Machine Learning Techniques for Heart Disease Diagnosis

By:

Max Chen

Dinidu Gunasena

Contents

1. Introduction

Heart disease is the leading cause of death for both men and women. About 610,000 people die of heart disease in the United States every year–that's 1 in every 4 deaths. Every year about 735,000 Americans have a heart attack. Of these, 525,000 are a first heart attack and 210,000 happen in people who have already had a heart attack. These harsh statistics motivate tons of research into detecting and preventing future heart disease related deaths. With medical data from previous heart disease patients, machine learning could help identify future cases of heart disease. The goal of this project is to develop an accurate classification model to predict a binary outcome of whether a patient has heart disease. We developed models using supervised learning methods such as Decision Trees, Neural Nets, Support Vector Machines (SVM), Random Forest, and XGBoost.

2. Related Work

Recently, there has been substantial research into trying to predict diseases in advance using machine learning techniques. These studies often try to mainly use data and information which can be gathered from the patient in quick and non-invasive ways, in order to create a quick, non-invasive diagnostic method. These types of studies have been performed for numerous types of diseases and conditions. For example, one fairly recent study tried to diagnose Alzheimer's from MRI and cognitive test data [1]. Another study tried to diagnose diabetes solely from urine samples, using random forest, SVM and neural network algorithms [2]. These recent studies have focused on using a different set of features, which are 'easier' to find, and ultimately give better results when used with machine learning algorithms

However, there have been many more studies which are more directly concerned with whether machine learning can be used to diagnose heart disease. One study we explicitly stated the many ways in which machine learning could be applied currently [3]. Other studies have worked to identify specific conditions, such as a study which the SVM algorithm to discover if significant coronary artery disease was present [4]. Ischemic heart disease. Many studies used machine learning algorithms (with all at least employing neural nets), in order to diagnose the Ischemic heart disease [5] [6] [7]. Other studies considered heart failure remission, comparing several machine learning methods including random forest, boosting and support vector machines [8] [9].

More recently, studies have moved on from just trying to determine if machine learning was useful, to what algorithm (and what parameters) was most effective. In one 2015 study, the Naive Bayes algorithm, Decision tree, Support Vector Machine, Logistic regression, Neural net, and adaboost algorithms were compared against each other [10]. This study found that adaboost algorithm performed significantly well, getting a maximum accuracy of 90%. Another study compared support vector machines, neural net algorithm and naive Bayes classifier, and also combined these classifiers into an ensemble classifier [11]. The paper found that the ensemble classifier always performed better than the stand-alone classifiers. An older study from 2008 also investigated ensemble machine learning methods (bagging and random forest) and compared these classification methods to more standalone algorithms (support vector machines and logistic regression) [12]. This study found that tree-based ensemble methods performed well, but they still did not perform significantly better than logistic regression methods. A more recent study in 2017 compared support vector machines, neural net algorithms and

the random forest ensemble classifier against each other [13]. This study shows, like the other papers, that ensemble methods work best, with random forest getting over 90% accuracy. There are also other papers, which made investigations into machine learning methods without comparing against ensemble methods. [14] [15] These papers compared decision trees, naïve bayes, neural nets, K-nearest neighbor, support vector machines and logistic regression methods, with 1 paper finding support vector machines having higher accuracy, while the other paper found that logistic regression methods were more accurate. Overall, however, recent research find that ensemble methods perform best in determining heart disease conditions.

3. Dataset

3.1 Dataset and features

The data was originally supplied by the Cleveland Heart Disease Database via the UCI Machine Learning repository, but we received it from the DrivenData website as part of their competition. The data includes non-invasive information which is used to find if a person has heart disease through supervised learning. The actual dataset has 13 attributes, with 270 examples (180 in the training dataset, and 90 in the test dataset).

The first attribute (slope_of_peak_exercise_st_segment) shows an integer value which represents the slope of the peak exercise st-segment. It was captured using an electrocardiography measurement, and it indicates the quality of blood flow to the heart. The second feature (thal) captures the results of thallium stress test, which also indicate blood flow to the heart. This particular value can have three string values: normal, fixed_defect and reversible_defect. The next feature considered in the dataset is the resting blood pressure, which is represented by a simple integer value. The fourth attribute (chest_pain_type) shows the chest pain type, which is represented by 4 integer values. No other data was given about this feature from either the competition, nor the UCI machine learning repository. The next feature (num_major_vessels) captures the number of major blood vessels colored by fluoroscopy. It can be a value from 0 to 3. The sixth feature (fasting_blood_sugar_gt_120_mg_per_dl) shows a Boolean value which measures if the patient has a fasting blood pressure greater than 120 milligrams per deciliter. The seventh attribute in this dataset (resting_ekg_results) captures the resting electrocardiographic results, and can be a value from 0 to 2. The next attribute (serum_cholesterol_mg_per_dl) shows the serum cholesterol in milligrams per deciliter, which is represented in an integer value. The ninth feature (oldpeak_eq_st_depression) is a measure of abnormality in electrocardiograms. The actuals value is a float, where the condition is induced by exercise relative to a rest. The tenth feature (sex) is a binary value which gives the sex of a patient as 0 for female, and 1 for male. The eleventh feature (age) is the measure of the age of the patient. The next attribute (max_heart_rate_achieved) shows the maximum heart rate achieved, which is represented as an integer value in beats per minute. The final feature (exercise_induced_angina) shows a Boolean value for exercise induced angina. If it is 1, then the patient has exercise induced chest pain.

