Classification for Heart Disease Diagnosis

Yafeng Wang

I. Introduction

Cardiovascular disease is a type of diseases affecting the cardiovascular system; it generally involves narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Cardiovascular disease is the number one cause of death across the world, accounting for more than 17.3 million death worldwide per year, and the death number is projected to reach 23.3 million by 2030. In the U.S alone, about 1 of every 3 deaths is due to cardiovascular disease, which is more than all forms of cancer and chronic lower respiratory disease combined. About 92 million American adults are living with some form of cardiovascular disease, and the direct and indirect costs of cardiovascular diseases are estimated to be more than $316 billion per year in the U.S. ([1])

Given the prevalence, the deadly nature of cardiovascular disease and the fact that it can attack at any time, it is important for the doctors to provide timely and efficient diagnosis for the patients. Currently, many patients complain about the tests conducted by hospitals for diagnosis, which are both costly and, in some cases, mistaken and result in the delay of starting proper treatments. One solution to this problem is to construct automated diagnostic systems based on the large amount of patient data. A good diagnostic system can serve as a second opinion and cross check on hospital diagnosis, potentially reducing the chance of mistaken diagnosis. Moreover, it can save both money and time, by speeding up the diagnostic process and by avoiding unnecessary diagnostic tests conducted on a patient. The purpose of this project is to compare various classification algorithms that can be used to construct such a diagnostic system, and to explore which predictors are most predictive of heart disease.

The rest of the paper is organized as follows: Existing literature on heart disease diagnosis is discussed in section II. In section III, the dataset as well as strategies for missing data handling are described. Section IV is a brief description of the main tasks of the projects and the methods used, and section V presents the experimental results and analysis. The study is concluded in section VI.

II. Literature Review

In recent years, a few studies have been done in comparing the classification accuracies of various algorithms as applied to the Cleveland heart disease data set, which is available on the UCI Machine Learning Repository. The Cleveland data set is a small data set with a total of 303 patients. The unprocessed version of the dataset contains over 76 predictors, whereas the processed version of the dataset contains only 13 predictors (which will be discussed in depth in the next section). Due to an error in the unprocessed version, machine learning researchers typically work on the processed version of the Cleveland dataset. The outcome variable of the Cleveland dataset is a categorical outcome variable with 5 levels (from 0 to 4). The data set is relatively complete, with only 6 patients having missing values.

A common theme of these studies is to treat the diagnostic problem in the context of the Cleveland dataset as a *binary* classification problem: They propose to collapse level 1-4 to 1 and interpret it as 'having heart disease'; level 1 is then contrasted with level 0, which is interpreted as 'healthy' (without heart disease). It is not entirely clear why such a choice is made. One possible interpretation is that in the original Cleveland dataset, the levels 1-4 do not represent different *types* of heart diseases (coronary heart disease, stroke, etc), but rather different *numbers* of major heart vessels that are either blocked or narrowed. It is possible that as long as there is a significant narrowing in at least one major blood vessel in the heart, a positive heart disease diagnosis is warranted.

In [2], Pouriyeh et al applied 10-fold cross validation to portion the Cleveland dataset into training and testing datasets, on the ground that this approach has a lower variance comparing with other estimating techniques such as single hold-out. They then used 7 different machine learning classifiers such as decision tree, Naive Bayes, Mutilayer perceptron, K-nearest neighbors, support vector machines, etc, and applied ensemble prediction of the classifiers (including bagging, boosting, etc) to the dataset. They concluded that the support vector machine method using the boosting technique has the best overall accuracy (84.15%).

In [3], Cherian et al evaluated the classification performance of the Naive Bayes classifier supplemented with the Laplace smoothing technique on the Cleveland dataset. For some reason, they only used 100 patients as the training dataset and another 50 patients as the test dataset, and they obtain a test set accuracy of 86%. They concluded that Naive Bayes algorithm with Laplace smoothing can serve as a decision support system to assist doctors for better clinical diagnosis.

Aside from the Cleveland dataset, there are three other heart disease datasets available on the UCI Machine learning Repository that share the same set of predictors and the same outcome variable: The Hungarian dataset (294 patients), the Switzerland dataset (103 patients), and the VA dataset (126 patients). Unlike the Cleveland dataset, however, nearly all the patients in these 3 datasets have some missing values with respect to the 13 predictors. [4] is a study that combines the

Cleveland data, the Hungarian data and the Switzerland data to obtain a larger dataset. The authors did not discuss their data preprocessing in detail, but it seems that they omitted any predictor that requires blood test and X-ray fluroscopy in order to reduce the number of missing values. After splitting the data using a stratified 4 to 1 train-test split, the authors applied four algorithms—random forests, logistic regression, support vector machines, and neural network—and used 5-fold validation for feature selection and parameter tuning. They concluded that maximum test accuracy is 78%, and that the test set error should be attributed to irreducible error in the problem.

