Universität Zürich UZH

# FINAL PAPER:
# PREDICTING HEART DISEASE

*Course:*
Supervised Machine Learning
Prof. Dr. Marco Steenbergen
Spring Semester 2020

*Written by:*
Yanik Lee Kipfer
[redacted]
Political Science (Major), Economics (Minor)

*Handed in on: 16.07.2020*

# Contents

# Introduction

Machine Learning techniques have been used to predict and classify various types of problems. However, one of the most promising fields of machine learning application can be found when dealing with medical diagnostics. Medical diagnostics are the first and most crucial step in the treatment and prevention of diseases, and the correct diagnosis of a disease can save lives. However, medical experts do not always have the time, tools or expertise to provide accurate diagnostics. Especially developing countries with weak healthcare systems are in desperate need of scalable diagnostic methods that decrease the need for health care resources. While machine learning techniques will not replace human health care workers, they can significantly facilitate their work and make them more efficient.

One of the leading causes of deaths in both developed and developing countries are cardiovascular diseases (CVDs). According to the World Health Organization (WHO), CVDs cost 17.9 Million lives per year, of which over three-fourths are lost in developing countries. Additional to the human cost, CVDs also carry an immense economic cost. In a study on the economic cost of CVDs, Gheorghe et al. (2018), concluded that the annual costs of CVD care exceed the health expenditure per capita in most developing countries, with a single episode costing anywhere between 500 to 5000 USD depending on the type of CVD. Thus, the active prevention of CVDs is not only a health problem but can also impact the economic development of a country and its people.

While CVDs can be prevented through behavioural changes, such as not smoking, eating a healthy diet and exercising, it is equally important to detect people who are already at risk, to prevent any further damage and costly treatments. Thus, to achieve early and reliable detection of CVDs, I propose using machine learning techniques to predict, whether a patient has heart disease.

# Approach

Modern medicine diagnoses heart disease based on the symptoms and biological markers a patient might show. Typical symptoms that might indicate the presence of heart disease are certain types of chest pain (angina), high blood pressure, the presence of high levels of cholesterol in the patient's blood, as well as arrhythmia (Lavanya & Supriya, 2019). Other factors that might increase the risk of heart disease are age and sex, with studies showing that older patients, as well as male patients, tend to show a higher risk for heart disease (Keyes, 2004). As such, the indicators of heart disease consist of numerical and categorical variables. If we want to predict the presence of heart disease within a patient, we can use these indicators to predict a binary outcome (Yes or No), which tells us whether a patient has heart disease or not. As such, we can use classification models for this type of problem. In the following, I will use two types of classification models, the C5.0 and Random Forest model, to classify patients according to their heart disease condition.

The C5.0 algorithm creates decision trees by splitting the tree according to the principle of node purity. A node is pure if all instances within the node belong to the same class. Thus, C5.0 partitions the data into subsets of similar classes. The C5.0 algorithm is optimal for the prediction of the heart disease problem, because it is capable of handling nominal, as well as numerical features, performs automatic feature selection and delivers easily interpretable results. Additionally, it can also handle asymmetric costs, which prove especially useful in diagnostics of diseases. However, the C5.0 algorithm is prone to problems of under- and overfitting and small changes in the data or feature set can provide massively different results.

Because of the weaknesses of the C5.0 algorithm, I also chose to include Random Forest as a second algorithm, to mitigate these shortcomings and produce more reliable results. The Random Forests is an ensemble algorithm which can be used both, for regression and classification tasks. Ensemble learning methods usually provide better predictive power than conventional machine learning algorithms by creating a multitude of sub-samples from the training set and training the model on each of the sub-samples, which reduces the problem of an overfitted model. The final prediction is gained by aggregating the results of all sub-samples. Thus, the random forest algorithm creates multiple decision trees during the training of the model, all of which include different features (random feature selection). An instance is than classified as belonging to the modal prediction of all the individual trees. As such, the Random Forest model does not suffer from the same shortcomings as the C5.0 algorithm. However, the Random Forest model is less interpretable than the C5.0 algorithm, since it relies on the aggregated result of a multitude of decision trees.

In the following sections, I will address the data collection and pre-processing steps, how I trained and evaluated the models and, lastly, conclude the findings of the study.

# Data Collection and Processing

The data was accessed through Kaggle and is based on the Heart Disease Data Set from the University of California. It consists of 297 observations on a total of 14 variables. Table 1 shows an overview of the dataset and provides a more detailed description of the variables included within the dataset.

