Comparison of Five Machine Learning Techniques for the Prediction of Cardiovascular Disease

Huan Yang Chan
School of Computer Sciences
University Science of Malaysia
Penang, Malaysia
huan-yang.chan@student.usm.my

Loganathan Muniandy
School of Computer Sciences
University Science of Malaysia
Penang, Malaysia
loganathan.muniandy@student.usm.my

Veerayen Mohanadas
School of Computer Sciences
University Science of Malaysia
Penang, Malaysia
veerayen.mohanadas@student.usm.my

Abstract— World Health Organization study shows cardiovascular diseases (CVD) causes 17.9 million deaths which equivalent to 31% of global mortality. [1] This is indeed a global issue and needed effective detection to take preventive steps. Currently, periodic health screening is widely used to detect CVD's presence. However, it is mostly a time-consuming process and costly. Thus, a feasible, fast and accurate prediction of CVD presence is crucial to help managing patient's health efficiently. For this project, 70000 patients record with 11 features from Kaggle [2] used to predict the presence of CVD with supervised machine learning techniques using R. In this study, a detailed comparison among different machine learning algorithms was used to determine the best performing model for detecting cardiovascular disease between Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Naïve Bayes (NB), k-Nearest Neighbors (kNN) and Random Forest (RF). Overall, Random Forest has the highest accuracy and also has the highest average running time.

Keywords— Cardiovascular diseases (CVD), supervised machine learning, Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Naïve Bayes (NB), k-Nearest Neighbors (KNN), Random Forest (RF)

I. INTRODUCTION

Cardiovascular disease is classified as a disease that generally involves the heart or blood vessels. CVD includes coronary artery diseases (CAD) such as angina and myocardial infarction (commonly known as a heart attack). Other CVDs are stroke, heart failure, hypertensive heart disease, rheumatic heart disease, cardiomyopathy, abnormal heart rhythms, congenital heart disease, valvular heart disease, carditis, aortic aneurysms, peripheral artery disease, thromboembolic disease, and venous thrombosis. [3]

Coronary artery disease, stroke, and peripheral artery disease involve atherosclerosis and could be caused by high blood pressure, smoking, diabetes mellitus, lack of exercise, obesity, high blood cholesterol, poor diet, and excessive alcohol consumption. High blood pressure is the main contributor to CVD deaths, followed by tobacco, diabetes, lack of exercise and obesity.

II. BACKGROUND AND LITERATURE REVIEW

Studies suggest that out of more than 17 million people who died from CVDs in 2008, about 17.6% of these deaths occurred before the age of 60 and could have largely been prevented. The percentage of premature deaths from CVDs ranges from 4% in high-income countries to 42% in low-income countries, leading to growing inequalities in the occurrence and outcome of CVDs between countries and populations. Interestingly, these deaths from CVDs have been declining in high-income countries, but have increased in low- and middle-income countries (LMIC). CVDs are mostly preventable with early and efficient diagnostic systems. However, health care systems in many LMIC are let down by a model based on hospital care focused on the treatment of diseases, often centered around high-technology hospitals that provide extensive treatment for only a small minority of citizens. Hospitals consume huge amounts of resources, and health ministries may spend more budgets on treatment services which depend on hospitals. This not serving and addressing well all those with CVDs. Thus, high number of people with high cardiovascular risk remain undiagnosed, and even those diagnosed have insufficient access to treatment at the primary health-care level. [4]

In order to assist the medical professionals, medical diagnostic based on the computer has been developed for analyzing the large volumes of the patient data. These system efficiency mainly depends on the features which are used must be correlated with some disease state.[5] Data mining is seen as important method in extraction of implied, unidentified and hidden information about data. Data mining is done through techniques like classification and clustering.

Literature review shows some of studies done in medical field earlier like Soni et al (2011) data mining techniques which are used in the current medical research mainly in prediction of heart disease. [6] Austin et al (2013) developed the alternate classification schemes based on the machine learning literature and data-mining which includes bootstrap aggregation (bagging), random forests, boosting and support vector machines. [7] Rotation Forest (RF) is constructed by Ozcift and Arif (2011) for evaluating their classification performances using heart diseases.[8] The classification technology is applied by Yeh et al (2011) for constructing an optimum prediction model of cerebrovascular disease. This model will extract and improve the prediction and diagnosis of cerebrovascular disease.[9] These suggest data mining and machine learning in the medical field is not totally new.

III. METHODOLOGY

We inspired by CRISP-DM methodology which stands for Cross-Industry Standard Process for Data Mining. We defined the machine learning workflow in 7 stages.

a) Data Collection
b) Data Pre-Processing
c) Exploratory Data Analysis
d) Model Researching
e) Model Training
f) Model Evaluation
g) Prediction

A. Data Collection

Cardiovascular Disease dataset is acquired from Kaggle. All of the dataset values were collected at the moment of medical examination. The dataset consists of 70,000 records of patients' data with the target (cardio) describing the presence or absence of heart disease using 11 features as described in Table 1. The input features are of three types: objective (containing factual information), examination (containing the results of a mearound 17.dical examination) and subjective (containing information given by the patient).

The target variable in this dataset is 'cardio'. Of the 70,000 records, 35,021 records are that of patients with cardio 0 and 34,979 records is that of patients with cardio 1.

Table 1 Twelve attributes of Cardiovascular Disease dataset.

