**Heart Disease and Machine Learning**

Authors: Tim Wendlandt

**Introduction**

In the United States, heart disease is the cause of twenty percent of all deaths. This means one out of five deaths is due to heart disease, making it the leading cause of death (1). If doctors could figure out what factors are most associated with heart disease, it’s possible that they would be able to use these as warning signs for their patients and intervene before it is too late. This project is looking at eleven independent variables and using both supervised and unsupervised machine learning algorithms to identify these patterns.

The specific variables I will be using in my models are age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiogram results, maximum heart rate achieved, exercise induced angina, oldpeak =ST, and the slope of the peak exercise ST segment. Some of these variables are self-explanatory, but others are not as common. These will be covered in a later section of this paper.

The algorithms I will be using for my analysis are Random Forest, Support Vector Machine (SVM), and K-means. Random Forest and SVM are both supervised machine learning algorithms and their output will be a binary zero or one to indicate if heart disease is present. Their results will be compared against each other to see which performs better and help inform the overall analysis. K-means is an unsupervised learning model and will be used to help interpret the results of the other models. Its output will be groups of variables, or clusters, that the algorithm deems similar.

**Dataset**

The dataset I am using for this analysis is called “Heart Disease Dataset” and is from Kaggle.com (2). The author is Manu Siddhartha and is owned by mexwell. This dataset contains patient records and collects several variables that the author believes may be indicative of heart disease. Specifically, there are eleven independent variables and one dependent variable. Of these twelve variables, four are binary and the rest are numerical. There are one thousand one hundred and ninety patient records for evaluation.

The eleven independent variables are age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiogram results, maximum heart rate achieved, exercise induced angina, oldpeak =ST, and the slope of the peak exercise ST segment. Age is the patient’s age in years and is represented by a whole number. Sex is the patient’s sex and is represented by either a one or zero indicating male or female. Chest pain type can be one, two, three, or four which indicate the patient’s pain is a typical angina, atypical angina, non-anginal pain, or asymptomatic. Resting blood pressure is the patient’s resting blood pressure in mm HG. Serum cholesterol is the patient’s cholesterol level in mg/dl. Fasting blood sugar is represented by either a one or zero which indicates if the patient’s fasting blood sugar is greater than one hundred and twenty mg/dl. Resting electrocardiogram results can be one zero, one, or two which indicates a normal result, a ST-T wave abnormality, or a left ventricular hypertrophy in the patient. Maximum heart rate achieved is the patient’s max heart rate and will be a whole number between seventy-one and two hundred and two. Exercise induced angina is represented by either a one or zero which indicates if the patient feels chest pain during exercise or while stressed (3). Oldpeak =ST is a whole number that represents a specific electrocardiogram pattern that is used as a marker for different health conditions (4). The slope of the peak exercise ST segment can be one, two, or three which indicates an upsloping, flat, or down sloping value. This is also an electrocardiogram pattern that is used as a marker for different health conditions (3).

The one dependent variable is target. This is represented by a one or zero and indicates if the patient has heart disease. A one represents that heart disease is present.

Since all the variables are numeric, no dummy variables will be created. Additionally, there are no null values, so no records need to be dropped or modified before modeling. However, the data has a wide value of ranges and units. This means that the mean for its variables is as low as .21 and as high as 140 and the data will need to be scaled before applying the SVM and K-means algorithms. I will not be scaling this data before applying the Random Forest algorithm, since feature scaling is irrelevant for that model type.

For the supervised algorithms I will be splitting the data randomly into test and training groups. The training group will represent eighty percent of the total data set, or nine hundred and fifty-two records. The test group will represent twenty percent of the total data set, or two hundred and thirty-eight records. Once the datasets are split, the same test and training values will be used for each algorithm. The data will not be split for the unsupervised K-means algorithm. Additionally, the K-means algorithm will only use the set of independent variables since I am looking for patterns without the influence of knowing whether heart disease is present.

**Machine Learning Algorithms**

The first supervised machine learning algorithm that I am using is Random Forest. Random Forest works by taking a random sample of data and building a decision tree, or for this project more specifically, a classification tree. A classification tree is built by continually splitting data based on input values. The splitting criterion can be decided by several methods, but the most popular is information gain. The algorithm will then repeat this process for the specified number of trees. Afterwords, each data point will be run through all the trees and the category that is selected by the most trees will be the data points classification. Although a single classification tree may not be very accurate, in combination with the other classification trees the results can be significantly improved.