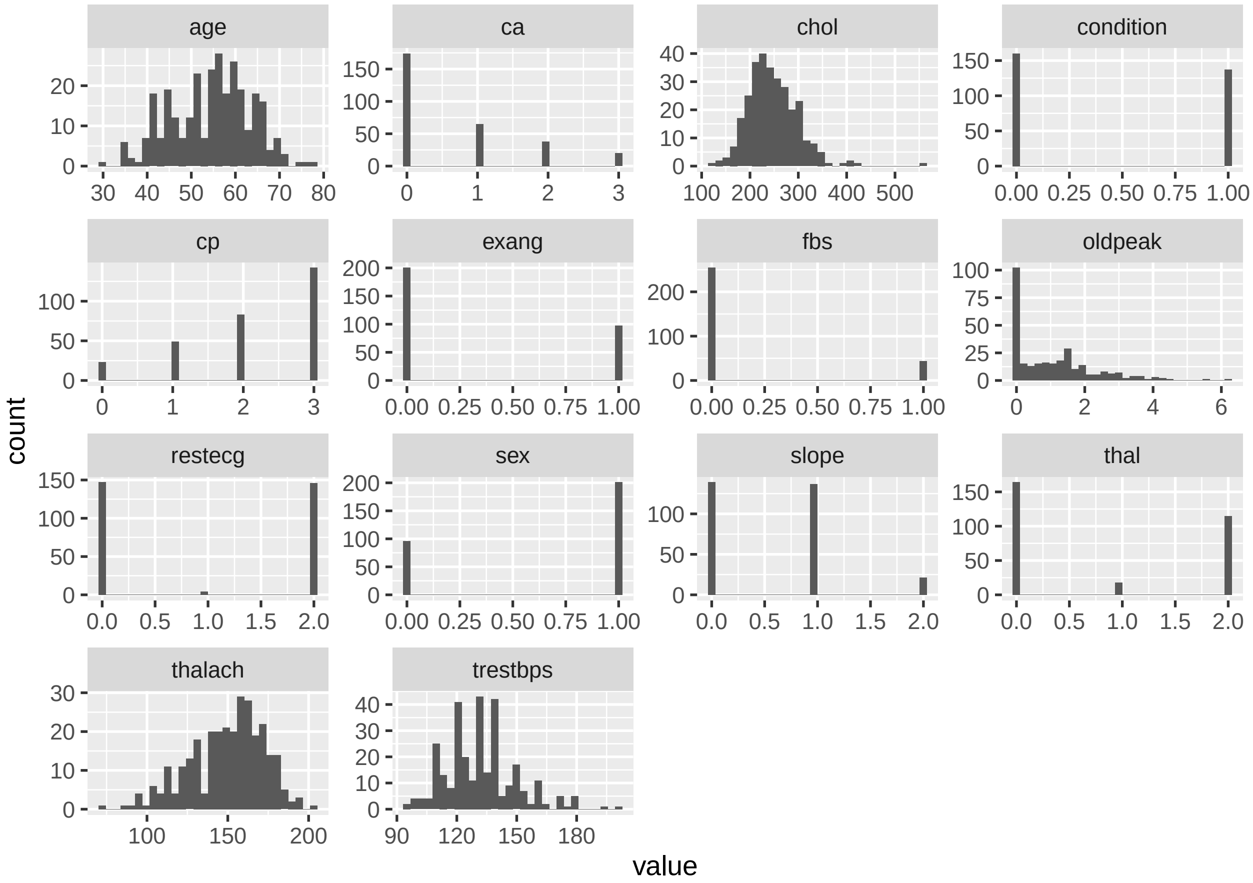
As can be seen, the variables consist of data on individual patients, ranging from demographic descriptions like the age and sex to biomarkers such as cholesterol and blood sugar levels, as well as electrocardiography diagnostics. As previously discussed, these variables might all serve as indicators for CVDs and should, therefore, be considered when predicting the presence of heart disease within a patient.