Attribute	Type	Description
age	numeric	age of the patient in days
gender	factor	1: women, 2: men
height	numeric	height of the patient in cm
weight	numeric	weight of the patient in kg
ap_hi	numeric	systolic blood pressure
ap_lo	numeric	diastolic blood pressure
cholesterol	factor	1: normal, 2: above normal, 3: well above normal
gluc	factor	1: normal, 2: above normal, 3: well above normal
smoke	factor	whether patient smokes or not
alco	factor	alcohol intake-binary feature
active	factor	physical activity-binary feature
cardio	factor	Presence or absence of cardiovascular disease

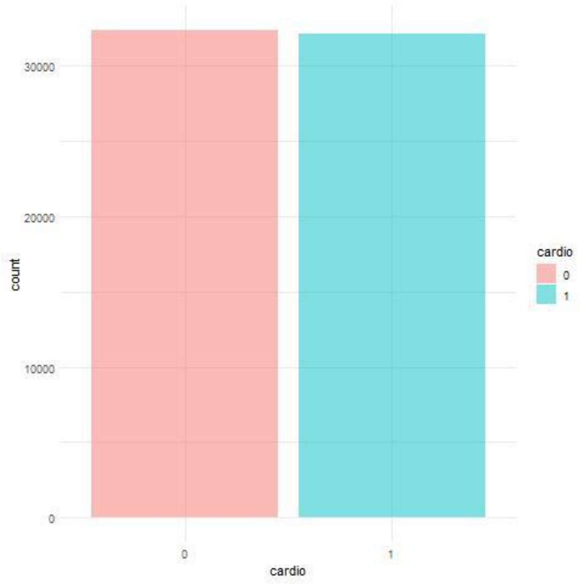


Fig.1 Presence or absence of cardiovascular disease.

.

B. Data Pre-Processing

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

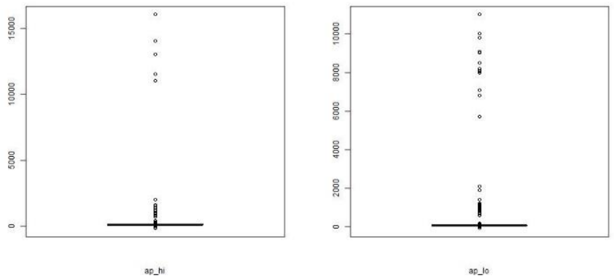
These are the pre-processing techniques that we used to clean the raw data.

i. Check out the missing values.
   - The dataset did not have any missing data.

ii. Identity numerical, ordinal and nominal categorical features.
   - Numerical Features: age, height, weight, ap_hi, ap_lo
   - Ordinal categorical: cholesterol, gluc
   - Nominal categorical: gender, smoke, alco

iii. Identify outliers.
   - By checking the boxplots of ap_hi and ap_lo, there are multiple readings with a value greater than 250[10] which makes no sense. This could be due to measurement error or recording error. The outlier values were removed using boxplot method.

Before remove outliers



After remove outliers

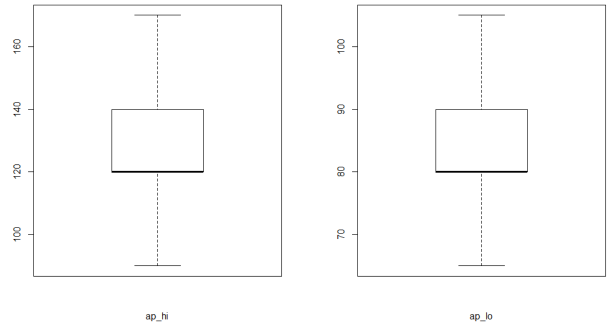


Fig.2 Boxplot of ap_hi and ap_lo before and after removal of outliers

iv. Conversion of data
   - Feature age originally was measured in days. We decided to convert it to become in years.

C. *Exploratory Data Analysis (EDA)*

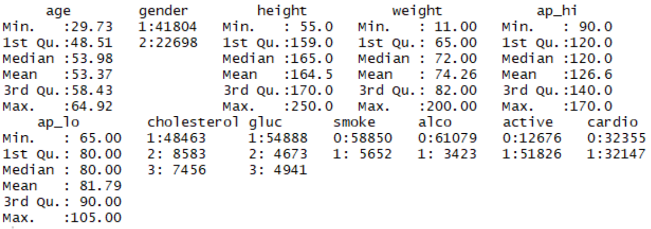
Having a clean dataset in hand, we need to understand the data, summarize its characteristics, and visualize it. Understanding the data is an iterative process between the data science team and the experts from the business side. It can help both sides to identify and construct important features, and later to build suitable machine learning models.

Some commonly used plots for EDA are:

i. Histograms: to check the distribution of a specific variable
ii. Univariate plots: to better understand each attribute.
iii. Multivariate plots: to better understand the relationships between attributes.
iv. Feature correlation plot (heatmap): to understand the dependencies between multiple variables