FEATURES

1. Slope_of_peak_exercise_st_segment
2. Thal
3. Resting_blood_pressure
4. Chest_pain_type
5. Num_major_vessels
6. Fasting_blood_sugar_gt_120_mg_per_dl
7. Resting_ekg_results
8. Serum_cholesterol_mg_per_dl
9. Oldpeak_eq_st_depression
10. Sex
11. Age
12. Max_heart_rate_achieved
13. exercise_induced_angina

3.2 Pre-processing

There is very little pre-processing that needs to be done for this dataset. The only variable that is not already in numerical form is in the second column of the data, which is used to represent the results of a thallium stress testing. The 3 different values for this test are normal, fixed defect, and reversible defect. The data in this column is then changed so that the value 0 is substituted for normal, 1 for fixed defect, and then 2 for reversible defect. Otherwise, the data will be scaled before its used in any algorithm.

The competition splits the 270 examples into 180 in the training dataset, and 90 in the test dataset. The training dataset contains labels, while the test dataset does not. The given training dataset will be split into 70% real training (126) and 30% validation datasets (54). After the creation and tuning of this model, it will then run against the given test dataset.

4. Experimental Evaluation

4.1 Methodology

4.1.1 Algorithm details

Decision Tree Learning:

Uses a decision tree as a predictive model. Each branching results from learning simple decision rules inferred from data features. The leaf nodes represent the conclusions on a target value. In order to choose the "best" split at each branch, information gain based on the concept of entropy is used as the metric where $p_i \dots p_j$ are fractions that add up to 1 and represent the percentage of each class present in the child node

$$H(T) = -\sum_{i=1}^{J} p_i log_2 p_i$$

Information gain is then entropy of the parent node subtracted by the weight sum of entropy of the children. [16]

$$IG(T, a) = H(T) - H(T|a)$$

Neural Net:

This algorithm learns a function (f) by training on a dataset, where (m) is the number of dimensions for input and (o) is the number of dimensions for output. [16]

$$f(\cdot): R^m \rightarrow R^o$$

Given a set of attribute features and a target, it can learn a non-linear function approximator. In between the input and output layer, there are one or more non-linear layers called hidden layers. Each neuron in the hidden layer changes values from the previous with a weighted linear summation followed by a non-linear activation function such as hyperbolic tan (g). [16]

$$g(x) = \frac{e^x - e^{-x}}{e^x - e^{-x}}$$

It trains these weights (w) through stochastic gradient descent, where (η) is the learning rate which controls the step-size in the parameter space search and ($Loss$) is the loss function used for the network. [16]

$$w \leftarrow w - \eta(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w})$$

SVM:

In support vector machines, a data point is viewed as a p-dimensional vector and we want to know whether we can separate such points with a (p - 1) dimensional hyperplane. This is a linear classifier. Often, data can turn out to be not linearly separable in that space. In this case, the original finite-dimensional space should be mapped to a higher one. Mappings can be defined in terms of a kernel function $k(x, y)$. [16]

Ensemble Learning:

Ensemble learning combines the predictions of several base estimator built with a given learning algorithm in order to improve generalizability.

Random Forest:

Random forest is based on ensemble learning and uses a bagging technique. The principle behind it is to create multiple estimators independently and then to average their predictions. Each tree in this ensemble is built from a bootstrap (drawn with replacement) sample from the training set. When splitting a node during construction, the split is chosen based on the best split of a random subset of features. From this randomness, the bias of the forest slightly increase, but by averaging, variance should be reduced. [19]

XGBoost:

In boosting, trees a built in which each subsequent tree tries to reduce the errors of the previous tree. Gradient boosting is applied to minimize the loss when adding new models. These base learners are weak learners in which the bias is high. The predictions from all of them are then combined though a weighted sum to produce the final prediction. [18]

4.1.2 Performance Metrics

The performance criteria for each heart disease classifier is the accuracy on correctly predicting the validation set. [16] [17]

- Tp - number of true positives
- Fp - number of false positives
- Tn - number of true negatives
- Fn - number of false negatives

Accuracy is the ratio (Tp + Tn) / (# number of datapoints) and is the ability of the classifier to correct label either positive or examples. The precision is the ratio Tp / (Tp + Fp). It is the ability of the classifier to not label positive examples as negative. The recall is the ratio Tp / (Tp + Fn). It is the ability of the classifier to find all the positive samples. The F-1 score is a weighted harmonic mean of the precision and recall, where F-1 scores are at its best value at 1 and worst value at 0.

The first step is to develop a model for testing the accuracy on the training set. Using the Grid Search CV technique a set of parameters were tested on each classifier to see which combination gave the best accuracy result. Using these models, the validation set was then tested for the accuracy.