III. Data Preprocessing

In terms of data collection, this project is a natural extension of the approach of study [4]. I decide to combine all the 4 heart diseases datasets available at the UCI website in order to obtain as large and representative a dataset as possible. The combination of the Cleveland data, the Hungarian data, the Switzerland data and the VA data give me a total of 920 patients, even though over two thirds of the patients have missing data.

Before discussing the issue of missing data, it is helpful to first examine the variables in the dataset in a little more detail. The combined dataset has 13 predictors and a categorical outcome variable, and most of the predictors are based on the results of various hospital tests for heart diseases such as exercise test, electrocardiogram, blood tests and X-ray fluoroscopy. Below is a list of predictor labels as well as their descriptions:

1) **Age**: Age in years. This is a numerical variable with values ranging from 28 to 77.
2) **Sex**: Gender. This is a binary variable where 0 represents female and 1 represents male.
3) **cp**: Chest pain type. This is a categorical variable with 4 values: Value 1 stands for ‘typical type’ (healthy), 2 stands for ‘typical type angina’, 3 stands for ’non-angina pain’ and 4 for ‘asymptomatic’.
4) **trestbps**: Resting blood sugar level in mm Hg based on blood test. This is a continuous variable ranging from 0[1] to 200.
5) **chol**: Serum cholesterol level in mg/dl. This is a numerical variable with values ranging from 0[2] to 603.
6) **fbs**: Fasting blood sugar in mg/dl. This is a *binary* variable: value 1 represents $>$ 120 mg/dl, and value 0 represents $<$ 120 mg/dl.
7) **Restecg**: Resting ECG result based on electrocardiogram. This is categorical variable with 3 values: value 0 means ‘normal’, value 1 means ‘ST-T wave abnormality’, and value 2 means ‘probable left ventricular hypertrophy’.
8) **thalach**: Maximum heart rate achieved. This is a numerical variable with values ranging from 60 to 202.
9) **exang**: Exercised induced angina. This is a binary variable where 0 represents ‘no’ and 1 represents ‘yes’.
10) **oldpeak**: ST depression induced by exercises relative to rest, based on the results of the electrocardiogram. This is a numerical variable ranging from -2.6 to 6.2.
11) **slope**: Slope of the peak exercise ST segment, based on the results of the electrocardiogram. This is a categorical variable with three values: 1 stands for ‘unsloping’, 2 stands for ‘flat’ and 3 stands for ‘downsloping’.
12) **ca**: Count of major vessels colored by floroscopy. This is a categorical variable with values 0-3.
13) **thal**: Thalassemia diagnosis. This is a categorical variable with 3 values: value 3 stands for ‘normal’, value 6 stands for ‘fixed defect’, and value 7 stands for ‘reversible defect’.

To reduce the number of missing values, I adopt the strategy of counting how many patients have missing values with respect to each of these 13 variables. The histogram of missing value counts is shown in Figure 1.

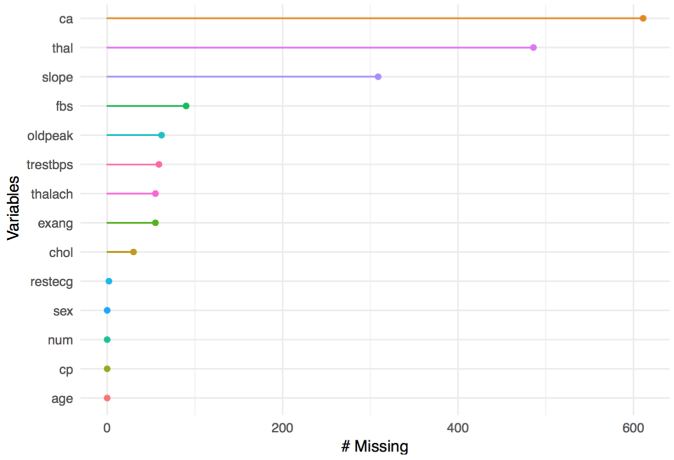

Fig. 1.