Statistical Summary

We take a look at a summary of each attribute. This includes the mean, the min and max values as well as some percentiles (25th, 50th or media and 75th e.g. values at these points if we ordered all the values for an attribute).



```
      age          gender        height          weight            ap_hi      
 Min.   :29.73   1:41804   Min.   : 55.0   Min.   : 11.00   Min.   : 90.0  
 1st Qu.:48.51   2:22698   1st Qu.:159.0   1st Qu.: 65.00   1st Qu.:120.0  
 Median :53.98             Median :165.0   Median : 72.00   Median :120.0  
 Mean   :53.37             Mean   :164.5   Mean   : 74.26   Mean   :126.6  
 3rd Qu.:58.43             3rd Qu.:170.0   3rd Qu.: 82.00   3rd Qu.:140.0  
 Max.   :64.92             Max.   :250.0   Max.   :200.00   Max.   :170.0  
     ap_lo         cholesterol gluc         smoke        alco         active       cardio   
 Min.   : 65.00   1:48463     1:54888   0:58850   0:61079   0:12676   0:32355  
 1st Qu.: 80.00   2: 8583     2: 4673   1: 5652   1: 3423   1:51826   1:32147  
 Median : 80.00   3: 7456     3: 4941                                          
 Mean   : 81.79                                                                
 3rd Qu.: 90.00                                                                
 Max.   :105.00                                                                
```

Fig.3 *Statistical summary of cardiovascular disease dataset.*

Histograms

We now have a basic idea about the data. We need to extend that with some visualizations. We start with some histograms to check the distribution of the numerical features. By looking at the shapes of the histogram, we can deduce that the numerical features somehow follow normal distribution.

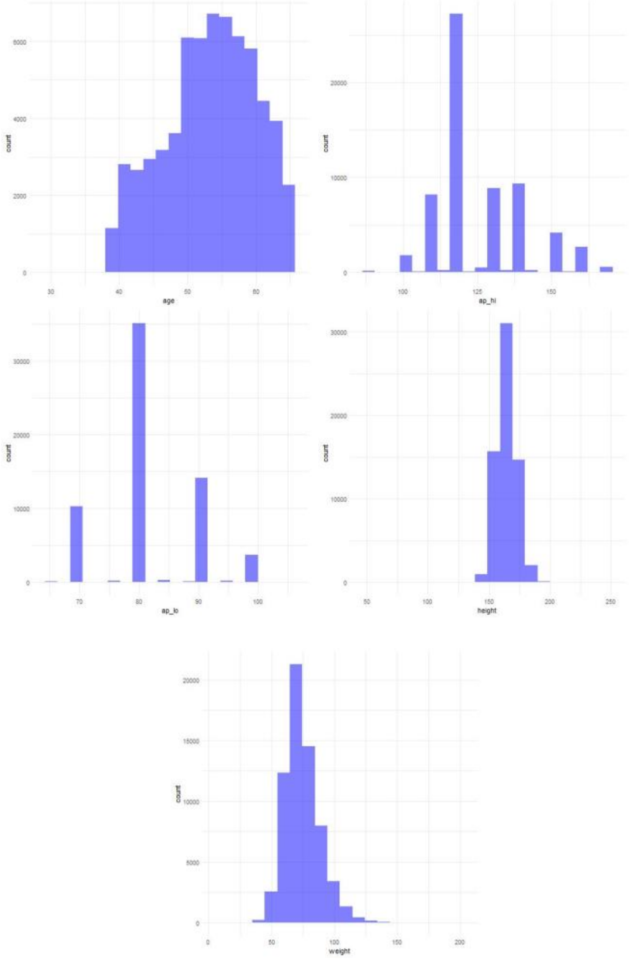
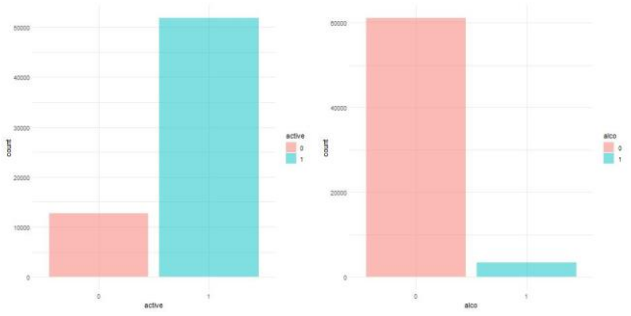


Fig.4 *Histograms of numerical features.*

Univariate Plots

Univariate analysis is the simplest form of analyzing data. It doesn't deal with causes or relationships (unlike regression) and its major purpose is to describe; it takes data, summarizes that data and finds patterns in the data. For categorical attributes we are using bar chart to visualize. By looking at the bar chart of cardio, this confirms what we learned in the last section, that the target classes are almost evenly distributed across the two classes.