4.2 Results

Decision Tree:

After applying the decision tree model, its results are shown in Table 1. The accuracy score on the validation set is 74%, while the training data set is a higher 87%. The training set performed way better than the validation set and it could be attributed to overfitting. It is notable that the decision tree classifier is not below 80%. The “thal” attribute was the most important feature in the decision tree classifier, and had double the weight of the previous attribute. The degree at which it outstrips the other features may be a result of how the sampled dataset was split into test and validation sets. The ROC curve shows that our model does better than the base benchmark of random guessing with an area of 79%.

Decision Tree Classification Report (validation):

	precision	recall	f1-score	support
0	0.86	0.71	0.77	34
1	0.62	0.80	0.70	20
avg / total	0.77	0.74	0.75	54

Figure 1: Decision Tree with optimal parameters: max depth = 5, minimum samples per leaf=5, minimum samples split=2 and minimum weight for fraction of a leaf =0.0

	ACCURACY SCORE
TRAINING DATASET	0.873015873015873
VALIDATION DATASET	0.7407407407407407

Table 1: comparison of the accuracy of the Decision Tree classifier when used on the dataset it was trained on, vs the validation dataset

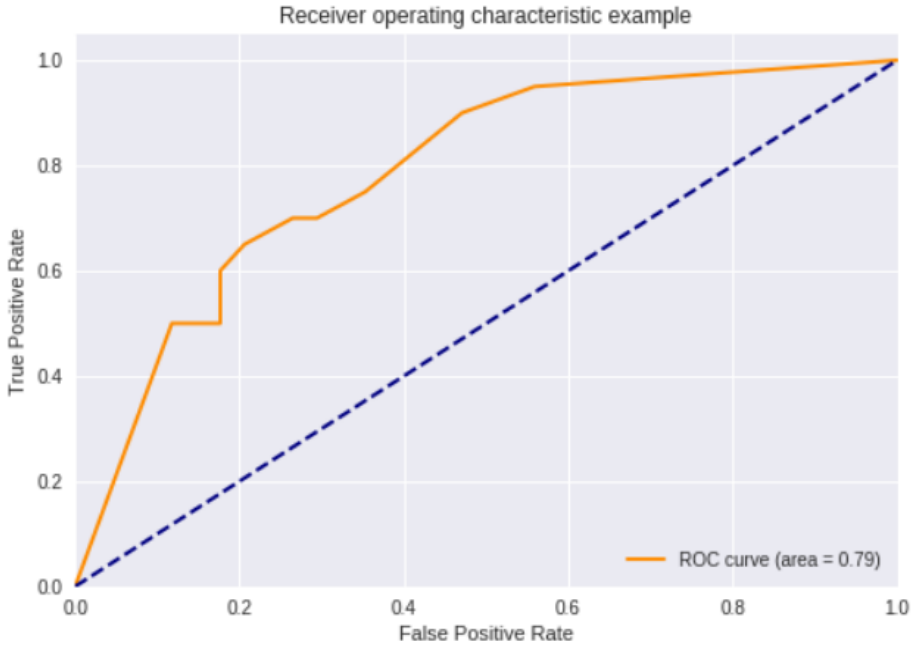

Figure 2: ROC curve which outlines the performance of the decision tree algorithm when used for the validation set

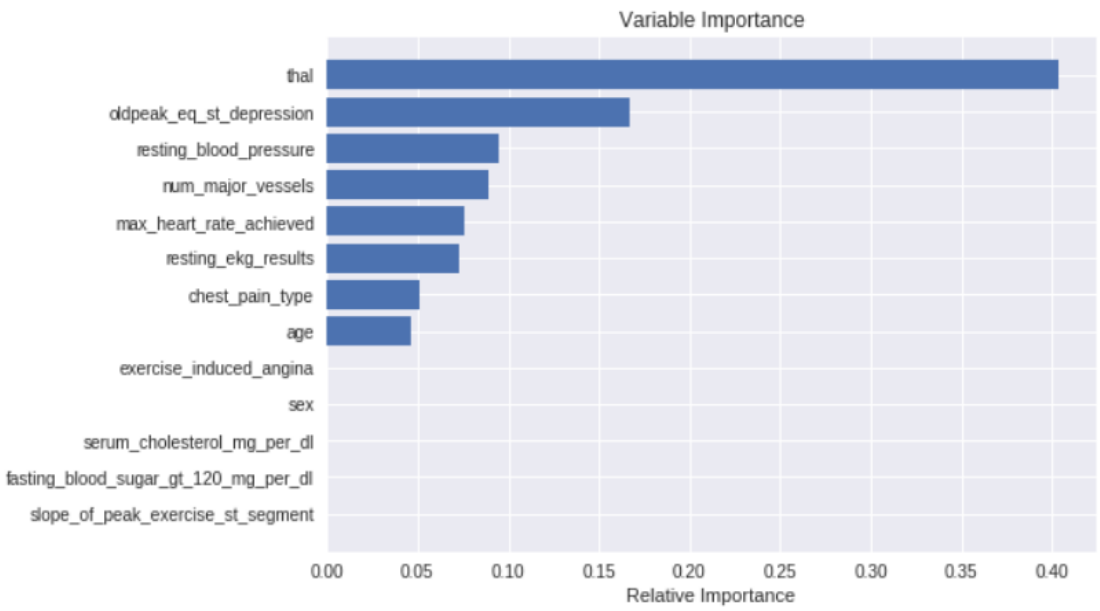

Figure 3: Bar graph which illustrates which features were most important when classifying with the decision tree classifier