It is clear from the histogram that 3 variables account for the vast majority of missing values, namely slope, ca and thal. After removing these 3 predictors and then omitting all the patients that still have missing values, I manage to obtain a total of 740 patients without missing values. This is be the combined dataset that I apply the machine learning algorithms to.

IV. Task Definition and Approach

A. Task Definition

I conceptualize cardiovascular disease diagnosis as a problem of classification: For each patient, the input is a medical record that contains information relevant to predicting cardiovascular disease, and the output is a cardiovascular disease label. Given that I used the UCI heart disease datasets, I follow the existing literature by simplifying the task as *binary classification*: For each patient, the input is a feature vector consisting of the values of 10 predictors (excluding slope, ca and thal), and the output is either 0 (no cardiovascular disease) or 1 (having cardiovascular disease). For instance,

[1] It is unclear how to interpret a value of 0, given that missing value is coded differently as NA: The value of 0 might be an error.

[2] Again, the value 0 is probably an error (should have been coded as NA but wasn’t)

given an input vector $[67, 1, 4, 160, 286, 0, 2, 108, 1, 1.5]$, the classifier is supposed to return either 0 or 1.

After data preprocessing, I split the set of 740 patients into a training and a test dataset using a 7 to 3 training-to-test ratio, obtaining a training set of size 471 and a test set of size 269. The goal of the project is (1) to train different machine learning algorithms on the training set and compare their performance in classifying the test set; and (2) to select the most predictive predictors based on the performances of different classification algorithms as well as on some measure of predictor importance.

B. Metric of Evaluation

The main metric of evaluation used in this project is the overall classification accuracy on the test set. For a given classifier applied to a test set, let TP be the number of true positive[3] instances, FP be the number of false positive instances, TN be the number of true negative instances, and FN be the number of false negative instances. We can then define test set accuracy as follows:

$$\text{accuracy} = \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}$$

In addition to accuracy, I also compare the performances of the algorithms with regard to a few other metrics:

- Precision = $\frac{\text{TP}}{\text{TP + FP}}$: It is a measure that tells us what proportion of patients who were diagnosed as having heart disease actually have heart disease.
- Recall = $\frac{\text{TP}}{\text{TP + FN}}$: It is a measure that tells us what proportion of patients that actually have heart disease were diagnosed as having heart disease.
- Specificity = $\frac{\text{TN}}{\text{TN + FP}}$: It is a measure that tells us what proportion of patients that are actually healthy were diagnosed as healthy.

Different performance metrics should be emphasized depending on the goal of the classification task. If the goal is to minimize false negatives, then we would want recall to be as close to 1 as possible without sacrificing precision (or specificity) too much; similarly, if the goal is to minimize false positives, then we would want precision (or specificity) to be as close to 1 as possible without sacrificing recall too much. In the case of heart disease diagnosis, it is arguable that false negatives have worse consequences than false positives: A healthy patient falsely diagnosed as having heart problem may get a scare, whereas a patient with heart problem who was falsely diagnosed as healthy may risk worsening heart conditions or even sudden death. Therefore, I emphasize recall more than precision or specificity in the context of heart disease diagnosis.

C. Approach

Among the 740 patients in the combined dataset, 357 patients are healthy and 383 patients have heart disease. Within the test set, 128 patients are healthy and 141 patients have disease. If the classifier were to classify every patient as having heart disease, we would obtain a test set classification accuracy of 0.52. This number helps putting the classification accuracy of the algorithms to be considered into context.

Baseline approach: The baseline approach I use is logistic regression, which estimates the probability that a given patient belongs to a particular heart disease category (healthy or unhealthy). If we let Y be the binary outcome variables, and $X_1, \ldots, X_{10}$ be the 10 predictors, then logistic regression model gives us us that

$$\mathbb{P}(Y = 1 | X_1, \ldots X_{10}) = \frac{e^{\beta_0 + \beta_1 X_1 \cdots + \beta_{10} X_{10}}}{1 + e^{\beta_0 + \beta_1 X_1 \cdots + \beta_{10} X_{10}}}$$

Where best-fit coefficients $\beta_0, \beta_1, \ldots \beta_{10}$ can be estimated via maximum likelihood. Even though logistic regression is a relatively simple model, it can perform as well as any competing method in certain classification tasks. Moreover, it is relatively straightforward to apply L1/L2 regularization to the logistic model; the coefficients of the logistic model are relatively easy to interpret; and it generalizes easily to multinomial classification problems. All these properties make logistic regression a natural starting algorithm.