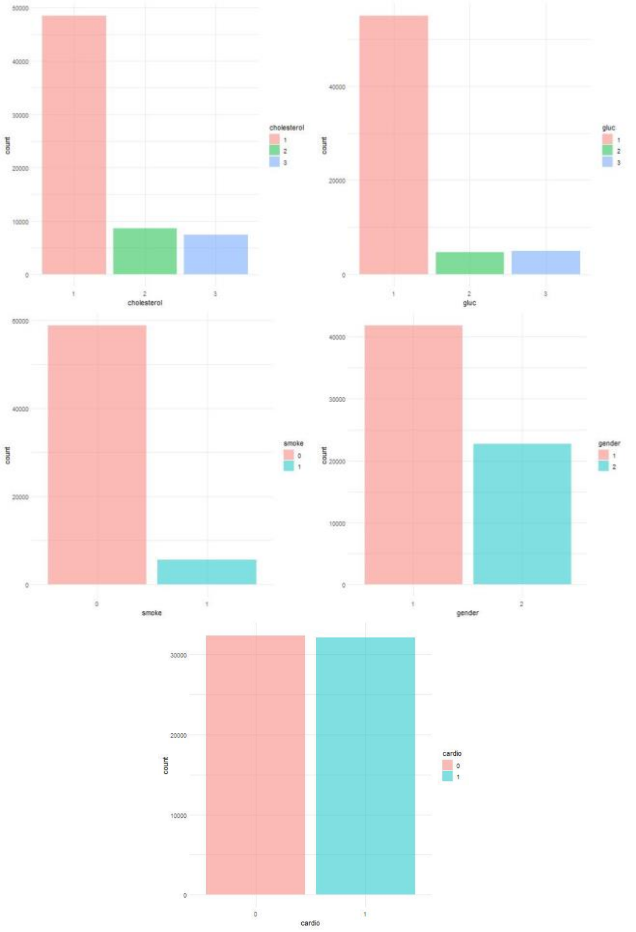
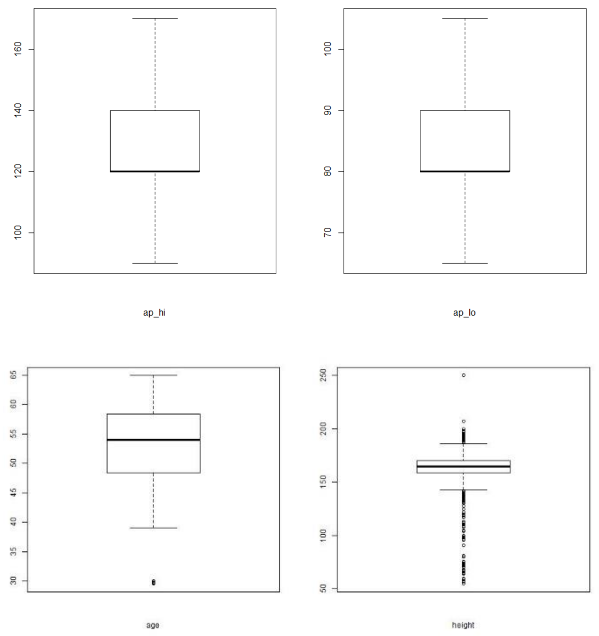


Fig.5 *Bar charts of categorical attributes.*

For numerical attributes, we are using a boxplot to visualize. This gives us a much clearer idea of the distribution of the numerical attributes and also the existence of outliers.



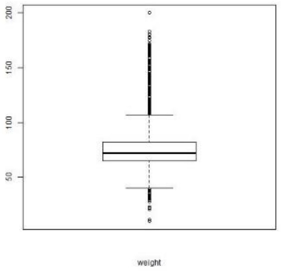
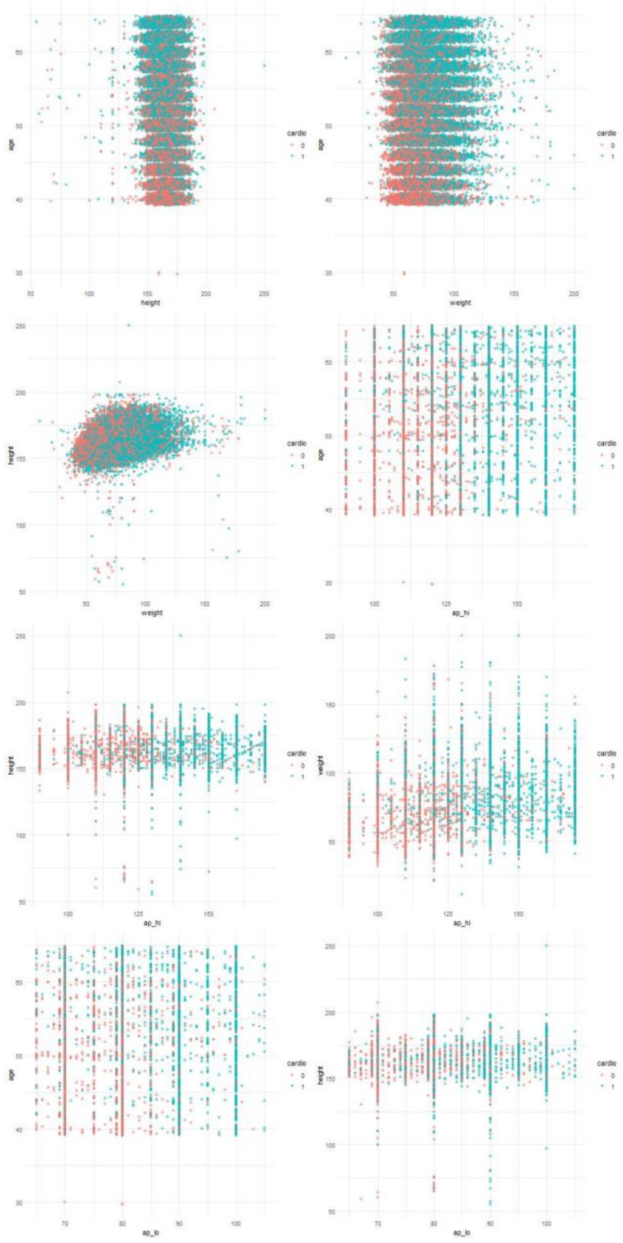


Fig.6 *Box plots of numerical attributes.*

Multivariate Plots

Now we can look at the interactions between the variables. First let’s look at scatterplots of all pairs of attributes and color the points by class. The scatterplots show that points for each class are generally difficult to be separated, we can’t really draw ellipses around them. We can’t see some clear relationships between the input attributes (trends) and between attributes and the class values.



