Prediction of Heart Disease Using Supervised Learning (June 2019)

Jason Huggy, Student, Lewis University

Abstract— This study served as a process to identify the indicators and to accurately predict the presence of heart disease in a person using several supervised machine learning methods. The primary methods utilized in this study were nearest neighbors and decision trees. Through exploratory data analysis, this study was able to identify key characteristics that could serve as markers of heart disease in a person. After using several machine learning techniques, the best model was found to predict the occurrence of heart disease at an average prediction accuracy of 84%.

I. INTRODUCTION

Even with all the advancements in medicine, heart disease is still a serious problem in the United States that affects families in great number. According to the Centers for Disease Control and Prevention (CDC) about 610,000 people die of heart disease in the United States every year and is the leading cause of death for both men and women [1]. Every year more research is done to find ways to prevent heart disease; and thankfully, more and more data is being collected to do so. This study utilizes patient data collected by the University of California, Irvine (UCI) [2]. The original data included 76 attributes to do with the heart; however, for this study only 14 are used. The goal of this study is to train and test a supervised machine learning model that can accurately predict the occurrence of heart disease based on the 14 attributes used.

A. Nearest Neighbors

The first primary machine learning method utilized is nearest neighbors, which can be used for classification or regression. For classification, which is used in this study, the model tries to pair new data points with already classified data points. When *k* is specified as one, the model will pair the new data point with the closest point to it. For instance, if a new data point is plotted closest to another already known point classified as heart disease being present, then the new point would be classified as the same.

The user may also change *k* to increase the number of points that must be considered around a new data point. For example, if *k* is equal to five then the model will consider the 5 closest points around the new data point. If four of the five points closest to the new data point are classified as heart disease being present, then the new data point will categorize with the group of points of the greatest number.

The method of choosing *k* is done in a way to reduce error as much as possible without over-fitting the data, and the method used in this study will be explained in the methodology section. More can be read about the nearest neighbors model on the website for scikit-learn [3].

B. Decision Trees

Decision trees are used for classification and regression problems with the goal of predicting the target variable by using simple decision rules. Decision trees are used for classification in this study. Decision trees look just like a tree, except usually depicted upside down. There is a main root that branches out into new branches of decision rules, and each branch continues to branch out until the model is trained to properly classify the data given. There can be issues with over-fitting, so in most cases it is essential to prune the tree. This is done best by limiting the depth of the tree. This may increase error of the training data but will help generalize the tree better, so it is more accurate on new data. More can be learned about the specifics of decision trees on scikit-learn’s website [4].

In this study, Random Forests are utilized to increase the efficiency of the decision tree method. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [5]. This essentially takes the same data and creates many different decision trees, and then utilizes the benefits from each one to predict the target variable with the most accuracy. The parameter *n_estimators* decides how many trees the model will use.

II. METHODOLOGY

This study utilizes the data from 303 patients, collected by the UCI. The data set used includes 14 attributes, which all relate to the health of the heart in each patient. Those attributes are:

- *Age* is the age of the patient in years. The age of patients used in this study range from 29 to 77 years of age.
- *Sex* is the gender of the patient. 1 for male and 0 for female. The data used includes almost twice as many male patients as females.
- *CP* describes what type of chest pain the patient is experiencing. Value 1: typical angina; Value 2: atypical angina; Value 3: non-anginal pain; Value 4: asymptomatic.
- *Trestbps* is the resting blood pressure in mm Hg on the patient's admission to the hospital.
- *Chol* is the measure of the patient's serum cholesterol in mg/dl.
- *FBS* signifies whether the patient's blood sugar was greater than 120 mg/dl upon arrival at the hospital. 1 for true and 0 for false.
- *Restecg* is the patients resting electrocardiographic results. Value 0: normal; Value 1: having ST-T wave abnormality; Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.
- *Thalach* is the patient's maximum heart rate achieved in bpm.
- *Exang* signifies whether the patient experienced exercise induced angina during testing. 1 for yes, 0 for no.
- *Oldpeak* is the ST depression induced by exercise relative to rest.
- *Slope* is the classification of the slope of the peak exercise ST segment. Value 0: upsloping; Value 1: flat; Value 2: down-sloping.
- *CA* shows the number of major vessels (0-3) colored by fluoroscopy.
- *Thal* shows 1 for normal, 2 for fixed defect, and 3 for reversable defect.
- *Target* represents the target variable and signifies whether the patient is determined to have heart disease or not. 1 means the patient has heart disease, 0 means they do not.

