Predicting heart disease with machine learning algorithms

Luka Tavčer (lt7339@student.uni-lj.si)
Faculty of Computer and Information Science, Ljubljana, Slovenia

Abstract—Heart disease is a leading cause of death worldwide. To classify if the patient has heart disease, he must visit the hospital, where they will examine him and perform needed tests. In the article, we tried to create a prediction model that can be used before the person was examined in the hospital with an ECG and other tests and classify him as a possible heart disease patient or not. Different machine learning algorithms were used such as decision trees, random forest, and support-vector machines. We see that it can predict disease with relatively usable accuracy.

Index Terms—Heart diseases, machine learning, predictions.

1 Introduction

Heart disease is a dangerous condition that can even be terminal. It is the leading cause of death in the United States, according to the Centers for Disease Control and Prevention (CDC) [1]. There are many types, and they affect the heart in different ways, of which some are preventable. Because of different reasons, the heart is unable to push enough blood to other organs and body parts, and therefore failures occur. The symptoms of heart disease are weakness, shortness of breath, pain in the chest. Heart disease diagnosis and treatment are very complex, because of the shortage of doctors and diagnostic machines. An experienced doctor can identify them, but doctors can only examine a limited number of patients per day.

We can use machine learning techniques and algorithms to try and help with the detection of such diseases. With accurate and fast diagnosis we can save many lives, reduce health-care costs and relieve doctors and medical workers from repetitive work. In the article we used a single decision tree, random forests and SVM machine learning algorithm to detect if a person has heart disease (or is potential to have it) or not, based on the heart disease data set of people with and without heart diseases. We evaluated our prediction model with different classification accuracy metrics, such as the receiver optimistic curves (ROC), area under the curve (AUC), mean average error (MAE) and root mean square error (RMSE). We have also calculated and compared the sensitivity and specificity of the models. Along the way, we tried to reduce the number of needed attributes for the prediction. In the end, we tried using only attributes that an average person can check at home, without the need to perform any more complex examinations such as an ECG.

2 Related work

In the article **A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms**, they have analysed and created a hybrid machine learning decision support system that will assist the doctors to diagnosis of heart patients efficiently [2]. They have used Relief, mRMR and LASSO feature selection algorithms and combined and tested them with many classifiers, such as KNN, ANN, SVM and Naive Bayes to predict heart disease. In their work they have selected the six most important features using Relief algorithm and performed calculations only on them. We expanded their work and checked if the same

accuracy of predictions can be achieved with less attributes.

Authors of the article titled **Random forests** have explained what random forests algorithm is and how it works [3]. They have also shown why it is good for many recent problems that have many input variables, i.e. medical diagnosis and document retrieval by performing experiments while comparing and showing how random forest is better than using single trees. In our work, we used random forest and compared its accuracy to using only a single decision tree.

3 DATA

Our model has been trained on the *Cleveland heart disease data set 2016*, from the University of California data mining repository [4]. It consists of 303 patients, where every row represents a patient with 14 columns as its features.

- **age**: age in years,
- **sex**: gender (male/female),
- **cp**: chest pain type
  - 1) typical angina,
  - 2) atypical angina,
  - 3) non-anginal pain,
  - 4) asymptomatic.
- **trestbps**: resting blood pressure (in mm Hg on admission to the hospital),
- **chot**: serum cholestoral in mg/dl,
- **fbs**: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false),
- **restecg**: resting electrocardiographic results
  - 0) normal,
  - 1) having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of ¿ 0.05 mV),
  - 2) showing probable or definite left ventricular hypertrophy by Estes' criteria.
- **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
  - 1) upsloping,
  - 2) flat,
  - 3) downsloping.
- **ca**: number of major vessels (0-3) colored by flourosopy,
- **thal**: thallium scan
  - 3 = normal,
  - 6 = fixed defect,
  - 7 = reversable defect.
- **target**: the predicted attribute, diagnosis of heart disease (angiographic disease status).

3.1 Data preprocessing

Our data set has some attributes with categorical values, for example, sex, which can be male or female. We need to convert these values to numbers, such as 0 and 1 since machine learning models are based on mathematical equations that can only use numbers to compute equations. We use so-called dummy variables, where we create two or more additional attributes from one attribute with more values. For example, if we originally had one attribute **sex** with two values, male and female. We then create two new attributes, *sex.male* with value 1 if user is male and 0 if not. We do a similar thing for *sex.female*.

We went even further by removing the attribute *sex.female* as the original attribute **sex** is binary and the model does not need two columns giving the same information. We could remove either male or female, as we know that if the user is not male, it is female, and vice versa. One column is redundant, so we removed *sex.female*.