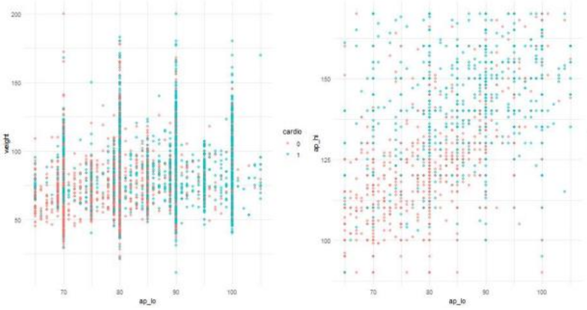


Fig.7 *Scatter plots of numerical attributes.*

Correlation Plot

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. In our correlation plot below, the observable pattern is that ap_lo and ap_hia are highly correlated (+0.71) with each other.

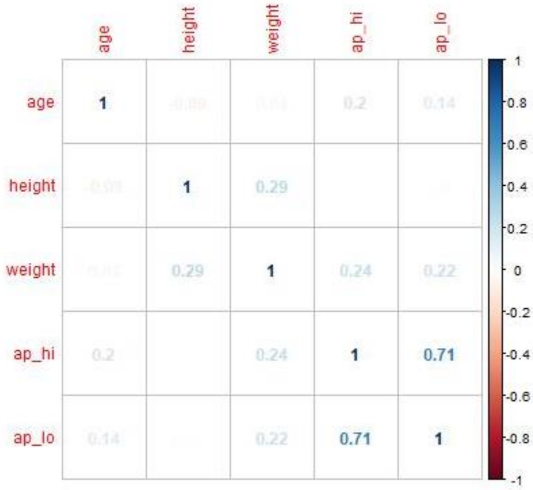


Fig.8 *Correlation plots of numerical attributes*

D. Model Researching

We don’t know which algorithms would be good on this problem or what configurations to use. We get an idea from the plots that all of the classes are quite difficult to be linearly separated, so we are expecting a bad result.

Let’s evaluate using 5 different algorithms:

1. Linear Discriminant Analysis (LDA)
2. Classification and Regression Trees (CART)
3. Naïve Bayes (NB)
4. k-Nearest Neighbors (KNN)
5. Random Forest (RF)

This is a good mixture of simple linear (LDA), nonlinear (NB, CART, KNN) and complex nonlinear methods (RF).

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a dimensionality reduction technique which is commonly used for the supervised classification problems. It is used for modeling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

For example, we have two classes and we need to separate them efficiently. Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure. So, we will keep on increasing the number of features for proper classification.

Classification and Regression Trees (CART)

A decision tree is a largely used non-parametric effective machine learning modeling technique for regression and classification problems. To find solutions a decision tree makes a sequential, hierarchical decision about the outcomes variable based on the predictor data.

The decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a decision tree with nodes and leaf nodes. The Understanding Level of Decision Tree algorithm is so easy as compared to the classification algorithm.

In the Decision tree algorithm, we solve our problem in a tree regression. Each internal node of the tree corresponds to an attribute. Each leaf node corresponds to a Class Label. In the decision tree for predicting a class label for a record, we start from the root of the tree. We compare the value of the root attribute with the record’s attribute on the basis of comparison. We follow the branch corresponding to that value & jump to the next node. We continue comparing our record’s attribute value with other internal nodes of the tree until we reach a leaf node.

Naïve Bayes (NB)

Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.

k-Nearest Neighbors (KNN)

KNN is a model that classifies data points based on the points that are most similar to it. It uses test data to make an “educated guess” on what an unclassified point should be classified as.

KNN is an algorithm that is considered both non-parametric and an example of lazy learning. Non-parametric means that it makes no assumptions. The model is made up entirely from the data given to it rather than assuming its structure is normal. Lazy learning means that the algorithm makes no generalizations. This means that there is little training involved when using this method. Because of this, all of the training data is also used in testing when using KNN.

Random Forest (RF)

Random forest is an ensemble machine learning algorithm that is used for classification and regression problems. Random forest applies the technique of bagging (bootstrap aggregating) to decision tree learners. There are many reasons why the random forest is so popular. These reasons are:

- Ensemble learning prevents overfitting of data
- Bootstrapping enables the random forest to work well on relatively small datasets
- Predictors can be trained in parallel
- Decision tree learning enables automatic feature selection

E. Model Training

Now it is time to create some models of the data. Here is what we are going to cover in this phase:

1. Split the dataset into the train set and test set.
2. Apply feature scaling on both train and test sets.
3. Set-up the test harness to use 10-fold cross-validation.
4. Build 5 different models and hyperparameter tuning.

Train-Test Split

We need to know that the model we created is any good. Later, we will use statistical methods to estimate the accuracy of the models that we create on unseen data. We also want a more concrete estimate of the accuracy of the best model on unseen data by evaluating it on actual unseen data.

That is, we are going to hold back some data that the algorithms will not get to see and we will use this data to get a second and independent idea of how accurate the best model might actually be. We will split the loaded dataset into two, 75% of which we will use to train our models and 25% that we will hold back as a test dataset.

Feature Scaling

Cardiovascular disease dataset contains features that highly vary in magnitudes, units, and range. Normalization should be performed when the scale of a feature is irrelevant or misleading and not should Normalize when the scale is meaningful.