A. EDA

To begin this study, exploratory data analysis (EDA) was conducted to visualize the data and make initial inferences. The Python packages Matplotlib and Seaborn are used to create the visualizations. The Seaborn boxplot was used for numerical attributes and count-plot was used for categorical attributes. Each attribute was visualized against the target feature.

B. Training and Testing Sets

For this study, the data used is split into a training and testing set. The training set contained 80% of the data and the testing set included 20%.

C. Nearest Neighbors

Scikit-learn's neighbors package, using Python, was used to create the models for this study. The first test only trained the model for 1 neighbor. The second utilized 6 nearest neighbors, after showing the most consistent trade-off between bias and variance.

D. Decision Trees

Scikit-learn is used to first create a single decision tree for the heart data. No changes are made to the default values. To improve this process, Scikit- learn's Random Forest Classifier is used. For this study *n_estimators* is left at 100.

E. Evaluation

Each method is evaluated using a confusion matrix and a classification report. Cross validation is also used at the final step of each method. The cross validation method performs the test 20 times each.

III. RESULTS & DISCUSSION

A. EDA

While conducting EDA, a lot of information was found that could be useful for the purpose of deciding if a patient has heart disease. Each attribute was visualized along with the target feature to show how the attribute affects the presence of heart disease in a patient. While not all the attributes proved to be useful, some definitely help paint a picture behind heart disease. Unfortunately, while some of the numbers may point to a phenomenon, they make not be right in every case.

I. Thalch

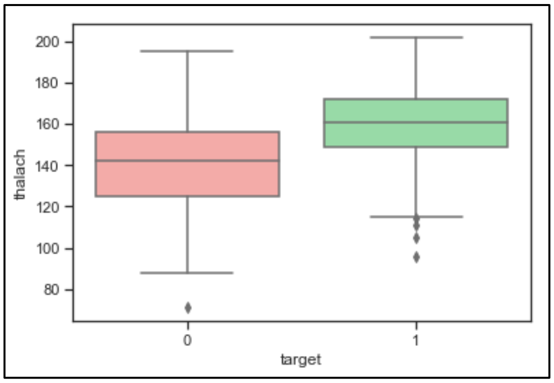

Fig. 1. Shows the maximum heart rate reached during testing. The green boxplot labeled “1” shows those with heart disease.

Thalch represents the maximum heart rate achieved while testing each patient. Fig. 1 shows the results with the green boxplot representing those with heart disease. This plot shows that a greater majority of patients with a heart rate of 150 or higher tend to be the ones with heart disease. There are several exceptions but this is a strong indicator to consider.

II. Exang

Exang signifies whether the patient experienced exercise induced angina during testing. 200 of the 303 patients tested did not experience exercise induced angina during testing. Out of those 200 patients, almost 70% were diagnosed with heart disease. For those who did experience exercise induced angina, only about 19% of those patients had a heart disease. This introduces a very important factor in the process of diagnosing heart disease. See Fig. 2.

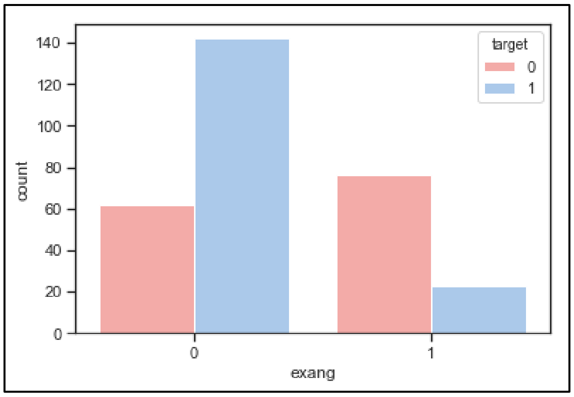

Fig. 2. Exang signifies whether the patient experienced exercise induced angina during testing. 1 for yes, 0 for no.

III. CA

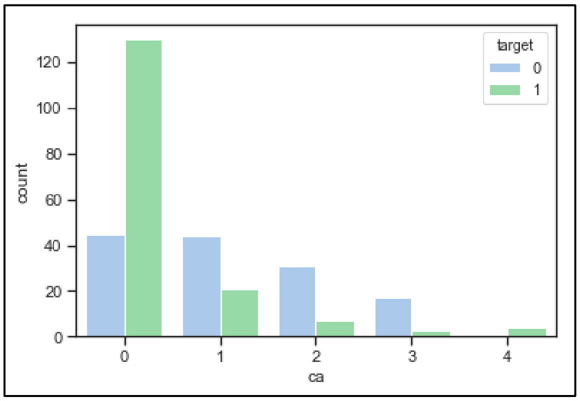

Fig. 3. Shows the number patients with/without heart disease based on the number of major vessels colored by fluoroscopy.