Table 1: Overview of Data and Variable Description

| age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | condition |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 69 | 1 | 0 | 160 | 234 | 1 | 2 | 131 | 0 | 0.1 | 1 | 1 | 0 | 0 |
| 69 | 0 | 0 | 140 | 239 | 0 | 0 | 151 | 0 | 1.8 | 0 | 2 | 0 | 0 |
| 66 | 0 | 0 | 150 | 226 | 0 | 0 | 114 | 0 | 2.6 | 2 | 0 | 0 | 0 |
| 65 | 1 | 0 | 138 | 282 | 1 | 2 | 174 | 0 | 1.4 | 1 | 1 | 0 | 1 |
| 64 | 1 | 0 | 110 | 211 | 0 | 2 | 144 | 1 | 1.8 | 1 | 0 | 0 | 0 |
| 64 | 1 | 0 | 170 | 227 | 0 | 2 | 155 | 0 | 0.6 | 1 | 0 | 2 | 0 |

| variable | description |
|---|---|
| age | age in years |
| sex | sex (1 = male; 0 = female) |
| cp | chest pain type; 0: typical angina, 1: atypical angina, 2: non-anginal pain, 3: asymptomatic |
| trestbps | resting blood pressure (in mm Hg on admission to the hospital |
| chol | serum cholestoral in mg/dl |
| fbs | fasting blood sugar > 120 mg/dl (1 = true; 0 = false) |
| restecg | resting electrocardiographic results; 0:normal, 1: ST-T wave abnormality, 2: left ventricular hypertrophy |
| thalach | maximum heart rate achieved |
| exang | exercise induced angina (1 = yes; 0 = no) |
| oldpeak | ST depression induced by exercise relative to rest |
| slope | the slope of the peak exercise ST segment; 0: upsloping, 1: flat, 2: downsloping |
| ca | number of major vessels (0-3) colored by flourosopy |
| thal | Thalassemia; 0 = normal; 1 = fixed defect; 2 = reversable defect |
| condition | Outcome Variable: 0 = no disease, 1 = disease |

The description of the variables lets us know, that the data consist of a mix of nominal and numerical features. However, when plotting the histograms of the variables, a notable observation is that the categorical variables are coded as numerical. Thus, in a first step I convert variables, such as sex or chest pain (cp), as well as the outcome variable (condition) into factors.



After having coded the categorical variables correctly, we take a second look at the distribution of the numerical variables. It is immediately noticeable that the numerical variables have widely differing scales, with variables such as *oldpeak* ranging from 0 to 6, while the variable *chol* ranges from 100 to 600. As such, before the creation of the models, I normalize all of the numerical variables, so that they only take on values between 0 and 1. With the pre-processing done, training the models can now be initiated.

```
'data.frame':   297 obs. of  14 variables:
 $ age      : num  0.833 0.833 0.771 0.75 0.729 ...
 $ sex      : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 2 2 1 2 ...
 $ cp       : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ trestbps : num  0.623 0.434 0.528 0.415 0.151 ...
 $ chol     : num  0.247 0.258 0.228 0.356 0.194 ...
 $ fbs      : Factor w/ 2 levels "0","1": 2 1 1 2 1 1 2 1 1 1 ...
 $ restecg  : Factor w/ 3 levels "0","1","2": 3 1 1 3 3 3 3 1 1 3 ...
 $ thalach  : num  0.458 0.611 0.328 0.786 0.557 ...
 $ exang    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
 $ oldpeak  : num  0.0161 0.2903 0.4194 0.2258 0.2903 ...
 $ slope    : Factor w/ 3 levels "0","1","2": 2 1 3 2 2 2 3 2 1 3 ...
 $ ca       : Factor w/ 4 levels "0","1","2","3": 2 3 1 2 1 1 1 3 1 1 ...
 $ thal     : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 3 2 1 1 3 ...
 $ condition: Factor w/ 2 levels "no.disease","disease": 1 1 1 2 1 1 1 2 1 1 ...
```

## Model Training

In the following, I will discuss how I split the data into training and test sets, as well as the parameters I chose to train the models.

To train the C5.0 and the Random Forest model, I first split the data into a test and a training set, which I do through the split sample approach. I set aside 80% of the data for the training set, and the remaining 20% are assigned to the test set. Based on this partition, I train the C5.0 and the Random Forest models with the training data. Additionally, according to the rule of choosing $\sqrt{P}$ parameters at each split for classification problems, I set the Random Forest model to randomly choose $\sqrt{13} \approx 4$ features at each split.

```r
# Set Seed
set.seed(9135, kind = "Mersenne-Twister", normal.kind = "Inversion")

# C5.0 Split Data
library(caret)
indx <- createDataPartition(data$condition, p = 0.8, list = FALSE)
dataTrain <- data[indx, 1:13]
dataTest <- data[-indx, 1:13]
labelTrain <- data[indx, 14, drop = TRUE]
labelTest <- data[-indx, 14, drop = TRUE]
```

```r
## Train model
library(C50)
mytree1 <- C5.0(dataTrain, labelTrain)

# Random Forest Split Data
library(randomForest)
Train <- data[indx, 1:14]
Test <- data[-indx, 1:14]

## Train Model
rf.fit <- randomForest(condition ~ ., data = Train, mtry = sqrt(13), importance = TRUE)
```

# Results

After having trained the models, I turn to the evaluation and, possible improvements that can be added to the models. I first discuss the C5.0 model and then turn to the Random Forest model evaluation.