Many machine learning algorithms perform better when numerical attributes are scaled to a standard range. Scaling the data means it normalizes it within a particular range, such as between 0 and 1. This is done because our variables are measured at different scales and do not contribute equally to the model fitting. Min-Max scaler does just that and normalization is done by the following equation:

Equation 1 (Min-Max normalization):

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

All attributes were then scaled using Min-Max scaler, even though tree-based models are usually not dependent on scaling, but others such as SVM, usually are [5]. We can then easily use other algorithms on the same preprocessed data set and training can converge faster.

We split our data set into two separate sets. Training and test set. Our models are learning on the training set which represents 80% of the original data set, and predictions are validated on the test set, which is 20% of the original data set.

3.2 Attribute correlations

As a part of exploring the data set, we have calculated correlations between all attributes, using Pearson's correlation coefficient. In Figure 1 we notice that **cp** (chest pain) and **thalach** (maximum heart rate achieved) are relatively highly positively correlated with the target attribute. Meanwhile **exang** (exercise induced angina) and **oldpeak** (ST depression induced by exercise relative to rest) are relatively highly negatively correlated with the target attribute. These four attributes are also among the six most important attributes, that were selected by Relief algorithm in the article [2].

ST depression (**oldpeak**) is the only attribute of these four that needs to be measured with electrocardiography (ECG). We would like to know if we can predict heart disease from the data that the user can normally measure at home. An average human usually doesn't own an ECG device to perform electrocardiography, so we will also try to predict heart disease without it. We will train our models with different attributes. On all 14 attributes, on the four most correlated attributes, and lastly, on the three most correlated attributes (without the oldpeak).

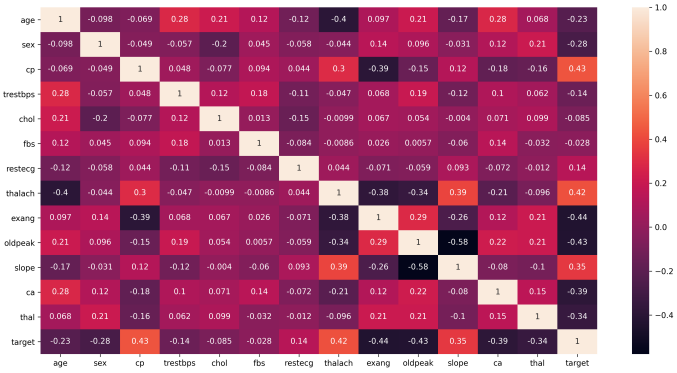

Fig. 1. Attribute correlations using Pearson correlation coefficient.

4 Algorithms Used

We used three algorithms to predict if a user has heart disease or not. We used decision trees, random forests, and support-vector machine (SVM) algorithms.

4.1 Decision trees

A decision tree is a supervised machine learning algorithm [6]. It is a tree-like model of decisions and their possible consequences. It analyzes observed values and creates a classifier that can predict a target class, where the target class is represented in its leaves. In our case, where our target class is discrete, can be 1 (has the disease) or 0 (no disease), we call it a classification tree. If the target variable is continuous, we call that tree a regression tree.

Decision trees are simple to understand and interpret. They don't require data normalization and perform well on large data sets. But they can be very non-robust, where a small change in the training data can change the tree structure and affect the target prediction greatly. Simple decision trees can easily become too complex and not generalize predictions well for the unseen data. This is called overfitting, as they perform greatly on the training data, but less on the new data. We can combat this with tree pruning, defining max tree length and defining max leaves. A visual example of the decision tree after our model was trained on all attributes is visible in Figure 2 we have visualized it with non-normalized values to make it easier for interpretation.

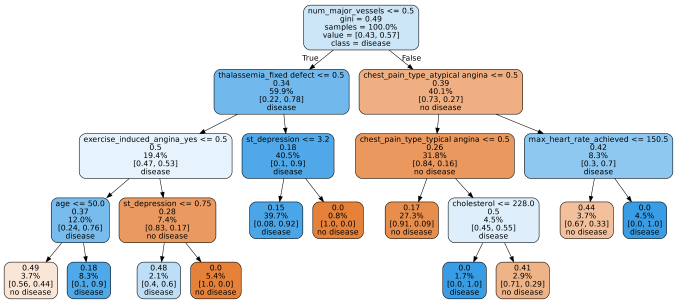

Fig. 2. Decision tree.