Neural Net:

After applying the neural net model, its results are shown in Table 2. The accuracy score on the validation set is 81%, while the training data set is a higher 88%. The model seems to perform well on the validation set and is 9% less accurate than the training set. There is some overfitting, but overall the modularity of deep learning helps. The ROC curve shows that our model does better than the base benchmark of random guessing with an area of 90%.

Neural Net Classification Report (validation):

	precision	recall	f1-score	support
0	0.88	0.82	0.85	34
1	0.73	0.80	0.76	20
avg / total	0.82	0.81	0.82	54

Figure 4: Neural Net classifier with optimal parameters: activation function = tanh, alpha = 0.01, hidden layer sizes = (10,5,5,5), learning rate type = constant

	ACCURACY SCORE
TRAINING DATASET	0.8888888888888888
VALIDATION DATASET	0.8148148148148148

Table 2: comparison of the accuracy of the Neural Net classifier when used on the dataset it was trained on, vs the validation dataset

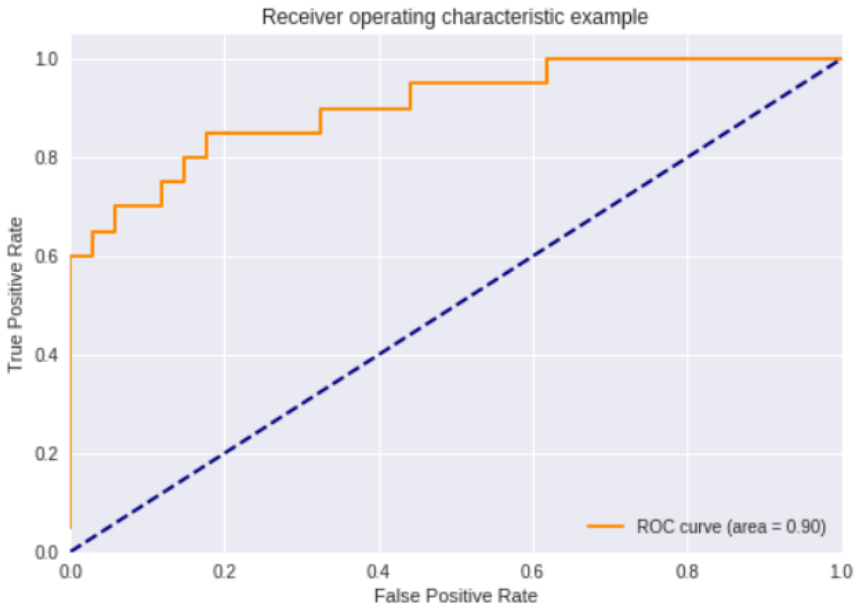

Figure 5: ROC curve which outlines the performance of the neural net algorithm when used for the validation set

SVM:

After applying the support vector machine model, its results are shown in Table 3. The accuracy score on the validation set is 83%, while the training data set is a slightly higher 84%. The model seems to perform very well on the validation set and is only 1% less accurate than the training set. The ROC curve shows that our model does better than the base benchmark of random guessing with an area of 91%.

Support Vector Machine Classification Report (validation):

	precision	recall	f1-score	support
0	0.90	0.82	0.86	34
1	0.74	0.85	0.79	20
avg / total	0.84	0.83	0.84	54

Figure 6: Support Vector Machine classifier with optimal parameters: C= 0.6, degree = 1, gamma = 0.001, kernel = linear, max_iterations = 1000

	ACCURACY SCORE
TRAINING DATASET	0.849063492063492
VALIDATION DATASET	0.8333333333333334

Table 3: comparison of the accuracy of the Support Vector Machine classifier when used on the dataset it was trained on, vs the validation dataset