Alternative approach 1: The first alternative approach I use is naive Bayes classifier, which is another simple probabilistic classifier based on Bayes' theorem together with the strong assumption that all the predictors are independent conditional on the outcome. The model is:

$$\begin{aligned}\mathbb{P}(Y = 1 | X_1, \ldots X_{10}) &= \frac{\mathbb{P}(X_1, \ldots X_{10} | Y = 1)\mathbb{P}(Y = 1)}{\mathbb{P}(X_1, \ldots X_{10})} \\ &= \frac{\prod_{i=1}^{i=10} \mathbb{P}(X_i | Y = 1)\mathbb{P}(Y = 1)}{\mathbb{P}(X_1, \ldots X_{10})}\end{aligned}$$

Naive Bayes classifier is easy to implement and is particularly useful for datasets with a large dimensionality. In spite of its simplicity, it is shown to be capable of outperforming more sophisticated methods. The particular version of the Naive Bayes classifier in this project is supplemented with Laplace smoothing to avoid over-fitting.

Alternative approach 2: The second alternative approach I use is support vector machines (SVMs). SVMs are based on the idea of finding a hyperplane that best separate the data into two classes. In my dataset, since each observation has 10 features and I want to classify all the observations into 2 classes, it would be great if there is a hyperplane in the 10-dimensional space that can separate all the observations to the two sides of the plane. If the observations can indeed be separated by a hyperplane in this way, then what we want is a hyperplane that has the greatest *margin*—i.e., greatest distance to the nearest points on either side. The classifier based on such as hyperplane is called the *maximal margin classifier*.

On the other hand, if the observations are not linearly separable, then what we want is a 'soft margin classifier' that correctly classifies most of the training observations while allowing some observations to be on the wrong side of the margin. Such a classifier is called the *support vector classifier*. When training a support vector classifier, we specify a tuning parameter C which can be thought of as

[3] That is, patients who have heart disease and are correctly classified as having heart disease.

the total 'budget' that can allow each observation to be on the wrong side of the margin for some distance.

Support vector machines is the generalization of support vector classifiers to handle nonlinear boundaries. The specification of the support vector machine requires a type of functions known as kernels of observations. In particular, I use a type of kernel known as *Gaussian radial kernel*, which is defined as follows for observed feature vectors x_i and $x_{i'}$ in the heart disease dataset:

$$K(x_i, x_{i'}) = \exp(-\frac{\sum_{j=1}^{10}(x_i^j - x_{i'}^j)^2}{2\sigma^2})$$

In general, the hyper-parameters C and σ are chosen by the approach of *cross validation*. To reduce training time, I keep σ at a constant value (0.53), and use 10-fold, 3-repeat cross validation (with accuracy as the performance metric) to select the value of C out of three candidates: 0.25, 0.5 and 1. In the end, $C = 1$ is selected.

Alternative approach 3: The third alternative approach I use is random forests. Random forest is built on the idea of an ensemble of decision trees. For each individual decision tree, we divide the feature space into a set of disjoint regions by recursive binary splitting: each feature together with a cutoff point provide a split, and we look for the feature choice and cutoff point that minimizes the classification error. Then for every test observation that falls into a given region, we classify them as the most commonly occurring class of training observations in that region.

Random forests adds two more ideas to how we build decision trees: First, when creating an individual decision tree, each time during the recursive splitting, instead of considering all available features, we only consider for a random sample of m predictors as split candidates. Using 10-fold, 3-repeat cross validation, I discover that $m = 2$ split candidates works best for the combined heart disease data set. Second, an ensemble of decision trees is built on bootstrapped training samples. For a given test observation, we record the class predicted by each of the trees, and then take a majority vote: the overall prediction is the most commonly occurring majority class among the all the predictions.

Alternative approach 4: The final alternative approach I use is single-layer neural network. Due to the limited size of the dataset and the limited number of predictors, I decide not to train multiple-layer neural network. Instead, I train a simple 1-layer neural network to check whether its performance is comparable with that of the other methods. In addition, I use 10-fold, 3 repeat cross validation to tune the size (number of nodes) of the hidden layer, and obtain size = 3 as the optimal size for the hidden layer.

V. Experimental Results and Discussions

A. Full Model Results

When all of 10 predictors are used in the training of the models, the classification performances of the five algorithms are displayed in Table 1.