The algorithms which use Euclidean Distance measure are sensitive to Magnitudes. Here feature scaling helps to weigh all the features equally.

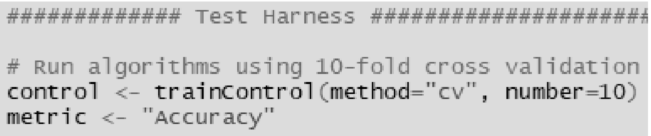
Formally, If a feature in the dataset is big in scale compared to others then in algorithms where Euclidean distance is measured this big scaled feature becomes dominating and needs to be normalized.

KNN requires feature scaling. Meanwhile, Naive Bayes, Linear Discriminant Analysis, and Tree-Based models such as CART and Random Forest are not affected by feature scaling. In Short, any Algorithm which is Not Distance-based is Not affected by Feature Scaling.

Test Harness

We will 10-fold cross-validation to estimate accuracy.

This will split our dataset into 10 parts, train in 9 and test on 1 and release for all combinations of train-test splits. We will also repeat the process 3 times for each algorithm with different splits of the data into 10 groups, in an effort to get a more accurate estimate.

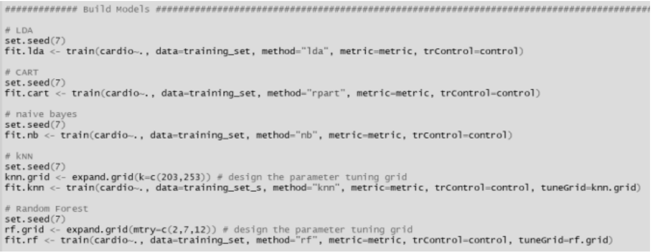


```
############# Test Harness ######################

# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

Fig.9 R code for train-control using 10-fold cross-validation and accuracy as metric

Build Models and Hyperparameter Tuning

Let’s build our five models:



```
############# Build Models ##############################################################################

# LDA
set.seed(7)
fit.lda <- train(cardio~., data=training_set, method="lda", metric=metric, trControl=control)

# CART
set.seed(7)
fit.cart <- train(cardio~., data=training_set, method="rpart", metric=metric, trControl=control)

# naive bayes
set.seed(7)
fit.nb <- train(cardio~., data=training_set, method="nb", metric=metric, trControl=control)

# kNN
set.seed(7)
knn.grid <- expand.grid(k=c(203,253)) # design the parameter tuning grid
fit.knn <- train(cardio~., data=training_set_s, method="knn", metric=metric, trControl=control, tuneGrid=knn.grid)

# Random Forest
set.seed(7)
rf.grid <- expand.grid(mtry=c(2,7,12)) # design the parameter tuning grid
fit.rf <- train(cardio~., data=training_set, method="rf", metric=metric, trControl=control, tuneGrid=rf.grid)
```

Fig.10 R code for training algorithm using Caret package

Caret does support the configuration and tuning of the configuration of each model. However due to high computing time, we will only tune the KNN (k value) and Random Forest (mtry value). Fork value, we will randomly choose two values (203 and 253) for parameter tuning. Meanwhile, for Random Forest mtry value, we will randomly choose three values (5,7,12) for parameter tuning. Mtry parameter is the number of variables available for splitting at each tree node.

We reset the random number seed before reach run to ensure that the evaluation of each algorithm is performed using exactly the same data splits. It ensures the results are directly comparable.

F. Model Evaluation

We are using the metric of “Accuracy” to evaluate models. This is a ratio of the number of correctly predicted instances divided by the total number of instances in the dataset multiplied by 100 to give a percentage (e.g. 95% accurate). It works well only if there are an equal number of samples belonging to each class.

We also measure the computing time of the train and test process of each model. Due to the high computing time, we only measure three replications of training and testing expressions.

G. Prediction

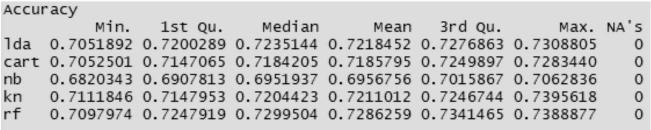
Now we want to get an idea of the accuracy of the model on our test set. This will give us an independent final check on the accuracy of the best model. It is valuable to keep a test set just in case we made a slip during such as overfitting to the training set or a data leak. Both will result in an overly optimistic result.

We can run the best model directly on the test set and summarize the results in a confusion matrix.

IV. RESULTS, MODEL AND DISCUSSION

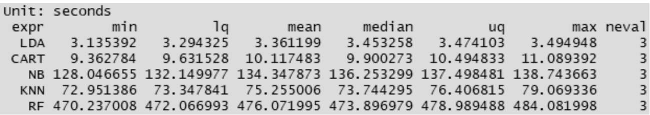
A. Comparison of Models Performance

We now have 5 models and accuracy estimations for each. We need to compare the models to each other and select the most accurate.



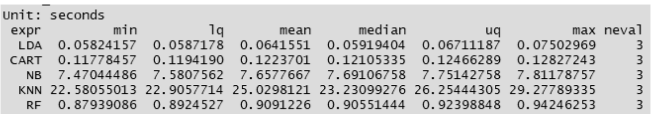
```
Accuracy
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
lda  0.7051892 0.7200289 0.7235144 0.7218452 0.7276863 0.7308805    0
cart 0.7052501 0.7147065 0.7184205 0.7185795 0.7249897 0.7283440    0
nb   0.6820343 0.6907813 0.6951937 0.6956756 0.7015867 0.7062836    0
kn   0.7111846 0.7147953 0.7204423 0.7211012 0.7246744 0.7395618    0
rf   0.7097974 0.7247919 0.7299504 0.7286259 0.7341465 0.7388877    0
```

Fig.11 Summary accuracy of models