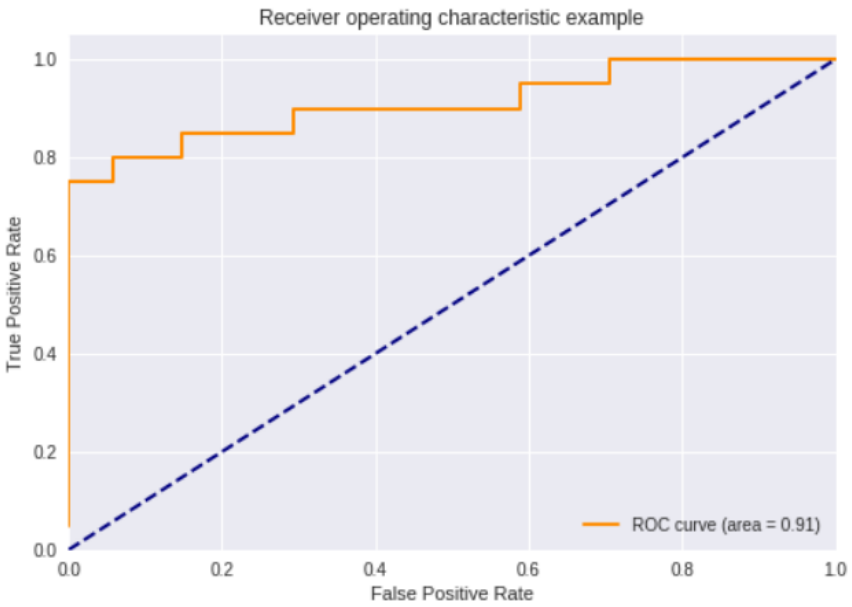

Figure 7: ROC curve which outlines the performance of the Support Vector Machine algorithm when used for the validation set

Random Forest:

After applying the random forest model, its results are shown in Table 4. The accuracy score on the validation set is 79%, while the training data set is a higher 88%. The model seems to perform well on the validation set and is 9% less accurate than the training set. There is some overfitting occurring. The ROC curve shows that our model does better than the base benchmark of random guessing with an area of 91%. The "max_heart_rate_achieved" and "thal" attributes were the most important feature in the decision tree classifier, and there seems to be gradual increase of importance with no feature dominating.

Random Forest Classification Report (validation):

	precision	recall	f1-score	support
0	0.85	0.82	0.84	34
1	0.71	0.75	0.73	20
avg / total	0.80	0.80	0.80	54

Figure 8: Random Forest classifier with optimal parameters: maximum features = 0.4, maximum samples = 0.9, n estimators = 20

	ACCURACY SCORE
TRAINING DATASET	0.8809523809523809
VALIDATION DATASET	0.7962962962962963

Table 4: comparison of the accuracy of the Random Forest classifier when used on the dataset it was trained on, vs the validation dataset

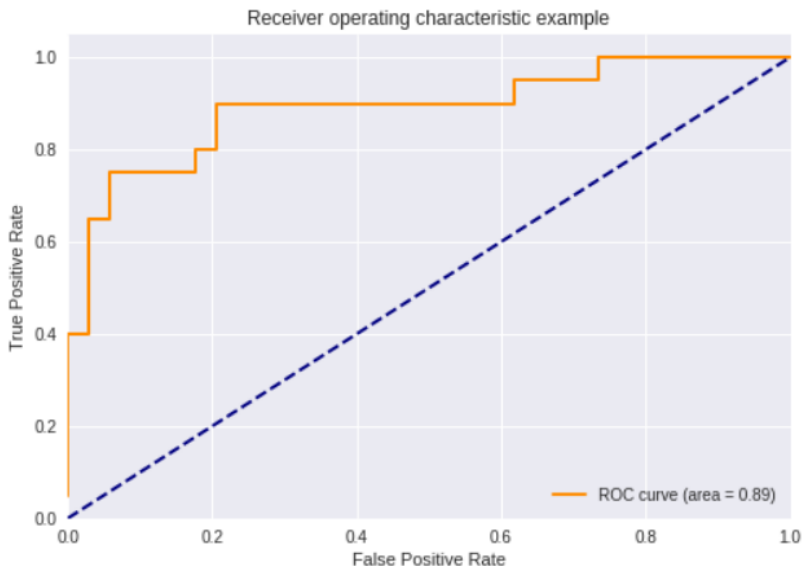

Figure 9: ROC curve which outlines the performance of the Random Forest algorithm when used for the validation set

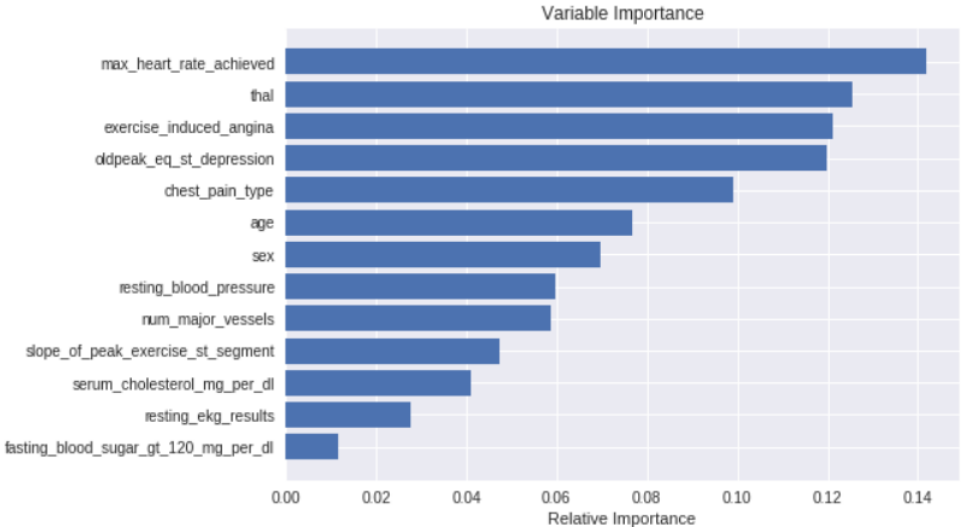

Figure 10: Bar graph which illustrates which features were most important when classifying with the Random Forest classifier

XGBoost:

After applying the XGBoost, its results are shown in Table 5. The accuracy score on the validation set is 85%, while the training data set is a lower 84%. The model seems to perform very well

on the validation set and is 1% more accurate than the training set. The ROC curve shows that our model does better than the base benchmark of random guessing with an area of 93%.