Metric Model	Accuracy	Recall	Precision	Specificity
Logistic Regression	0.7993	0.7569	0.8516	0.8480
Naive Bayes	0.8132	**0.8000**	0.8210	0.8263
SVM	0.8141	0.7708	0.8672	0.8640
Random Forests	**0.8178**	0.7842	0.8516	0.8538
1-layer NN	0.8067	0.7533	**0.8828**	**0.8739**

TABLE I

A few conclusions can be drawn from the experimental results: First, the test set accuracy of the five classifiers are very close to each other (all around 80%), with random forests performing slightly better than the rest. Since the five approaches involve very different assumptions, it is reasonable to infer that at least some portion of the 20% test set classification error may be irreducible, due to the idiosyncrasies and noise in the dataset itself. Second, Naive Bayes is the best in minimizing false negatives (has the best recall), whereas 1-layer Neural Network is the best in minimizing false positives (has the best precision and specificity). Assuming that false negatives are more costly than false positives in heart disease diagnosis, we may have some reason to prefer Naive Bayes classifier if its recall performance is robust (generalizable to larger datasets).

B. Feature Selection

The second goal of the project is to select a small number of predictors that are the most predictive of heart disease according to the classifiers. The idea is to take a few different measures of variable importance based on different classifiers; and if the different importance measures agree on the same subset of important variables, we will have stronger reasons to believe that the selection of the important predictors is robust rather than due to the idiosyncrasies of a particular classifier.

The simplest approach I use is to rely on the logistic model: By looking at the statistical significance (0.05 level) results of all the coefficients in the logistic model, we can see that **sex** (gender), **cp** (chest pain type), **thalach** (maximum heart rate achieved), **exang** (exercised induced angina), and **oldpeak** (ST depressed induced by exercise) are the only statistically significant predictors in the model. Based on this result, I re-train each of the five aforementioned approaches using only these five features in the model, and the classification performances are depicted in table II.

Metric Model	Accuracy	Recall	Precision	Specificity
Logistic Regression	**0.8067**	**0.7714**	0.8438	0.8450
Naive Bayes	0.7955	0.7552	0.8438	0.8413
SVM	0.7881	0.7320	**0.8750**	**0.8621**
Random Forests	0.7881	0.7483	0.8359	0.8333
1-layer NN	0.7658	0.7070	0.8672	0.8482

TABLE II

We can see that the predictive accuracy of logistic regression slightly improves using only these five predictors, whereas the accuracy of the other four algorithms slightly drop. Overall, the accuracy of the models using only the 5 selected predictors are reasonably close to the full models, which is consistent with the hypothesis that these 5 predictors are the most important heart disease predictors with respect to the combined UCI dataset.

Before moving on to other ways of selecting important predictors, we may pause here and ask the question: *why* are these predictors useful for predicting heart disease? For instance, why should gender be a good predictor of heart disease? The machine learning algorithms themselves do not provide an answer to this question, and my suggestion is that at least some of the predictive power of the variables come from the biases in the data. For instance, Figure 2 shows the health-to-disease ratio of the female group versus that of the male group in the combined UCI dataset:

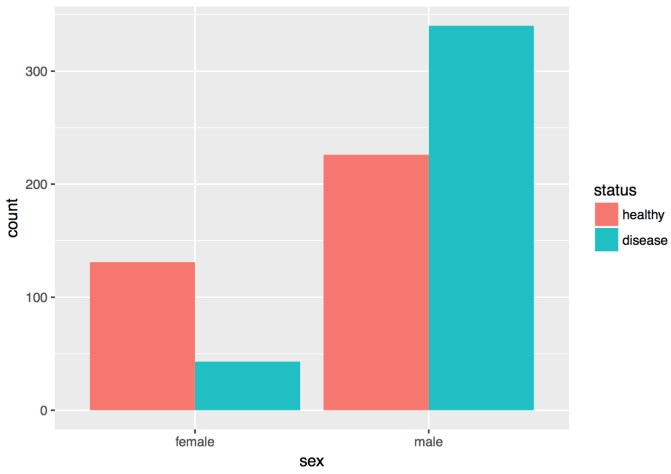

Fig. 2.

Figure 2 shows why sex is a good predictor of heart disease with respect to the combined dataset: In this dataset, men have a much higher risk of heart disease than women. However, note that this is an idiosyncratic characteristic of the UCI dataset that is not reflective of the general population: For instance, although heart disease is sometimes thought of as a 'man's disease', around the same number of women and men die each year of heart disease in the United States.([1]) Hence as far as this particular variable is concerned, its predictive power with regard to heart disease almost comes entirely from the bias within the dataset.

