see that the

tuned model has high precision for class 1, meaning that it more correctly classifies heart attacks than non-heart-attacks. More specifically, the model is correct about heart attack classifications 89% of the time. Looking at the recall, we see that the model has a rate of 78% for class 1, meaning out of all of the cases that have heart attacks the model is correct 78% of the time. This is good because the precision and recall for both classes increased. The f1-score is .84 for class 0 and .83 for class 1 which is the harmonic mean of precision and recall, ideally, a higher f1-score is desirable, which was achieved from the base model. The support measures the occurrences of the class in our dataset, seeing that the numbers are very close we can conclude that our dataset is balanced.

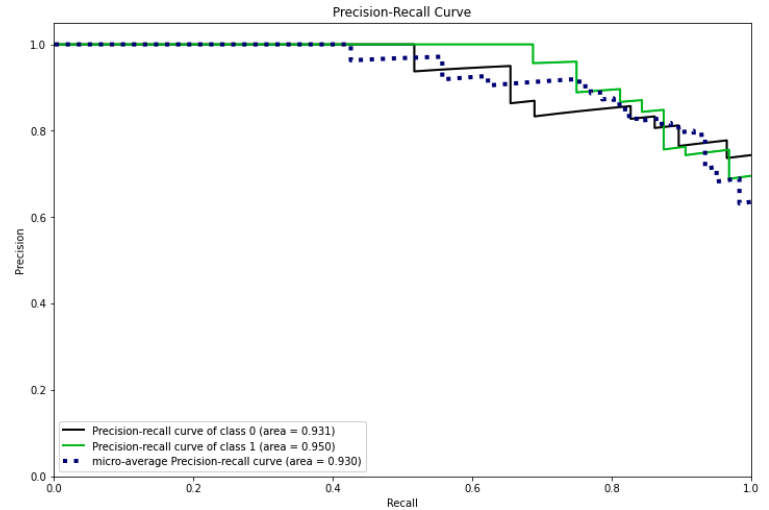
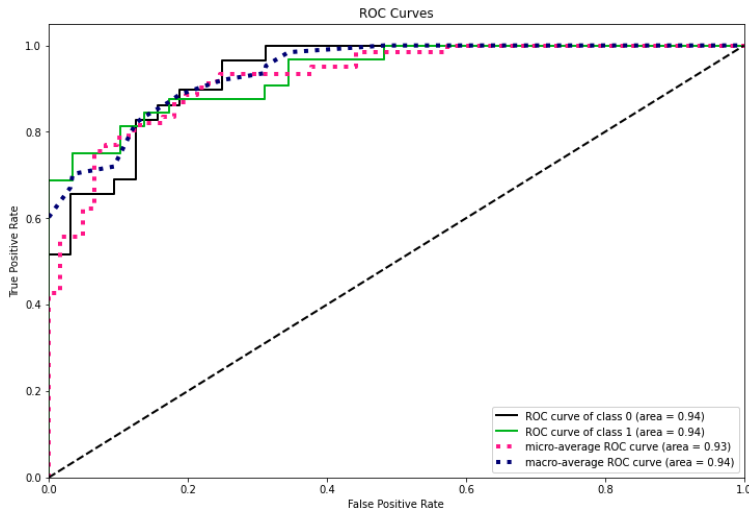
Accuracy Score: 0.8360655737704918				
	precision	recall	f1-score	support
0	0.79	0.90	0.84	29
1	0.89	0.78	0.83	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

Looking at the confusion matrix below, we can see that the model had a high true negative and true positive rate, with a count of 26 for class 0 and 25 for class 1. Also, the model only had 3 false positives and 7 false negatives.



4.2.4 ROC/ Precision-Recall Curve

The ROC has an area under the curve of .94 for both classes. This is significantly higher than the baseline model. Similarly the precision-recall curve has a larger area under the curve for both cases than the baseline model.



4.2.5 Overall Thoughts on Model

Overall, the tuned model performed better than the base model. The target metrics all increased to desirable measures, indicating a significant improvement. More specifically, the tuned model reduced the number of false positives and increased the number of true positives. Unfortunately, the tuned model did not decrease the number of false negatives, which as described earlier is risky for our problem domain.

5. XGBoost Model

We then moved to building an XGBoost Model. We looked at accuracy score, precision, recall, F1 score, ROC curves and feature importance plots to measure the performance of the model.