XGBoost Classification Report (validation):

	precision	recall	f1-score	support
0	0.88	0.88	0.88	34
1	0.80	0.80	0.80	20
avg / total	0.85	0.85	0.85	54

Figure 11: XGboost classifier with optimal parameters: booster= gblinear, learning rate = 0.2, max delta step = 0.0001, n-estimators=5

	ACCURACY SCORE
TRAINING DATASET	0.8412698412698413
VALIDATION DATASET	0.8518518518518519

Table 5: comparison of the accuracy of the XGboost classifier when used on the dataset it was trained on, vs the validation dataset.

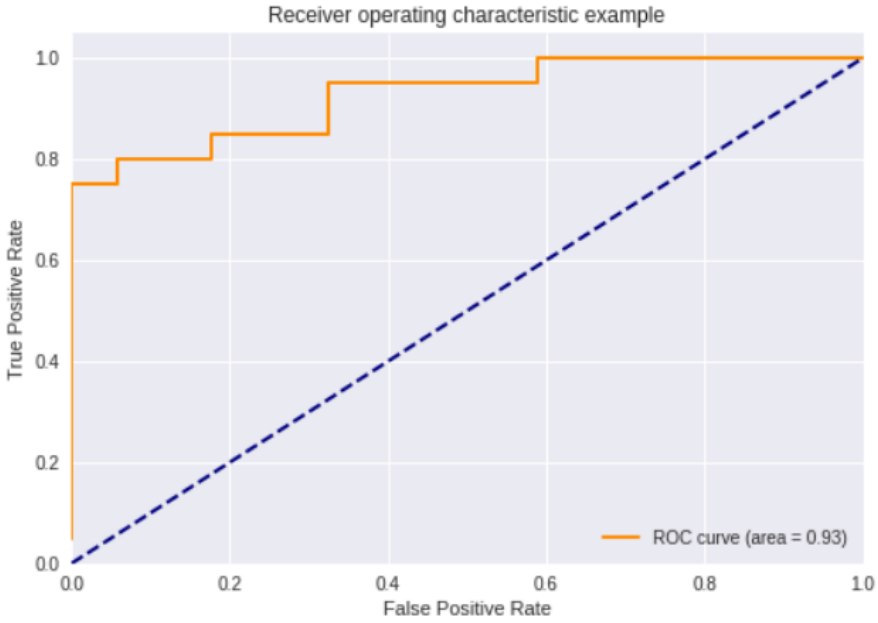

Figure 12: ROC curve which outlines the performance of the XGboost algorithm when used for the validation set

4.3 Discussion

First, there are 2 special methods, decision tree and random forest, which both provided variable importance graphs (figure 3 and figure 10, which measured the importance of each feature for the classifier. The graphs are quite different, suggesting that the 2 classifiers found very different ways of using the features to classify each entry. The decision tree classifier consistently found the "thal" feature to be extremely important, while in the random forest classifier, it was only moderately valuable, and certainly not that much more than any other variable. This suggests that the decision tree classifier might be over-relying on the "thal" variable. Figure 10 shows that random forest has several features which it relies on, illustrating a much healthier spread of importance. These results suggest that regardless of the situation, it will always be a massive improvement to use a random forest classifier, rather than a decision tree classifier.