To check the robustness of the above feature selection, I also tried out a few different metrics of variable importance, based on different models: For instance, random forests provide a variable importance measure called 'mean decrease in accuracy', which is defined as follows: for each tree, the prediction accuracy on the out of the bag portion of the data is recorded. Then for every variable, we permute the values of this variable and record the prediction accuracy; the difference between these two accuracies then is averaged over all trees and normalized by the standard error. The R package *caret* contains an implementation of the random-forest based backward elimination algorithm that makes use of this measure of variable importance, and running this algorithm implementation shows that the same five predictors are ranked as the most important predictors by the random forests model.

Other approaches to variable importance obtain somewhat different results. For instance, for the SVM classifier, we can conduct a ROC curve analysis on each predictor, and use the ROC value for each variable as its importance measure. The top ranked variables according to this approach are **cp**, **oldpeak**, **thalach**, **exang**, and **age**, whereas sex drops to the sixth spot. If we conduct the ROC curve analysis on the predictors with respect to the neural network model, the top ranked variables become **cp**, **exang**, **oldpeak**, **restecg** and **sex**, whereas thalach has a minimal importance rating in this model. By comparing these variable importance results, we can conclude that there is a significant overlap in the set of important variables selected, and therefore the variable selection is largely robust, at least with respect to the particular dataset that I use.

C. *Best Result*

During a random experimentation, I discovered that if I used all the predictor except for trestbps (resting blood pressure) in training the Naive Bayes model, I could achieve a best test set accuracy of 0.841 and a best recall rate of 0.8195. Call this a 'all but trestbps' approach to feature selection. Applying 'all but trestbps' to the other classifiers produced different results: It improves the accuracy and recall of logistic regression and SVM (although not as much as Naive Bayes) compare to the full model approach (using all the predictors), and it decreases the accuracy and recall of random forests and 1-layer neural network. I have no understanding of why this particular feature selection works so well for the Naive Bayes classifier, but it does provide the best accuracy and recall I can get. Table III displays the results of applying the 'all but trestbps' feature selection to all the five classifiers:

Metric Model	Accuracy	Recall	Precision	Specificity
Logistic Regression	0.8104	0.7655	0.8672	0.8629
Naive Bayes	**0.8401**	**0.8195**	0.8516	0.8603
SVM	0.8216	0.7817	0.8672	0.8661
Random Forests	0.8104	0.7770	0.8438	0.8462
1-layer NN	0.803	0.7517	**0.8750**	**0.8667**

TABLE III

VI. Conclusions

In this project, five different machine learning algorithms—logistic regression, naive Bayes, support vector machines, random forests and 1-layer neural networks—are applied to a combination of four heart disease datasets obtained from the UCI ML repository. The estimations of classification accuracy and the selections of important variables based on different approaches are largely consistent with each other, supporting the hypothesis that the classification accuracy and the feature selection obtained are (for the most part) independent of the idiosyncrasies of particular methods. The major worry with regard to the project concerns the limitation of the dataset: It is clearly biased, and it is possible that the limitation in size and in the choice of predictors makes it difficult to differentiate the performances of different algorithms. A future continuation of the project will require examining larger, higher dimensional heart disease datasets as well as examining a wider range of classification and feature selection techniques, in order to construct an automatic heart disease diagnostic system that is usable in practice.

VII. Appendix

The code for the project is implemented in R, with the help of standard R machine learning packages. The R-markdown file containing the code can be found by clicking here, and the result of running the R-markdown file can be found by clicking here.

References

[1] Center for Disease Control and Prevention Fact Sheets, https://www.cdc.gov/heartdisease/facts.htm

[2] S. Pouriyeh, S. Vahid, G. Sannino, G. D. Pietro, H. Arabnia, J. Gutierrez, “A Comprehensive Investigation and Comparison of Machine Learning Techniques in the Domain of Heart Disease”, IEEE Symposium on Computers and Communications (ISCC), July 2017.

[3] V. Cherian, M.S.Bindu, “Heart Disease Prediction Using Naive Bayes Algorithm and Laplace Smoothing Technique”, International Journal of Computer Science Trends and Technology (IJCST), 5(2): 68-73, 2017.

[4] B. Dun, E. Wang, S. Majumder, “Heart Disease Diagnosis on Medical Data Using Ensemble Learning”, 1(1): Article 1, 2016.

[5] R. El-Bialy, M.A. Salamay, O.H. Karam, M.E. Khalifa, “Feature Analysis of Coronary Artery Heart Disease Data Sets”, Procedia Computer Science 65: 459-468, 2015