5.1 Base Model

5.1.1 Metrics

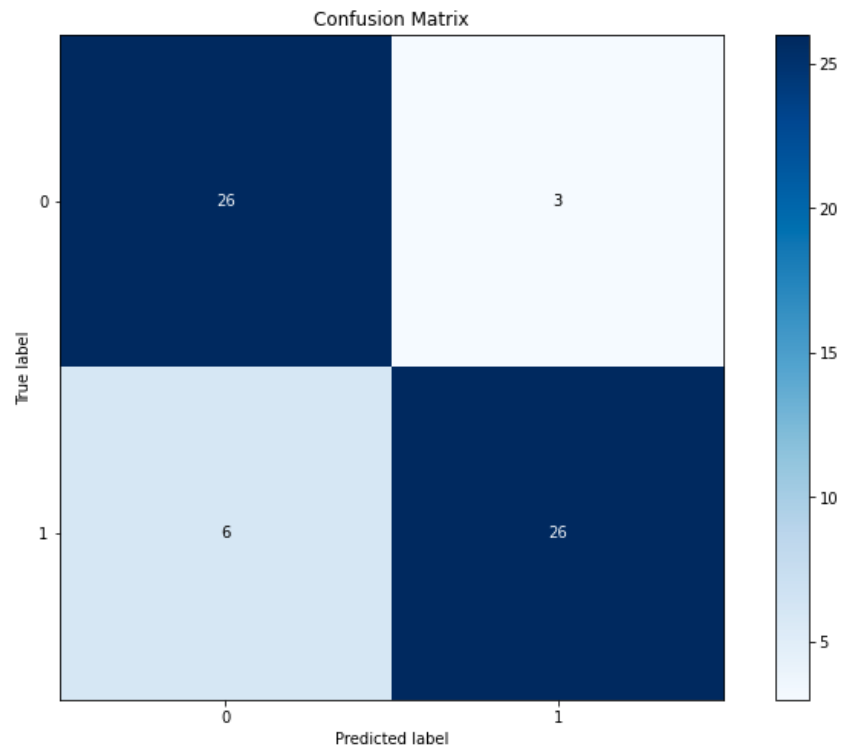
Looking at the classification report below, we can see that the accuracy of the base model is about 85%, which is good. Also we see that the base model has high precision for class 1, meaning that it more correctly classifies heart attacks than non-heart-attacks. More specifically, the model is correct about heart attack classifications 90% of the time. Looking at the recall, we see that the model has a rate of 81% for class 1, meaning out of all of the cases that have heart attacks the model is correct 81% of the time. The f1-score is .85 which is the harmonic mean of precision and recall, ideally, a higher f1-score is desirable. The support measures the occurrences of the class in our dataset, seeing that the numbers are very close we can conclude that our dataset is balanced.

```
Accuracy Score: 0.8524590163934426
              precision    recall  f1-score   support

     0       0.81         0.90         0.85         29
     1       0.90         0.81         0.85         32

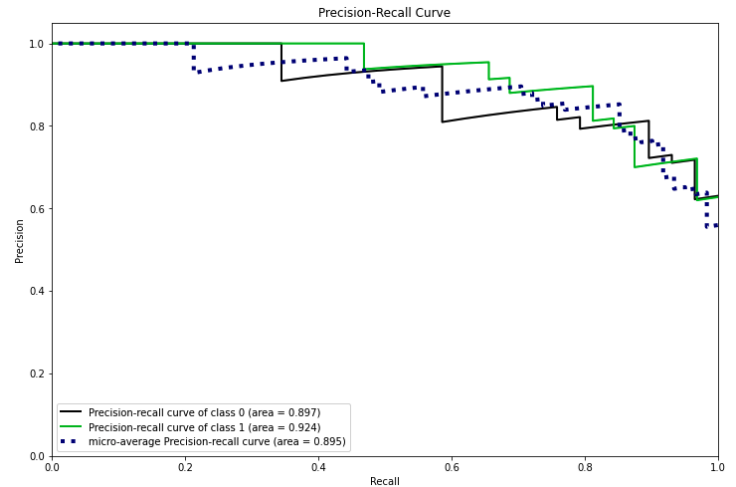
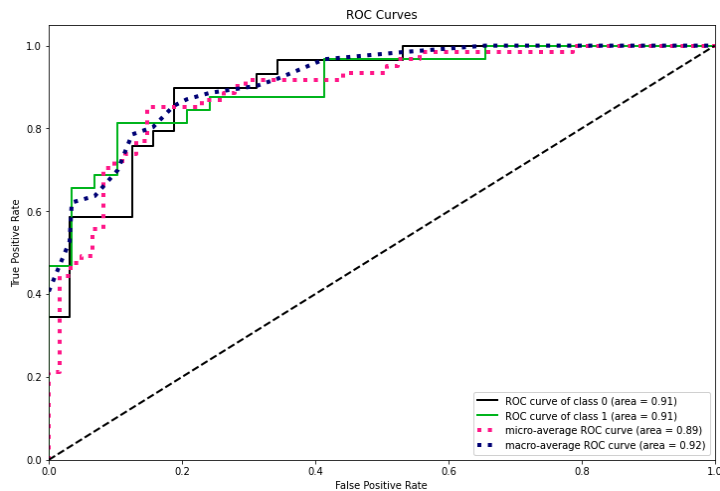
   accuracy                   0.85         61
  macro avg                   0.85         0.85         61
weighted avg                   0.86         0.85         61
```