## Model Evaluation - C5.0

By looking at the model metrics, it becomes clear that the C5.0 model has managed to predict 86% of the instances in the test set correctly, which is quite an achievement. Furthermore, Cohen's Kappa tells us we are doing 73% better, than if we just assigned cases to classes by chance, which once again points towards our model performing well.

Plotting the decision tree for our model shows that the C5.0 algorithm created a tree with 14 nodes and 19 leaves, which is a very complex and hard to interpret decision tree. As such we see that an instance will be classified as having heart disease if it has a *thal* value of 1 or 2 and *cp* of 3. However, a further nine paths can be identified that will lead to an instance being classified as having a heart disease.

Table 2: C5.0 Model Evalution Metrics

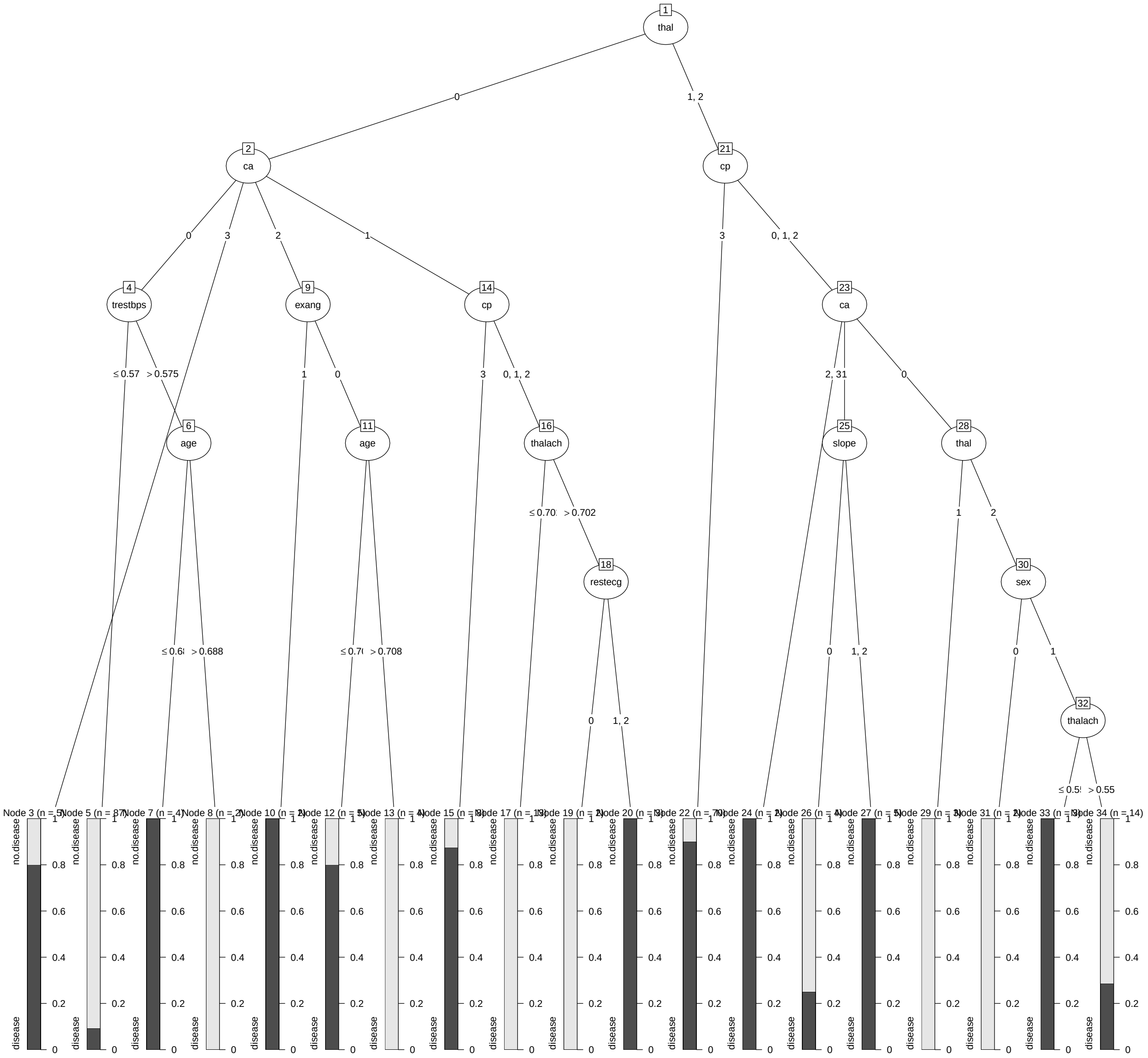
| | mycon1$overall |
|---|---|
| Accuracy | 0.86 |
| Kappa | 0.73 |
| AccuracyLower | 0.75 |
| AccuracyUpper | 0.94 |
| AccuracyNull | 0.54 |
| AccuracyPValue | 0.00 |
| McnemarPValue | 0.29 |