4.2 Random forest

Decisions trees often have a hard time classifying and are prone to overfitting. Because we have a relatively small data set, we have also used the random forest to classify if patients have heart disease or not. Random forest uses many trees and decides by calculating the average of all sub-tree predictions with the more formal definition defined in the Definition 1 [3]. It works like democracy, where every tree in the group votes for the class, and then the most popular or the class with most votes is taken. Usually, it has a much better prediction accuracy than a single decision tree.

Definition 1: A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

The algorithm can be good because it usually doesn't need a lot of effort to tweak the hyperparameters as it works well with default parameters. They always converge so that overfitting is not a problem. It also plays a major role in problems like medical diagnosis and document retrieval, as these problems usually have a lot of attributes and each with only a small amount of information. A single tree will have slightly better accuracy than if we randomly select the class. But if we combine more trees using random forests, we get much better accuracy. We can visualize one of the decision trees in the random forest in Figure 3.

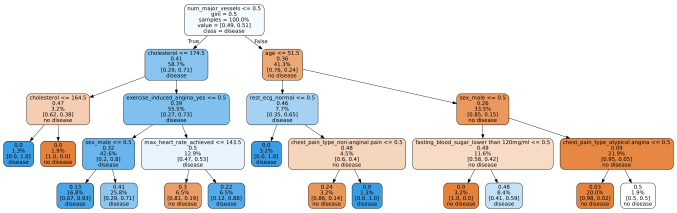

Fig. 3. One decision tree from the random forest.

4.3 SVM

Support vector machine (SVM) is relatively new and a promising method for learning separating functions in classification or pattern recognition tasks [7]. SVMs are supervised learning models that analyze data, based on statistical learning. SVM training algorithm builds a model that assigns new examples to one category or the other, making it non-probabilistic binary linear classifier. They have different kinds of applications in data classification, hand-written character recognition, and speech processing. We will use it as our final prediction model and compare its classification accuracy with the other two algorithms.

5 Results

After our models were trained on each algorithm, we tested them with different metrics. We calculated sensitivity and specificity, which are defined as:

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives}$$

They are both statistical measures of the performance of a binary classification test that are widely used in medicine [8]. Sensitivity measures the proportion of positive predictions that are correctly identified (i.e. of those who are correctly identified that they have heart disease) or how many sick people are correctly diagnosed with heart disease. Specificity measures the proportion of negatives that are correctly identified (i.e. of those who are correctly identified as healthy) or how many healthy people are correctly identified as not having heart disease.

In our case, sensitivity could have a higher weight as it is more important to identify those at risk for heart disease. Our model serves only to quickly and inexpensively identify if a user has heart disease. It works as a support system to medical workers that will later more thoroughly examine these patients that are supposedly at risk. Even if we falsely identify someone with heart disease, medical workers still need to confirm or disprove our decision, as they will have more data and can use other subjective clues after talking to the patient. In another case, if our model has a very low specificity it can overwhelm medical staff and doctors with additional unnecessary tests and examinations. However, we must explore the trade-off between sensitivity and specificity in the receiver operating characteristic curve (ROC) analysis, we can see them in figs. 4 to 6.

In figure 4 our algorithm was trained on all 14 attributes. We see that the random forest algorithm performs better than a single decision tree and SVM, results are visible in table 1. Random forest predicts with 84% accuracy and 88% sensitivity. Sensitivity is quite high, with the false negative rate (FNR) of 12% we see that it falsely diagnosed 4 people out of 33 with heart disease as healthy.

TABLE 1
Metrics performed on all 14 attributes.

	Decision Tree	Random Forest	SVM
Accuracy	0.75	**0.84**	0.8
Sensitivity	0.76	**0.88**	0.85
Specificity	0.74	**0.79**	0.75
AUC	0.73	**0.91**	0.87
MAE	0.25	**0.18**	0.2
RMSE	0.5	**0.4**	0.44

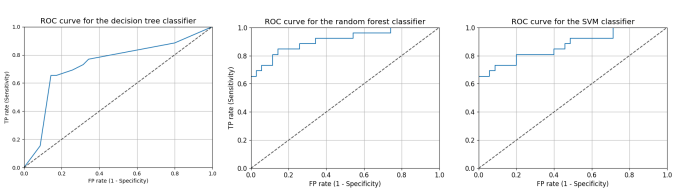

Fig. 4. ROC curves on all attributes.

Then we tried to reduce the number of attributes needed to identify heart disease. In figure 5 we trained our algorithm on the four most correlated attributes with the target value, along with age and sex, which were taken into training every time:

- age
- sex
- cp (chest pain)
- thalach (maximum heart rate achieved)
- exang (exercise induced angina)
- oldpeak (ST depression induced by exercise relative to rest)

Now we see that SVM is the most accurate algorithm of all three. Accuracy has dropped by 4%, and we see a drop in the sensitivity by 1% if we compare it against the best model that was trained on all attributes. Our model now failed to identify 4 out of 28 people with a health disease. The specificity of the best model has also dropped by 6%. The random forest model's accuracy has dropped by 10% but is still better than a single decision tree, which is again the worst of all three models, with an accuracy of 70%.