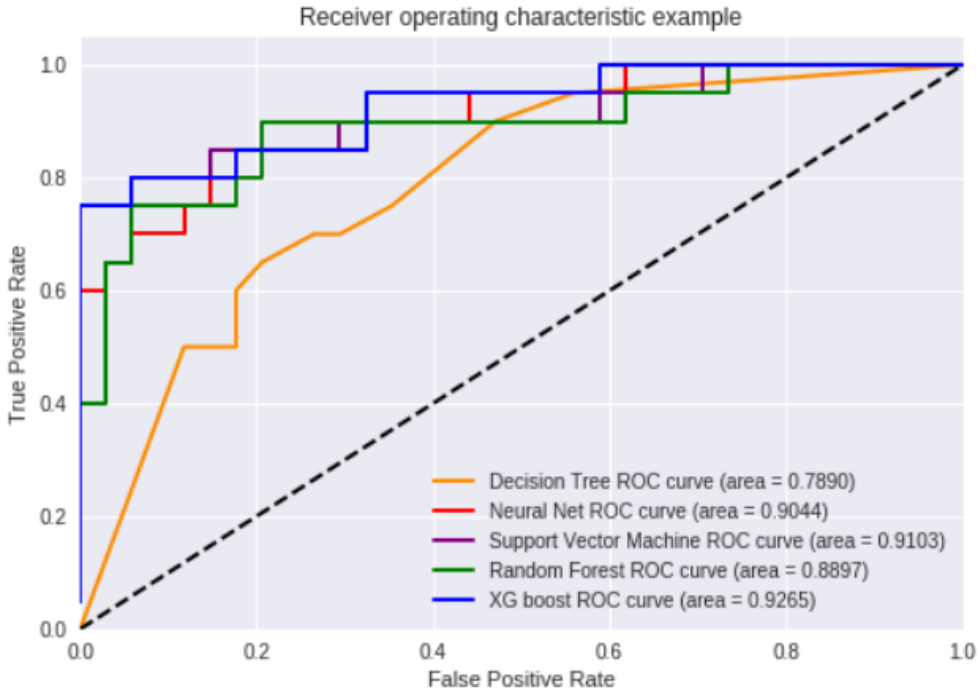

Figure 13: Comparison of the ROC curves of all the algorithms used in this study

The ROC curves confirm that the decision tree is the worst classifier to use. It has the least amount of ear under the curve, significantly less than other methods. The other algorithms have a fairly similar amount of area under their ROC curves. XGboost performs the best, with .93 area coverage, suggesting that it might be the best classifier for determining heart problems. Support Vector machine is second best, with .91 area coverage, suggesting that this method might also be fairly accurate. The neural net algorithm also performs similarly well, having the third largest area coverage with a value barely above 0.9. Finally, random forest does perform very well, with an area under its ROC curve being 0.89. However, this is still the fourth largest area, coming in after stand-alone algorithms like svm and neural nets. This contradicts the related work we mentioned earlier, which has consistently shown Random forests being one of the best classification methods. This suggests that the Random forest might not be properly tuned.

Algorithm	Best Parameters	Avg Precision	Avg Recall	Avg F1	Accuracy Score	Training Accuracy
decision tree	max_depth = 5, min_samples_leaf = 5, min_samples_split = 2, min_weight_fraction_leaf = 0.0, random_state = 0	0.7362637362637363	0.7529411764705882	0.7349228611500701	0.7407407407407407	0.873015873015873
neural net	activation = tanh, alpha = 0.01, hidden_layer_sizes = (10, 5, 5, 5), learning_rate = constant, random_state = 0	0.8011363636363636	0.8117647058823529	0.8051948051948052	0.8148148148148148	0.8888888888888888
support vector machine	C = 0.6, degree = 1, gamma = 0.001, kernel = linear, max_iter = 1000, random_state = 1	0.8211781206171107	0.8367647058823529	0.8261180679785332	0.8333333333333334	0.8492063492063492
random forest	criterion = entropy, max_depth = 3, max_features = 0.01, n_estimators = 16, random_state = 0	0.7813852813852814	0.7867647058823529	0.7837641062977794	0.7962962962962963	0.8809523809523809
XGboost	booster = gblinear, learning_rate = 0.2, max_delta_step = 0.0001, n_estimators = 5	0.8411764705882353	0.8411764705882353	0.8411764705882354	0.8518518518518519	0.8412698412698413

Figure 14: Comparison of results for each algorithm along with its optimal parameters.

Ultimately, the best algorithm is the one with the highest precision and recall scores. These two scores working together describe how well an algorithm finds all the positive cases (of heart disease), and how unlikely false negatives are. An algorithm with a higher precision and recall will generally have more false positives and false negatives. Considering that a false negative would be to miss the signs for heart disease, it is far better for the algorithm to have false positives rather than false negatives.

Considering the results, the XGboost algorithm is the best machine learning algorithm for finding heart disease. However, its advantage in precision and recall over the second best machine learning algorithm (Support Vector Machine), is less than its advantage in accuracy. Since there is such a small difference, and the support vector machine is reliable (not overfitting), we consider that both the xgBoost and svm algorithms can be recommended to be used in order to identify heart disease. The neural net algorithm is the third best algorithm out of all the methods we tested. While it does overfit a little, unlike svm, suggesting its results may not be as reliable. The fourth best method is random Forest, which has a recall and precision lower than .8. While the algorithm does perform relatively well, when considering the other results, it is still worse than most other algorithms. These results, as discussed earlier, do contradict the findings in other papers, and should be investigated more in future research. Finally, the decision tree algorithm has the worst results. While these results are better than random guessing, it is still a very poor performance and not recommended for actual use in diagnosing heart disease.

5. Future Work

One thing that needs to be investigated in the future is a better model for random forest. Many other papers have used the random forest classifier to much better effect. It's very strange that the random forest classifier performed so poorly in our tests.

In addition, future testing should consider much more data. More data would ensure a more reliable outcome and a better trained algorithm. As it stands, there is some randomness involved when splitting the data (into test and validation) due to how little data exists.