Table 3: C5.0 Variable Importance

| | Overall |
|---|---|
| thal | 100.00 |
| ca | 70.59 |
| cp | 54.20 |
| trestbps | 39.08 |
| thalach | 14.71 |
| sex | 7.98 |
| age | 6.30 |
| exang | 4.62 |
| slope | 3.78 |
| restecg | 2.10 |
| chol | 0.00 |
| fbs | 0.00 |
| oldpeak | 0.00 |

While looking at the decision tree, already gives us a notion of the most important features for the classification of heart disease, the metrics for variable importance provide a numerical value, which tells us about the percentage of cases that were classified based on a feature. Table 3 shows that the variable *thal* has been used to make 100% of the classifications. Moreover, the variables *ca*, *cp* and *trestbps* also contributed considerably to the classification of heart disease. The variables *chol*, *fbs* and *oldpeak*, however, do not seem to matter for the classification. In total, the model used 10 out of 13 features to some degree, to classify the instances.

Figure 1: C5.0 Decision Tree



To sum up, the C5.0 model is overly complex. Additionally, since we are trying to predict heart disease to facilitate fast and early treatment, the prevention of false negatives (where we predict no disease, but the disease is actually present) should carry more weight in our prediction. Thus, to address this problem, I incorporate asymmetric costs into the model. To train the C5.0 model with asymmetric costs, I first create a cost matrix, which will tell the model that false-negative should carry a three times higher cost, than false positives. Table 4 shows the cost matrix. Using the cost matrix, I re-train the model on the same training data.

Table 4: Cost Matrix

| | Actual | |
|---|---|---|
| | no.disease | disease |
| no.disease | 0 | 3 |
| disease | 1 | 0 |

Table 5: New C5.0 Model Evalution Metrics

| | mycon2$overall |
|---|---|
| Accuracy | 0.75 |
| Kappa | 0.51 |
| AccuracyLower | 0.62 |
| AccuracyUpper | 0.85 |
| AccuracyNull | 0.54 |
| AccuracyPValue | 0.00 |
| McnemarPValue | 0.00 |

The model evaluation metrics (table 5) show that the new model performs much worse than our previous model. The new model only attains a 75% accuracy and a Kappa of 51%. The decrease in accuracy can be attributed to the fact, that due to the asymmetric costs, the model prefers to classify someone without a disease as having a disease when it is not completely sure, to reduce false negatives. A look at the confusion matrix shows that the asymmetric model creates no false negative predictions (0 instances where the new model predicted no disease, but the patient was actually sick). However, compared to the naive model it has a higher false positive classification (15 instances where the new model predicts disease, where no disease is present).

Table 6: Confusion Matrix:Naive C5.0

| | Actual | |
|---|---|---|
| | no.disease | disease |
| no.disease | 26 | 2 |
| disease | 6 | 25 |