TABLE 2
Metrics performed on the four most correlated attributes.

	Decision Tree	Random Forest	SVM
Accuracy	0.7	0.75	**0.8**
Sensitivity	0.76	0.86	**0.87**
Specificity	0.64	0.67	**0.73**
AUC	0.78	0.87	**0.88**
MAE	0.3	0.25	**0.2**
RMSE	0.54	0.5	**0.44**

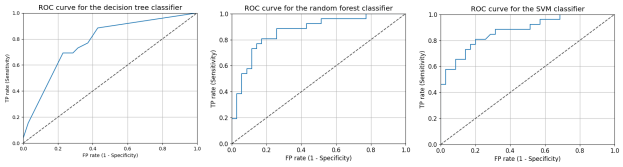

Fig. 5. ROC curves on the four most correlated attributes.

Among the previously used four most corre-

lated attributes is also the **oldpeak** attribute, which needs to be measured with the ECG. We supposed that an average person doesn't have an ECG device at home, we tried to train and test our model without the **oldpeak** attribute. The reason behind that was trying to get the most accurate prediction for heart disease by trying symptoms that ordinary people can check at home, before even going to the doctor. We trained our model with the following attributes:

- age
- sex
- cp (chest pain)
- thalach (maximum heart rate achieved)
- exang (exercise induced angina)

In table 3 we see that the random forest model is just slightly better than the SVM model. They have equal accuracy of 75%, which is 5% worse than the best model trained with the **oldpeak** attribute, and 9% worse than the best model trained on all attributes. Although they both have the same accuracy, the random forest has a better sensitivity rate by 2% than SVM. A single decision three is again the worst algorithm of all three.

TABLE 3
Metrics performed on the 3 most correlated attributes (without oldpeak).

	Decision Tree	Random Forest	SVM
Accuracy	0.64	**0.75**	**0.75**
Sensitivity	0.71	**0.83**	0.81
Specificity	0.57	0.68	**0.69**
AUC	0.75	**0.83**	0.81
MAE	0.36	**0.25**	**0.25**
RMSE	0.6	**0.5**	**0.5**

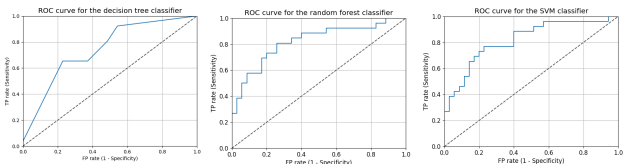

Fig. 6. ROC curves on the 3 most correlated attributes (without oldpeak).

6 Conclusion

We have trained and evaluated our model on three different machine learning algorithms. The random forest model was the best when we trained it with all attributes (84% accuracy) and when we trained it with attributes that an average user can measure at home, with an accuracy of 75%. When the model was trained on the four most correlated attributes with the target attribute, we achieved an accuracy of 80% with the SVM model. Decision trees performed the worst in all three cases.

Our main objective was to see if we can still achieve reasonable accuracy with attributes that we can measure at home, for example without the ECG. The best model (random forest) achieved 75% accuracy and 83% sensitivity, which can still be used to identify those with heart disease. It failed to identify 5 out of 30 users with heart disease. However, it achieved a specificity of 68%. It has falsely predicted that 10 of 31 healthy people have heart disease.

With some more easy attainable attributes we could maybe even increase the accuracy of the model. For now, we see that the model could be used to predict if someone has heart disease, but we should not rely only on the model alone.

References

[1] "Heart disease in the united states," https://www.cdc.gov/heartdisease/facts.htm, 2020, (Accessed on 27/12/2020).
[2] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Information Systems*, vol. 2018, 2018.
[3] L. Breiman, "Random forests," *UC Berkeley TR567*, 1999.
[4] "UCI machine learning repository - heart disease data set," https://archive.ics.uci.edu/ml/datasets/heart+disease, 2020, (Accessed on 27/12/2020).
[5] S. G. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015.
[6] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2007.
[7] A. Ukil, *Support Vector Machine*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 161–226. [Online]. Available: https://doi.org/10.1007/978-3-540-73170-2$_4$

[8] Wikipedia contributors, "Sensitivity and specificity — Wikipedia, the free encyclopedia," 2020, [Online; accessed 3-January-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sensitivity$_a nd_s peci f icity$