MATH 6359

Project Part 2: Project Proposal (Group 8)

Group Members: Elizabeth Garcia, Hussein Gaziza, Sara Nafaryeh, Tony Nguyen, Thomas Su

Abstract:

We all are aware of how machine learning has revolutionized our world in recent years and has made a variety of complex tasks much easier to perform. The recent breakthroughs in implementing Deep learning techniques has shown that superior algorithms and complex architectures can impart human-like abilities to machines for specific tasks. Several machine learning algorithms have been increasingly utilized for cardiovascular disease prediction. We aim to assess and summarize the overall predictive ability of ML algorithms in cardiovascular diseases.

We also hope to use hypothesis testing to determine if cholesterol and high blood pressure contribute to coronary heart disease.

Research Questions:

- How does age and gender affect coronary heart disease?
- Which total cholesterol level is associated with high risk for heart disease?
- What level of blood pressure can cause a heart disease?
- Which Machine Learning Techniques accurately predicts the leading features causing heart disease?

Machine Learning Techniques (Classification):

- PCA
- Decision Tree
- Random Forest
- K-nearest neighbor (KNN)

Data Source:

https://www.kaggle.com/ronitf/heart-disease-uci

Dataset Description:

The dataset has 303 observations and 14 attributes (13 attributes and 1 target column):

1. Age is the age of candidate.
2. Sex has numeric values. 1 is male and 0 is female.
3. (cp) Chest Pain pain has values between 0-3. The types of angina that are described in the research paper. The higher the number, the lesser are the odds of heart attack. Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain.
4. (trestbps) Resting blood pressure is normal pressure with no exercise.
5. (chol) Cholesterol means the blockage for blood supply in the blood vessels.
6. (fbs) Fasting blood sugar > 120 mg/dl (1 = true; 0 = false) blood sugar taken after a long gap between a meal and the test. Typically, it's taken before any meal in the morning.
7. (restecg) Rest ECG results means ECG values taken while a person is on rest which means no exercise and normal functioning of heart is happening.
8. (thalach) The Maximum Heart Rate achieved.
9. (exang) Exercise induced angina (1 = yes; 0 = no) is chest pain while exercising or doing any physical activity.
10. (oldpeak) ST Depression is the difference between the value of ECG at rest and after exercise.
11. (slope) The slope of the peak exercise ST segment. Value 1: upsloping, Value 2: flat, Value 3: downsloping.
12. (ca) The number of major blood vessels (0-3) supplying blood to the heart is blocked.
13. (thal) The Types of thalassemia (3 = normal; 6 = fixed defect; 7 = reversible defect).
14. (target) (predicted attribute): diagnosis of heart disease (angiographic disease status): Value 0: < 50% diameter narrowing, Value 1: > 50% diameter narrowing.

PCA

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal. Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling.

Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules. Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. While making decision tree, at each node of tree we ask different type of questions. Based on the asked question we will calculate the information gain corresponding to it.

Random Forest

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity. Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the

“bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

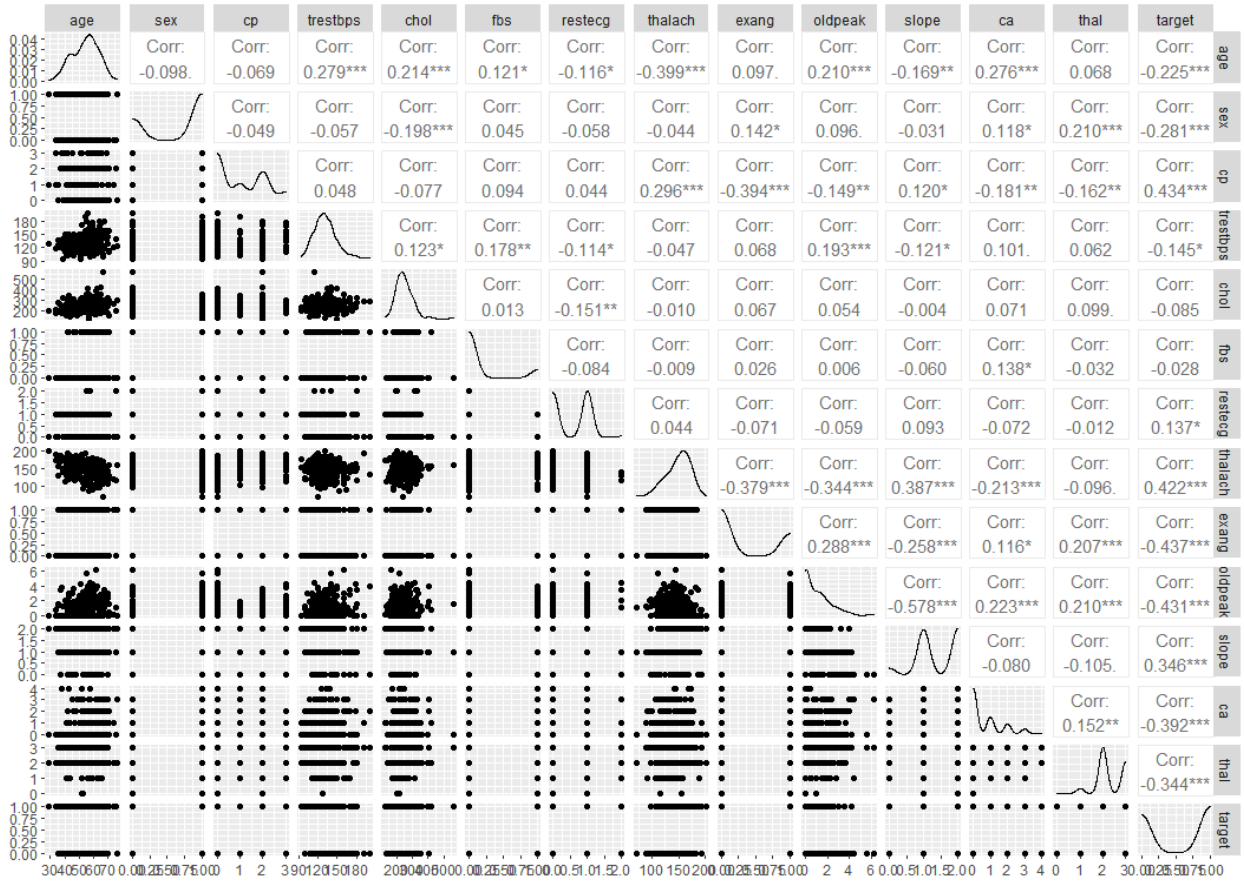
Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