6. Conclusion

In this paper we compared multiple machine learning algorithms against each other in order to find the best algorithm to diagnose heart disease. The best algorithms for diagnosis which we found, were Support vector machine and XGBoost algorithms. Both algorithms generally had an accuracy around 85%, which was slightly better than the competition. Both of these were also the most consistent, avoiding overfitting. Future algorithms for heart diagnosis can be built and improved with these results.

7. Contributions

Dinidu - Wrote the basic code; wrote code to display results; wrote report; helped investigate results and what they meant

Max - Fine-tuned the parameters for algorithms; wrote report; helped to investigate results and find what they meant

Bibliography:

[1] R. Casanova, H. Fang-Chi and K. Sink, "Alzheimer's Disease Risk Assessment Using Large-Scale Machine Learning Methods," *PLOS ONE,* vol. 8, no. 11, p. e77949, 2013.

[2] A. Martinez-Vernon, J. Covington and R. Arasaradnam, "An improved machine learning pipeline for urinary volatiles disease detection: Diagnosing diabetes," *PLOS,* vol. 13, no. 9, p. e0204425, 2018.

[3] S. Al'Aref, K. Anchouche, G. Singh, P. Slomka and K. Kolli, "Clinical applications of machine learning in cardiovascular disease and its relevance to cardiac imaging," *Applied Medical Informatics,* vol. 36, no. 1, pp. 23-32, 2015.

[4] G. Wang, F. Xu, J. Wang and B. Ma, "A novel machine-learning model for identification of significant coronary artery disease," *Journal of the American College of Cardiology,* vol. 70, no. 16, p. C113, 2017.

[5] M. Kukar, C. Grošelj, I. Kononenko and J. Fettich, "An application of Machine Learning in the diagnosis of ischaemic heart disease," *Conference on Artificial Intelligence in Medicine in Europe,* vol. 1211, pp. 461-464, 1997.

[6] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj and J. Fettich, "Analysing and improving the diagnosis of ischaemic heart disease with machine learning," *Artificial Intelligence in Medicine,* vol. 16, no. 1, pp. 25-50, 1999.

[7] T. Tantimongcolwat, T. Naenna, C. Isarankura-Na-Ayudhya and M. J. Embrechts, "Identification of ischemic heart disease via machine learning analysis on magnetocardiograms," *Computers in Biology and Medicine,* vol. 38, no. 7, pp. 817-825, 2008.

[8] B. Mortazavi, N. Downing, E. Bucholz and K. Dharmarajan, "Analysis of Machine Learning Techniques for Heart Failure Readmissions," *Circ Cardiovasc Qual Outcomes.,* vol. 9, no. 6, pp. 629-640, 2016.

[9] J. Frizell, L. Liang and P. Schulte, "Prediction of 30-Day All-Cause Readmissions in Patients Hospitalized for Heart Failure: Comparison of Machine Learning and Other Statistical Approaches," *AMA Cardiol.,* vol. 2, no. 2, pp. 204-209, 2017.

[10] V. Dominic, D. Gupta and S. Khare, "An Effective Performance Analysis of Machine Learning Techniques for Cardiovascular Disease," *Applied Medical Informatics,* vol. 36, no. 1, pp. 23-32, 2015.

[11] J. Zhang, R. L. Lafta, X. Tao, Y. Li, F. Chen, Y. Luo and X. Zhu, "Coupling a Fast Fourier Transformation With a Machine Learning Ensemble Model to Support Recommendations for Heart Disease Patients in a Telehealth Environment," *IEEE Access,* vol. 5, pp. 10674-10685, 2017.

[12] P. C. Austin, J. V. Tu, J. E. Ho, D. Levy and D. S. Lee, "Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining

classification of heart failure subtypes," *Journal of Clinical Epidemiology,* vol. 66, no. 4, pp. 398-407, 2013.

[13] A. Mustaqeem, S. M. Anwar, A. R. Khan and M. Majid, "A statistical analysis based recommender model for heart disease patients," *International Journal of Medical Informatics,* vol. 108, pp. 134-145, 2017.

[14] S. Pouriyeh, S. Vahid, G. Sannino, G. De Pietro, H. Arabnia and J. Gutierrez, "A comprehensive investigation and comparison of Machine Learning Techniques in the domain of heart disease," *Proceedings - IEEE Symposium on Computers and Communications,* pp. 204-207, 2017.

[15] "Heart Disorders and Diseases - Heart Disease; Researchers from Maulana Azad National Institute of Technology Detail Findings in Heart Disease (Performance evaluation of different machine learning techniques for prediction of heart disease)," *Journal of Robotics & Machine Learning,* p. 93, 2018.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in {P}ython," *Journal of Machine Learning Research,* vol. 12, pp. 2825--2830, 2011.

[17] "sklearn.metrics.precision_recall_fscore_support," scikit learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support. [Accessed 29 November 2018].

[18] R. B. Sundaram, "An End-to-End Guide to Understand the Math behind XGBoost," Analytics Vidhya, [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/. [Accessed 29 November 2018].

[19] U. Malik, "Random Forest Algorithm with Python and Scikit-Learn," Stack Abuse, [Online]. Available: https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/. [Accessed 29 November 2018].