Looking at the confusion matrix below, we can see that the model had a high true negative and true positive rate, with a count of 26 for class 0 and 25 for class 1. Also, the model only had 3 false positives and 6 false negatives.



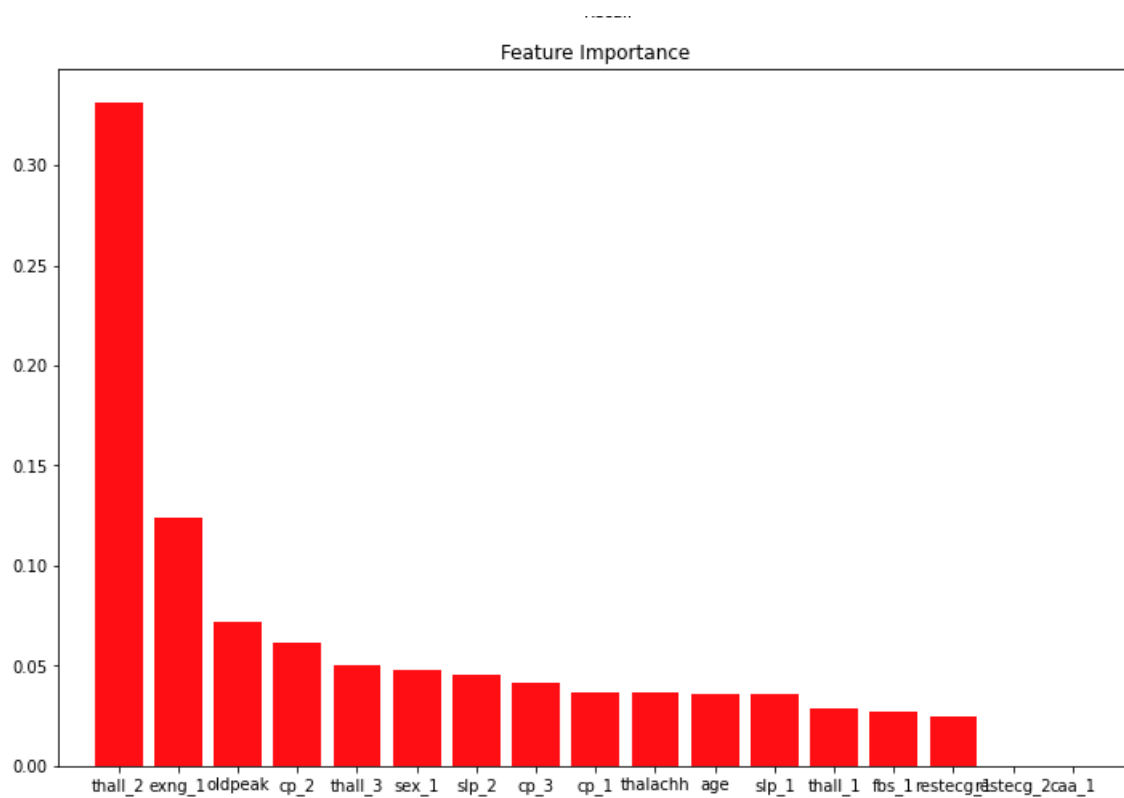
5.1.2 ROC/ Precision-Recall Curve

The Precision-Recall Curve shows the tradeoff between precision and recall. A high area under the curve would be preferable in this plot. For class 1 the area is .91 and for class 0 is .91. These are relatively good values for a simple baseline model. This shows a good balance of class values in our data set because the precision-recall curves for both classes are relatively similar