```
Unit: seconds
 expr        min         lq       mean     median         uq        max neval
  LDA   3.135392   3.294325   3.361199   3.453258   3.474103   3.494948     3
 CART   9.362784   9.631528  10.117483   9.900273  10.494833  11.089392     3
   NB 128.046655 132.149977 134.347873 136.253299 137.498481 138.743663     3
  KNN  72.951386  73.347841  75.255006  73.744295  76.406815  79.069336     3
   RF 470.237008 472.066993 476.071995 473.896979 478.989488 484.081998     3
```

Fig.12 Summary training time of models



```
Unit: seconds
 expr         min         lq       mean      median          uq         max neval
  LDA  0.05824157  0.0587178  0.0641551  0.05919404  0.06711187  0.07502969     3
 CART  0.11778457  0.1194190  0.1223701  0.12105335  0.12466289  0.12827243     3
   NB  7.47044486  7.5807562  7.6577667  7.69106758  7.75142758  7.81178757     3
  KNN 22.58055013 22.9057714 25.0298121 23.23099276 26.25444305 29.27789335     3
   RF  0.87939086  0.8924527  0.9091226  0.90551444  0.92398848  0.94246253     3
```

Fig.13 Summary testing time of models

As shown in Figure 14, it can be visualized that the highest accuracy of 72.86% was obtained using the RF as the classification technique. Then, RF is followed by LDA and KNN with 72.18% and 72.11% accuracies respectively. As we can see, RF obtains the highest classification accuracy. However, it also takes a long time for RF.

Table 2 Performance of Machine Learning Models Using Different Classification Methods

Model	Mean			
	Accuracy	Training Time (s)	Testing Time (s)	Total Running Time (s)
LDA	72.18%	3.36	0.06	3.42
CART	71.86%	10.12	0.12	10.24
NB	69.57%	134.35	7.66	142.01
KNN	72.11%	75.26	23.23	98.49
RF	72.86%	476.07	0.91	476.98

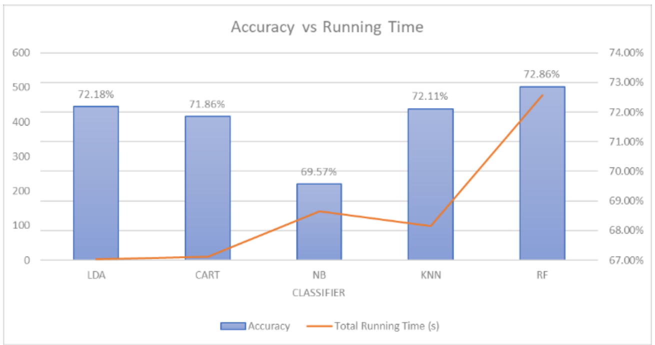
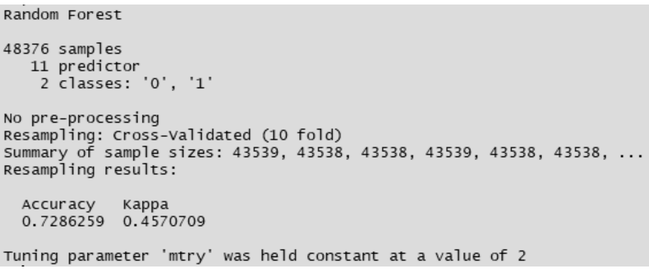


Fig.14 Accuracy v Total Running Time plot

B. Prediction using Test Set

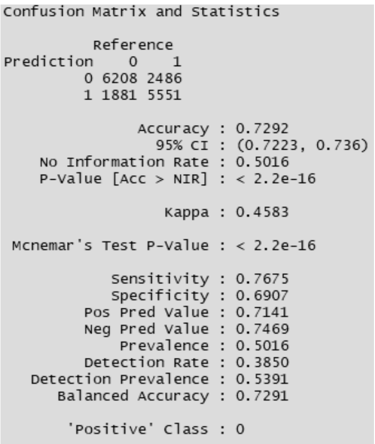
We can run the best accuracy model which is the RF model directly on the test set and summarize the results in a confusion matrix.



```
Random Forest

48376 samples
   11 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 43539, 43538, 43538, 43539, 43538, 43538, ...
Resampling results:

  Accuracy   Kappa
  0.7286259  0.4570709

Tuning parameter 'mtry' was held constant at a value of 2
```

Fig.15 Random Forest Model Summary

We can see that the accuracy is 72.92%. It was a small test dataset (25%), but this result is within our expected margin of 72.86% +/-4% suggesting we may have an accurate and reliably accurate model.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6208 2486
         1 1881 5551

               Accuracy : 0.7292
                 95% CI : (0.7223, 0.736)
    No Information Rate : 0.5016
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4583

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7675
            Specificity : 0.6907
         Pos Pred Value : 0.7141
         Neg Pred Value : 0.7469
             Prevalence : 0.5016
         Detection Rate : 0.3850
   Detection Prevalence : 0.5391
      Balanced Accuracy : 0.7291

       'Positive' Class : 0
```

Fig.16 Confusion Matrix and Statistics

C. Feature Importance

Random forests are among the most popular machine learning methods thanks to their relatively good accuracy, robustness, and ease of use. Most importantly, they also provide the feature importance of a dataset.