Table 7: Confusion Matrix:Asymmetric C5.0

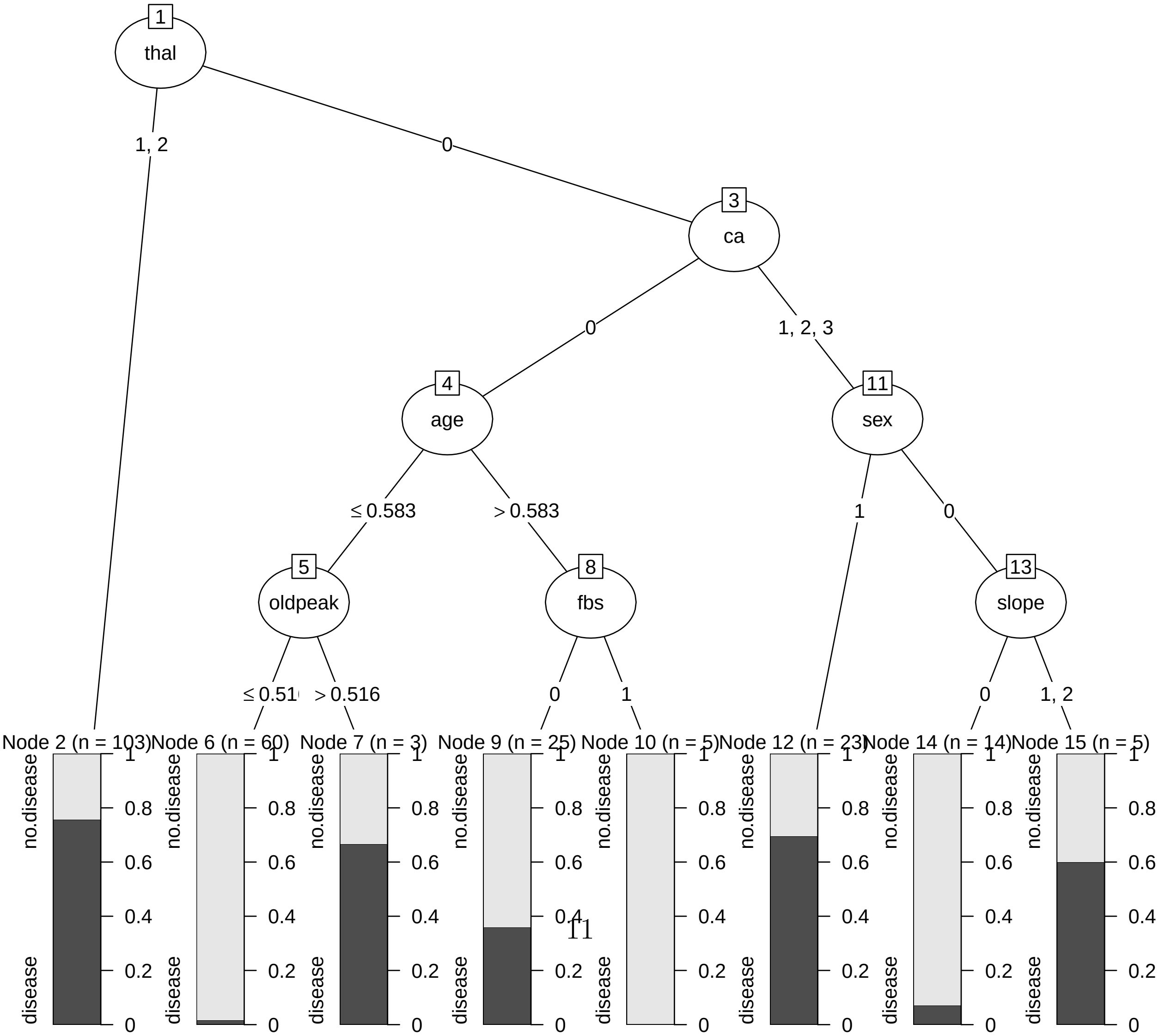
| | Actual | |
|---|---|---|
| | no.disease | disease |
| no.disease | 17 | 0 |
| disease | 15 | 27 |

Moreover, the implementation of asymmetric costs has also significantly decreased the complexity of the model. Our new decision tree (Figure 2) only has 7 nodes and 8 leaves, and 4 paths can be identified through which a patient would be classified as having an heart disease.

Table 8: New C5.0 Variable Importance Metrics

| | Overall |
|---|---|
| thal | 100.00 |
| ca | 56.72 |
| age | 39.08 |
| oldpeak | 27.31 |
| sex | 17.65 |
| fbs | 11.76 |
| slope | 7.98 |
| cp | 0.00 |
| trestbps | 0.00 |
| chol | 0.00 |
| restecg | 0.00 |
| thalach | 0.00 |
| exang | 0.00 |

Figure 2: New C5.0 Decision Tree



With the model being of lesser complexity it is unsurprising, that the variable importance metrics have also changed. The new model relies mostly on the variables *thal*, *ca*, *age* and *oldpeak* for the classifications, while the variables *cp*, *trestbps*, *chol*, *restecg*, *thalach* and *exang* are ignored. The new model, thus, relies on fewer variables to classify heart disease. All in all, the new C5.0 model only needed 7 out of 13 variables to classify the instances.