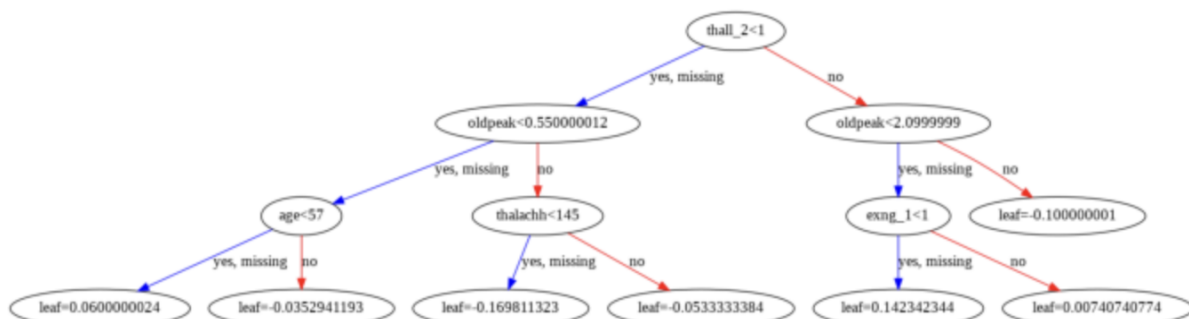


5.1.3 Feature Importance

Looking at the feature importance plot below, thall_2 is the most important feature and caa_1 is the least important feature.



5.1.4 Tree Visualization



5.2 Tuned Model

5.2.1 Parameters Tuned

The parameters that we chose to tune the model with are booster, max depth, and n_estimators. The reason why we choose to try to adjust these parameters is because the original xgboost model was predicting too many false negatives, which is risky for our problem domain and suspected some overfitting occurring.

5.2.2 Parameter Tuning Results

The best parameter values were{'booster': 'gbtree', 'max_depth': 1, 'n_estimators': 170}. The best score was approximately 0.822. This represents the mean cross-validated score of the best_estimator.

5.2.3 Metrics

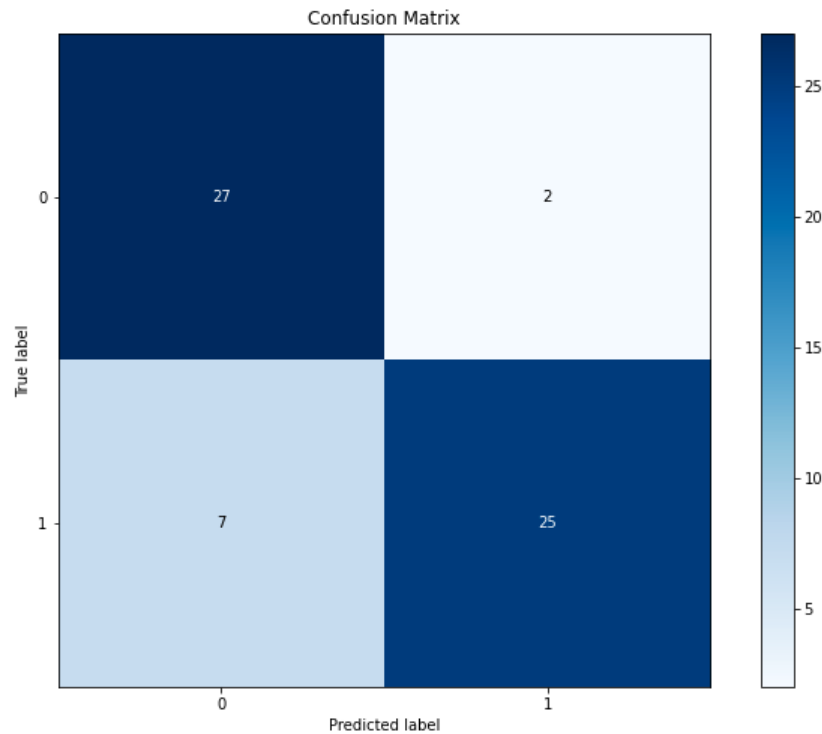
Looking at the classification report below, we can see that the accuracy of the tuned model is about 85.2%, which is an improvement from the base model. Also we see that the tuned model has high precision for class 1, meaning that it more correctly classifies heart attacks than non-heart-attacks. More specifically, the model is correct about heart attack classifications 93% of the time. Looking at the recall, we see that the model has a rate of 79% for class 1, meaning out of all of the cases that have heart attacks the model is correct 79% of the time. This is good because the precision and recall for both classes increased. The f1-score is .86 for class 0 and .85 for class 1 which is the harmonic mean of precision and recall, ideally, a higher f1-score is desirable, which was achieved from the base model. The support measures the occurrences of the class in our dataset, seeing that the numbers are very close we can conclude that our dataset is balanced.

```
Accuracy Score: 0.8524590163934426
              precision    recall  f1-score   support

     0       0.79       0.93       0.86         29
     1       0.93       0.78       0.85         32

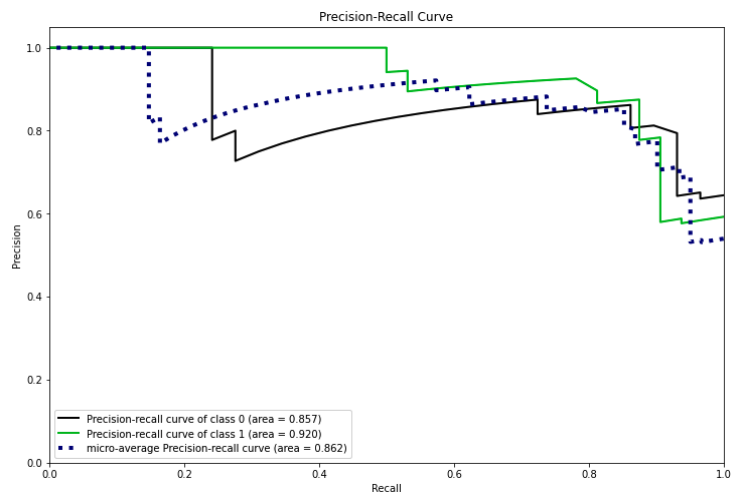
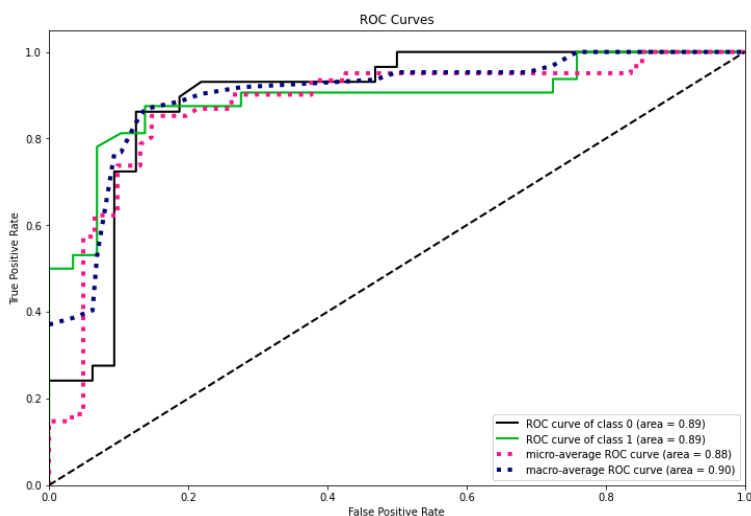
 accuracy                   0.85         61
 macro avg       0.86       0.86       0.85         61
 weighted avg    0.86       0.85       0.85         61
```