As shown in Figure 17, we can see the top 3 most important features are:

1. ap_hi
2. age
3. ap_lo

What seems surprising though is that cholesterol values turned out to be not so important compared to the age

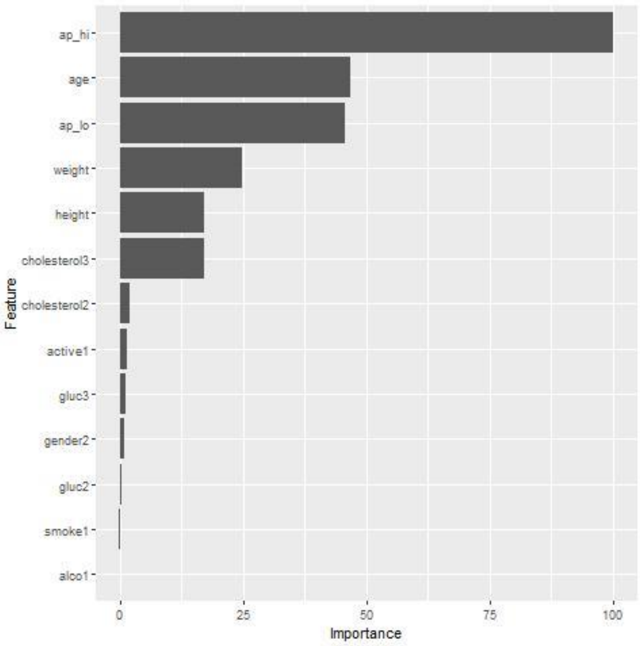


Fig.17 Feature Importance

D. Challenges

The biggest challenge associated with training this dataset is that of high training time, limiting the number of models that can be trained. Of all the trained models, the Random Forest model is seen to consume the most time given their computational complexity. Support Vector Machine model was excluded from the consideration as it is expected to consume even more time than the Random Forest model.

Given the high computation time, building and evaluating models over large hyper-parameter ranges are time expensive. For example, hyper-parameter tuning of the KNN takes close to min per cycle. Some common practices are resorted to while selecting hyper-parameters of time expensive models. For example, the value of k is set to be odd and equal to the square root of the number of samples for K-Nearest Neighbors (k = 253). We also restricted the parameter grid of Random Forest to only three values as one training process of RF could take up to 739s.

V. CONCLUSION AND FUTURE WORK

Cardiovascular disease is difficult to predict using current medical practices like periodic health screening. It would be helpful if the doctor can predict the presence and absence of heart disease and classify the patients as different groups. The results in the study indicate that machine learning algorithms are able to predict presence or absence of heart disease with satisfactory accuracy. In this study, a detailed comparison of different machine learning algorithms was used to determine the best performing model for detecting cardiovascular disease. Overall, Random Forest has the highest accuracy and also has the highest average running time.

Many different hyperparameter tuning, dimensional or feature reduction, and more advanced machine learning models have been left for the future due to lack of time (i.e. the experiments with training usually very time consuming, requiring even days to finish a single run). The following ideas could be tested in future work:

1. Perform feature reduction. As shown in Figure 17, we can reduce our number of features by almost half by only choosing ap_hi, age, ap_lo, weight, height, and cholesterol. By doing so it will help to lower the computational complexity and speed up the training process.
2. Using more values for hyperparameter tuning. In previous study, hyperparameter tuning has not been performed thoroughly due to high training time. However, for this time more values can be chosen after we speed up the training time by using feature reduction.
3. Using more advanced machine learning algorithms such as Support Vector Machine and Neural Network.

VI. REFERENCES

[1] World Health Organization, Cardiovascular diseases,
https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] Kaggle, Cardiovascular Disease dataset,
https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

[3] Wikipedia, Cardiovascular disease,
https://en.wikipedia.org/wiki/Cardiovascular_disease

[4] Mendis S, Puska P, Norrving B (2011). Global Atlas on Cardiovascular Disease Prevention and Control (PDF). World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization.

[5] R. Subha, K. Anandakumar, A. Bharathi, Study on Cardiovascular Disease Classification Using Machine Learning Approaches

[6] Soni, Jyoti, Ujma Ansari, Dipesh Sharma, and Sunita Soni. "Predictive data mining for medical diagnosis: An overview of heart disease prediction. "International Journal of Computer Applications 17, no. 8 (2011): 43-48.

[7] Austin, Peter C., Douglas S. Lee, Ewout W. Steyerberg, and Jack V. Tu. "Regression trees for predicting mortality in patients with cardiovascular disease: What improvement is achieved by using ensemble ‐ based methods?. "Biometrical journal 54, no. 5 (2012): 657-673.

[8] Ozcift, Akin, and Arif Gulten. "Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. " Computer methods and programs in biomedicine 104, no. 3 (2011): 443-451.

[9] Yeh, Duen-Yian, Ching-Hsue Cheng, and Yen-Wen Chen. "A predictive model for cerebrovascular disease using data mining. " Expert Systems with Applications 38, no. 7 (2011): 8970-8977.

[10] Blood Pressure UK,
http://www.bloodpressureuk.org/microsites/u40/Home/facts/Whatisnormal