As shown in Fig. 3. there is a significant increase in the risk of heart disease when zero major arteries are visible through fluoroscopy. Almost 75% of the patients with zero major arteries visible were diagnosed with heart disease, showing a major correlation between the two instances.

IV. EDA Conclusion

Overall, it is easy to see that there are several indicators that show a strong correlation with the presence of heart disease. Not listed are *slope*, *thal*, and *oldpeak*, which all further help define what leads to heart disease; however, the separation in data is not as pronounced so it was left out. All in all, it is important to look at these features in order help in the classification of heart disease; although, not a single feature is definite. Through the use of machine learning, these features put help to improve upon the prediction of heart disease in a patient.

B. Nearest Neighbors

The first test using the nearest (k = 1) neighbor algorithm was relatively accurate overall in predicting heart disease. The test produced 19 true positives, 19 true negatives, 13 false positives, and 10 false negatives. This is a great start, but to better the model a greater number of neighbors is needed. For the second run, k is changed to 6. 6 is chosen because it returns the lowest error rate, with the most consistent results; thus, minimizing over-fit. When running k-nearest neighbors with k equal to 6, the test produced 21 true positives, 23 true negatives, 9 false positives, and 8 false negatives. Table 1 shows the evaluation metrics from both tests. A very clear improvement is seen in the scores when *k* is raised to 6.

	Precision	**Recall**	**F1-score**
k = 1	0.62	0.62	0.62
k = 6	0.72	0.72	0.72

Table 1. Shows the evaluation metrics for both nearest neighbor tests.

While these numbers are promising, cross-validation was run after this to test and found that the model is not as accurate as stated. While cross-validating the model with 20 iterations, the average accuracy was about 0.65. This isn't the worst prediction model, but it would show to be more inaccurate than preferred when faced with new data. Table 3 shows the results of the cross-validation test.

C. Decision Trees

Two different tests were run with decision trees. The first applied only one tree. The tree model performed well, returning 26 true positives, 22 true negatives, 10 false positives, and 3 false negatives. As with only a single nearest neighbor, this test performed well but could do better. The second run with decision trees utilized a random forest model, creating an ensemble of decision trees. This returned 24 true positives, 29 true negatives, 3 false positives, and 5 false negatives. Table 2 shows the evaluation metrics for both decision tree tests.

	Precision	**Recall**	**F1-score**
Decision Tree	0.8	0.79	0.79
Random Forest	0.87	0.87	0.87

Table 2. Shows the evaluation metrics for both decision tree tests.

After cross-validating with 20 iterations, the model was found to have an average accuracy of 0.84, which is a much better outcome than found with k-nearest neighbors. These numbers are much more reliable, though not perfect. Table 3 shows the cross validation results from the random forest test.

D. Machine Learning Outcome

Overall, machine learning proved to be very useful at producing an effective predictive machine; though, it would still need to be tested on further data for reliability standards. The nearest neighbor models were useful at generating a predictive model, but decision trees proved to be much more reliable at classifying the data.

	Mean	**STD**
K-Nearest Neigbors	0.65	0.11
Random Forest	0.84	0.08

Table 3. Shows the cross-validation results from k-nearest neighbors, with k = 6, and the random forest test.

IV. CONCLUSION

In closing, this study used the data taken from 303 patients to attempt to create a predictive model reliable enough to predict heart disease based on 14 attributes. Nearest neighbors and decision trees were employed to create this model. Overall, decision trees demonstrated to be the best tool for predicting heart disease in this study. The overall accuracy of the random forest model, which returned the best model, was 0.84. Further testing is needed, with more data, to confirm the reliability of this model. The usefulness of this model would also have to be reviewed by a medical professional.

V. References

[1] Centers for Disease Control and Prevention. *Heart Disease*. [Online] Available at: https://www.cdc.gov/heartdisease/facts.htm [Accessed 15 Jun. 2019].

[2] Kaggle.com. (2018). *Heart Disease UCI*. [Online] Available at: https://www.kaggle.com/ronitf/heart-disease-uci [Accessed 15 Jun. 2019].

[3] Scikit-learn. *Nearest Neighbors*. [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html [Accessed: 15 Jun. 2019].

[4] Scikit-learn. Decision Trees. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html [Accessed: 15 Jun. 2019].

[5] Scikit-learn. *Sklearn.ensemble.RandomForestClassifier*. [Online]. Available: https://scikit-

learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accessed: 15 Jun. 2019].