## Model Evaluation - Random Forest

I now turn to the evaluation of the Random Forest model. Calling the model output we see that the model generated 500 trees and used 4 features at each split. Moreover we have an Out-Of-Bag (OOB) error estimate of 19.75%. This means that 80.25% of OOB samples were correctly classified by our model.

```
Call:
 randomForest(formula = condition ~ ., data = Train, mtry = sqrt(13),      importance =
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 19.75%
Confusion matrix:
           no.disease disease class.error
no.disease        105      23   0.1796875
disease            24      86   0.2181818
```

Applying the Random Forest model on the test data gives us the metrics shown in table 9. It becomes immediately clear that the Random Forest model outperforms both C5.0 models. All in all we the model has an 90% accuracy. The Random Forest model counteracts the problem of overfitting prone in C5.0 models, through bagging and random feature selection. It creates a model that is much less prone to pick up on the noise present in the training

Table 9: Random Forest Performance Metrics

| | overall |
|---|---|
| Accuracy | 0.90 |
| Kappa | 0.80 |
| AccuracyLower | 0.79 |
| AccuracyUpper | 0.96 |
| AccuracyNull | 0.54 |
| AccuracyPValue | 0.00 |
| McnemarPValue | 0.68 |

data and through aggregation of the result from the individual tree, it is also able to create better predictions.

Looking at the confusion matrix, we can tell that the model predicts false negatives (2 observations where we predicted no disease, but the patient was actually sick). As already addressed in the C5.0 evaluation, false negatives are problematic for the diagnosis of heart disease. However, comparing the sensitivity values of all three models sheds more light on the capability of the models to predict true positives. As can be seen, the Random Forest and the naive C5.0 Model have the same sensitivity value, meaning they predict true positives equally well. The asymmetric C5.0 model, however, has a perfect sensitivity score. Thus, the asymmetric model managed to correctly classify all sick patients.

Table 10: Random Forest Confusion Matrix

| | Actual | |
|---|---|---|
| | no.disease | disease |
| no.disease | 28 | 2 |
| disease | 4 | 25 |

Table 11: Sensitivity By Model

| models | values |
|---|---|
| Naive | 0.93 |
| Asymmetric | 1 |
| RF | 0.93 |

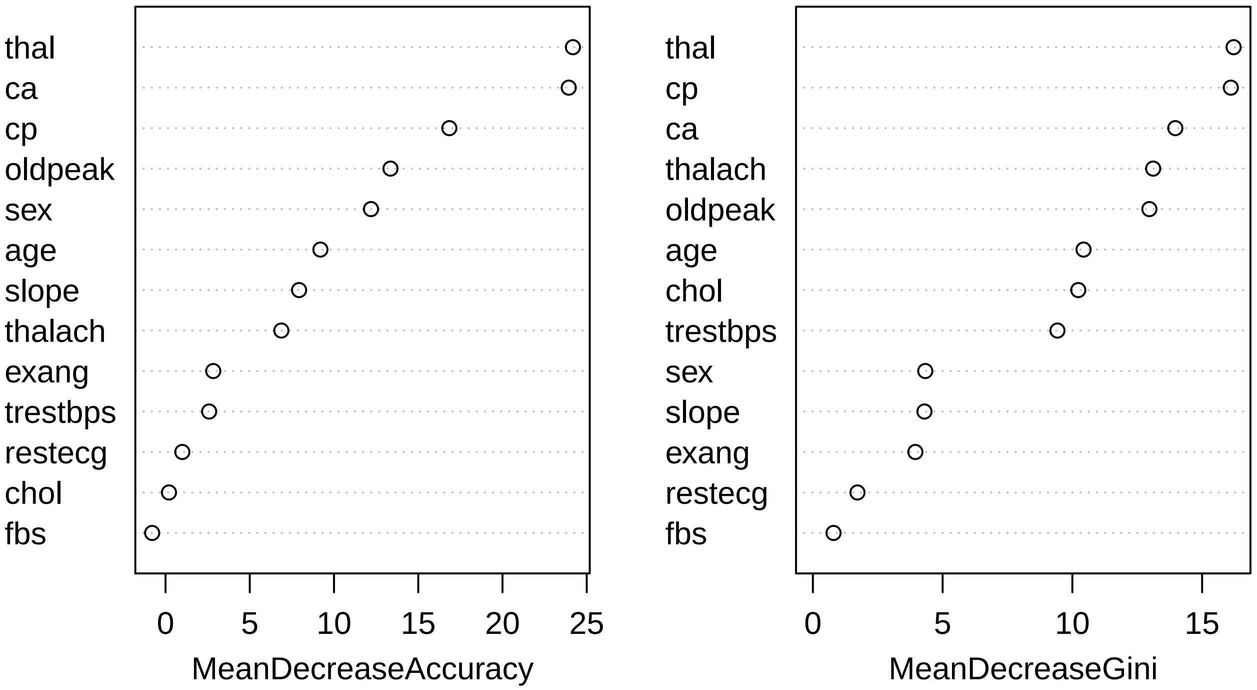
Lastly, turning once again to the topic of variable importance, Figure 3 depicts the importance of the features for the Random Forest model, according to the mean decrease in accuracy and mean decrease in Gini.