For this project, I am using grid search to help me optimize my Random Forest algorithm inputs. Specifically, I asked grid search to combine values from the following three hyperparameters: max\_depth, max\_features, and n\_estimators. Max\_depth limits how many levels each decision tree within the forest is allowed to have. The default for this hyperparameter is none, which means the trees have no level requirement. For this project I chose to test a max\_depth of five, ten, and none. Max\_features limits how many features the algorithm is allowed to try. The default for this hyperparameter is the square root of the total number of variables. For this project I chose to test a max\_features limit of three, five, seven, nine, and eleven. N\_estimators determines how many decision trees make up the forest. The default for this hyperparameter is ten. For this project I chose to test ten, fifty, one hundred, one hundred and fifty, and three hundred trees. There are several other hyperparameters that can be manipulated, but to reduce the algorithms runtime to a reasonable length I am only choosing to manipulate these three.

The second supervised machine learning algorithm that I am using is SVM. SVM classifies data by finding the optimal separating hyperplane, or the hyperplane that maximizes distance between support vectors. Once the hyperplane has been established, data points can be fed into the algorithm and classified based on where they fall in relation to the hyperplane. What is great about SVM is that it can be “tricked” to work on linear and non-linear data. For non-linear data we can map our data to a higher dimensional space which may make it possible to find a better hyperplane.

For this project, I am using grid search to help me optimize my SVM algorithm inputs. Specifically, I asked grid search to combine values from the following three hyperparameters: C, gamma, and kernel.

C is the margin of distance between the separating hyperplane and the support vectors (5). The default value for this hyperparameter is one (6). For this project I chose to test a C value of .1, one, ten, one hundred, and one thousand. Gamma tells our algorithm how much weight to assign records that are far away from the hyperplane. The higher the gamma, the more weight far away records will have. In SciKit learn you can choose between gamma equations. These equations are then used to determine the gamma. The default equation for this hyperparameter is auto (6). For this project I chose to test the gamma equations auto and scale. Kernel is how the algorithm maps data to higher dimensions. The default value for this hyperparameter is rbf (6). For this project I chose to test the kernel value linear and rbf.

The non supervised machine learning algorithm I chose to use is K-means. K-means groups similar data points together in clusters to discover patterns without a specific goal in mind. To do this, the algorithm takes the input cluster value and randomly assigns each cluster a centroid. Once centroids have been chosen, each data point is assigned to a cluster by proximity to the centroid. Then new centroids are calculated, and data points are reassigned. This is repeated until no reassignment occurs.

To choose the number of clusters to use in my algorithm I am going to graph the within cluster sum of square (wcss) values from twenty different K-means algorithms, each of which uses a different number of clusters ranging from one to twenty. I will then identify visually where the “elbow” of the graph is, or the distance between wcss values begins to significantly decrease. I will also graph the silhouette score for those twenty algorithms and visually identify the highest score within the “elbow” range of my wcss graph.

**Discussion And Conclusions**

For the Random Forest algorithm, grid search selected the following hyperparameters for optimal results. It chose a max\_depth of ten, max\_features of three, and three hundred classification trees. With those hyperparameters the model returned an f1-score of .95 and the following confusion matrix.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted Values** |  |
|  |  | Has Heart Disease | Doesn’t Have Heart Disease |
| **Actual Values** | Has Heart Disease | 113 | 6 |
|  | Doesn’t Have Heart Disease | 7 | 112 |

This means that when applied to the test data set, the algorithm correctly predicted one hundred and thirteen patients had heart disease and one hundred and twelve patients did not. It also incorrectly predicted seven patients had heart disease and six patients did not.

These results are promising, only misclassifying thirteen records or five percent of the test data. However, there are significant implications to misclassifying someone with heart disease and telling them they don’t. I would be happier with a lower f1 score and less false negatives.

For the SVM algorithm, grid search selected the following hyperparameter for optimal results. It chose a C value of one hundred, the scale gamma equation, and the rbf kernel. With those hyperparameters the model returned an f1-score of .89 and the following confusion matrix.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted Values** |  |
|  |  | Has Heart Disease | Doesn’t Have Heart Disease |
| **Actual Values** | Has Heart Disease | 105 | 14 |
|  | Doesn’t Have Heart Disease | 12 | 107 |

This means that when applied to the test data set, the algorithm correctly predicted one hundred and five patients had heart disease and one hundred and seven patients did not. It also incorrectly predicted twelve patients had heart disease and fourteen patients did not.