Looking at the confusion matrix below, we can see that the model had a high true negative and true positive rate, with a count of 27 for class 0 and 25 for class 1. Also, the model only had 2 false positives and 7 false negatives.



5.2.4 ROC/ Precision-Recall Curve

The ROC has an area under the curve of .89 for both classes. This is significantly lower than the baseline model. Similarly the precision-recall curve has a larger area under the curve for both cases than the baseline model. Also, we can see that the precision-recall curves for the two classes are very different, indicating that the tuned model seemed to be biased towards a single class, in this case being class 1.



5.2.5 Overall Thoughts on Model

Overall, the tuned model performed better than the base model if we look at simply the metrics. However, if we examine the balance of the model, the base was better: the precision-recall curves were much closer for the two classes and the auROC was larger in the base model. This indicates that the tuned model was more biased to classify patients to have heart attack, which is also shown in the confusion matrix as the false positive rate reduced. This is because, even though the model was classifying more patients as having heart attack, it was more accurate on those data points. Unfortunately, however, in doing this the tuned model increased the false negative rate which is very risky for our problem domain.

6. Conclusion

The goal of this project was to accurately predict heart attack in medical patients. This problem domain was tackled by using classification via 4 machine learning models: Decision Tree, Ada Boost, Random Forest, and XGBoost which were evaluated using accuracy, precision, recall, f1-score, ROC curves and feature importance plots.

Through our experimentation of these models, we saw that the base XGBoost and tuned Random Forest models were the best for our problem domain, because they had the highest scores in terms of the performance metrics. But more importantly, they had the lowest false negative rates. This is crucial because it is very risky to not predict heart attack in a patient that actually has it, making this the key metric for model evaluation. Also, the other models had a very large imbalance in the predictions to class 1, meaning they were predicting heart attack more often than not, but were not always very accurate. Ideally, this problem statement should be addressed with an ensemble of base XGBoost and the tuned Random Forest, taking a majority vote of the outputs of the two models. This would likely be an extension to this project.