On the one hand, the mean decrease in accuracy tells us about the feature importance, by showing us the mean decrease in accuracy the model would suffer, if the feature were to be removed.The more the accuracy of the model decreases due to the exclusion the feature, the more important that variable is.

On the other hand, the mean decrease in Gini coefficient tells us how much a variable contributes to the purity/homogeneity of the nodes and leaves of the model. Higher values in the Gini coefficient indicate the presence of purer nodes and leaves within the model. The model experiences higher decrease in the Gini coefficient, when variables that create higher purity nodes are removed.

Looking at the Random Forest's variable importance metrics, it becomes apparent that according to the mean decrease in accuracy and the mean decrease in Gini, the most important variables for the correct classification of the model are *thal* and *ca*. While the most important variables for node purity are *thal* and *cp*. Removing these features would result in a 50% decrease in accuracy and a decrease of the Gini coefficient by 32%, thereby, significantly decreasing the performance of the model. Overall we can say that the feature importance of the Random Forest model is similar to those of the naive and asymmetric C5.0 model, in that *thal*, *ca* and *cp* tend to be the most important features.

Figure 3: Random Forest Variable Importance



## Discussion & Conclusion

This study set out to create a model capable of predicting heart disease in patients, using Data on 13 heart disease indicators of 297 patients. I chose to apply two types of classification algorithms to solve the classification problem, the C5.0 and the Random Forest algorithm. The choice of these two models was mainly driven by the desire to balance out the weakness in one model, with the strength of another model. Additionally, I created two types of C5.0 models. One with asymmetric costs and one without (the naive model).

The study found that the model that performed best in predicting heart disease was the Random Forest model with an accuracy of 90%. The naive C5.0 model provided the second-best classification performance 86%, while the asymmetric C5.0 model performed the worst with only a 75% accuracy. The most important feature for the prediction of heart disease across all three models seemed to be the presence of *thalassemia* (thal), *chest pain type* (cp) and the *number of major vessels coloured by* flourosopy* (ca). Moreover, the findings on feature importance do not only allow for the creation of less complex and more efficient models, it also permits healthcare workers to more narrowly focus on specific indicators, reducing the need for more extensive lab tests and, thereby, costly diagnostics.

The final question that needs to be discussed is, which model should be preferred? On the one hand, through boosting and random feature selection, the Random Forest model manages to mitigate the danger of overfitting usually present in overly complex C5.0 models and produce better predictions. On the other hand, the C5.0 models are more easily interpretable than the Random Forest, since they allow for the depiction of the decision rules used for the classification. The ease of interpretation is an important aspect, especially in the medical field, where wrong decisions might cost lives, it is important to create models that both patients and doctors can comprehend. Furthermore, when dealing with classification models used in medical diagnostics, we want to ensure that our model correctly identifies all patients in need of treatment. Since the early treatment of a disease can significantly improve the survival chance, the model should be overly careful when deciding whether or not to classify someone as not having the disease and should prefer assigning healthy patients as sick, instead of sick patients as healthy. As such, given the aforementioned requirements our model should fulfill, it becomes apparent that the asymmetric C5.0 model is the best fit. Through the use of asymmetric costs, we can constrain the model to predict less false negatives. While the asymmetric cost significantly deteriorates the overall accuracy of the model, it should be a tradeoff we are willing to make, in exchange for better identification of sick patients. Across all three models, the asymmetric C5.0 model had the highest sensitivity and was able to successfully identify all sick patients.

# References

Gheorghe, A., Griffiths, U., Murphy, A., Legido-Quigley, H., Lamptey, P., & Perel, P. (2018). The economic burden of cardiovascular disease and hypertension in low-and middle-income countries: a systematic review. BMC Public Health, 18(1), 975.

Lavanya, C., & Supriya, A. (2019). Hybrid Machine Learning Strategies for Heart Disease Prediction in Healthcare. Journal of the Gujarat Research Society, 21(16), 2553-2559.

Keyes, C. L. (2004). The nexus of cardiovascular disease and depression revisited: The complete mental health perspective and the moderating role of age and gender. Aging & Mental Health, 8(3), 266-274.