When we compare these results to the Random Forest algorithm results, we see a decrease in performance across the board. The SVM model has a lower f1-score and a higher rate of false positives and false negatives. Additionally, both models have a very similar rate of false negatives. Given the possible consequences of not identifying heart disease in a patient, a lower f1-score could be overlooked in favor of a lower percentage of false negatives. However, this is not the case, so I would consider the Random Forest model to be superior.

For the K-means algorithm, I used the elbow and silhouette method in combination as described earlier. These graphs can be seen in my jupyter notebook, but I determined that three was likely the appropriate number of clusters to apply. This resulted in the following cluster distribution.

|  |  |
| --- | --- |
| **Cluster** | **Count** |
| 0 | 201 |
| 1 | 570 |
| 2 | 419 |

The first thing that stands out is when I add the cluster identifier back into my original dataset, there is a huge difference between how many patients had heart disease in groups zero and two vs group one.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster** | **Count** | **Heart Disease Count** | **% with Heart Disease** |
| 0 | 201 | 178 | 89% |
| 1 | 570 | 110 | 19% |
| 2 | 419 | 341 | 81% |

This tells me that however the algorithm chose to separate patients into group zero and two may give us a hint as to which variables are more indicative of heart disease. Pursuing this further, I created three separate dataframes and filtered each to show the attributes for each cluster. I then ran the summary statistics for each attribute and compared them against each other.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cluster** | **Mean Age** | **Mean Sex** | **Mean Chest Pain** | **Mean Max Heart Rate** | **Mean ST Slope** |
| 0 | 57 | .95 | 3.7 | 121 | 1.88 |
| 1 | 50 | .67 | 2.7 | 155 | 1.22 |
| 2 | 57 | .81 | 3.7 | 128 | 2.05 |

This resulted in five variables that appeared to be significant in assigning patients a cluster value of zero or one. Some of these seem to have an obvious link to the presence of heart disease. For example, it makes a lot of sense that older patients are more likely to have heart disease. However, others are less obvious. For example, it appears that having a sex value of one has some correlation to heart disease. I have no idea why this would be the case without further research.

**Limitations and Future Work**

The results of this project and the above analysis have raised several questions which I would want to explore further. Specifically, I would like to further explore the relationship between each variable and the outcome. Perhaps a simpler logistic regression model would help with this, or maybe there is a way to parse out factor importance from the existing Random Forest and SVM algorithms. I also think it is necessary to learn more about what each of the variables means medically, so I can better understand correlation vs causation.

I also would like to explore the K-means algorithm further. I would like to experiment with different cluster values to help me better understand what variables are correlated and why. This would also include looking further at the differences between the original cluster groups zero and two. After some examination, it was easy to see why cluster one was different, but it was much less obvious why clusters zero and two were split from each other.

Finally, one of the dangers of the Random Forest algorithm is overfitting. I would like to simplify the model and compare the results to see how drastic a difference there is. Additionally, I would like to try some other supervised learning algorithms to see if they can come close to the Random Forest algorithms results or further reduce the number of false negatives.

**Reference**

1. CDC. “Heart Disease Facts.” *Centers for Disease Control and Prevention*, 15 May 2023, www.cdc.gov/heartdisease/facts.htm#:~:text=Heart%20disease%20is%20the%20leading.
2. Siddhartha, Manu. “🫀 Heart Disease Dataset.” *Www.kaggle.com*, www.kaggle.com/datasets/mexwell/heart-disease-dataset?rvi=1. Accessed 28 Apr. 2024.
3. “Angina: Symptoms, Diagnosis and Treatments.” *Harvard Health*, 5 Dec. 2014, www.health.harvard.edu/heart-health/angina-symptoms-diagnosis-and-treatments#:~:text=Angina%20(pronounced%20ANN%2Djuh%2D.
4. Rowden, Adam. “ST Depression on ECG: What It Means and Causes.” *Www.medicalnewstoday.com*, 29 July 2022, www.medicalnewstoday.com/articles/st-depression-on-ecg.
5. https://twitter.com/cassiasamp. “Understanding SVM Hyperparameters.” *Stack Abuse*, 21 Apr. 2023, stackabuse.com/understanding-svm-hyperparameters/.
6. “SVM.” *Amagash.github.io*, amagash.github.io/pages/exploration/supervised-learning/svm.html.