K-nearest neighbor (KNN)

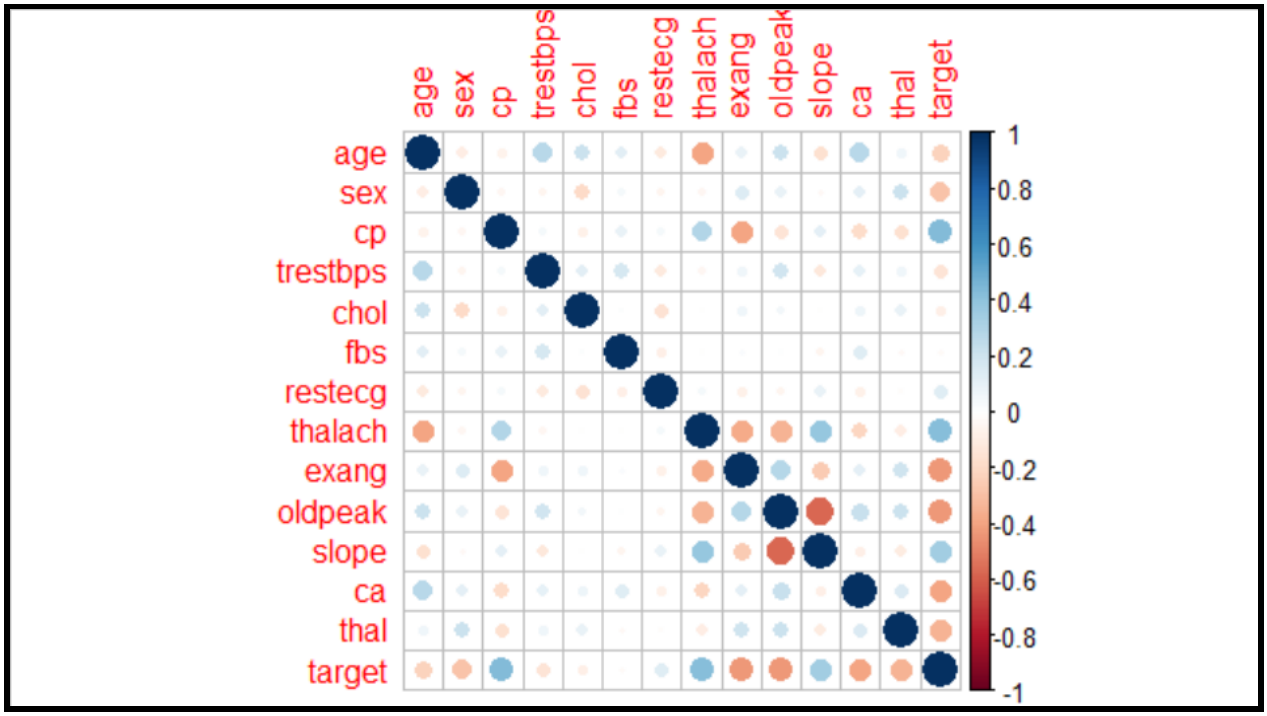
K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problems. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors. The data is assigned to the class which has the nearest neighbors. As you increase the number of nearest neighbors, the value of k, accuracy might increase. It uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

Correlation Matrix

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.
Using the ggpairs() function in the R package ggplot2, we are able to display a scatter plot matrix, where our correlations are shown in the top right triangle, and numeric variables are drawn on the bottom left half of the triangle. Through a quick overview, there doesn't appear to be a strong relationship between the variables. The highest value appears to be slope and oldpeak at -0.578.



We can also use the function corrplot() in the R package corrplot to generate a heatmap. By doing so we can view the data in a visually colorful way and identify possible multicollinearity. The color in the plot varies based on the sign/ direction of the correlation: blue for positive and red for negative signs. While the size of the circle demonstrated the strength of the correlation: the larger the circle, the closer it is to 1 (or -1). Once again we can see that the highest relationship value between two variables appears to be slope and oldpeak at -0.578.




```
> sample = sample.split(heart$target, SplitRatio = .80)
> train = subset(heart, sample == TRUE)
> test  = subset(heart, sample == FALSE)
> dim(train)
[1] 242  14
> dim(test)
[1] 61 14
```

```
> model1 <- randomForest(as.factor(target) ~ ., data = train, importance = TRUE);model1

Call:
 randomForest(formula = as.factor(target) ~ ., data = train, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 19.01%
Confusion matrix:
   0   1 class.error
0 87  23   0.2090909
1 23 109   0.1742424
> model2 <- randomForest(as.factor(target) ~ ., data = train, ntree = 500, mtry = 6, importance = TRUE);model2

Call:
 randomForest(formula = as.factor(target) ~ ., data = train, ntree = 500,      mtry = 6, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 6

        OOB estimate of  error rate: 19.01%
Confusion matrix:
   0   1 class.error
0 86  24   0.2181818
1 22 110   0.1666667
```

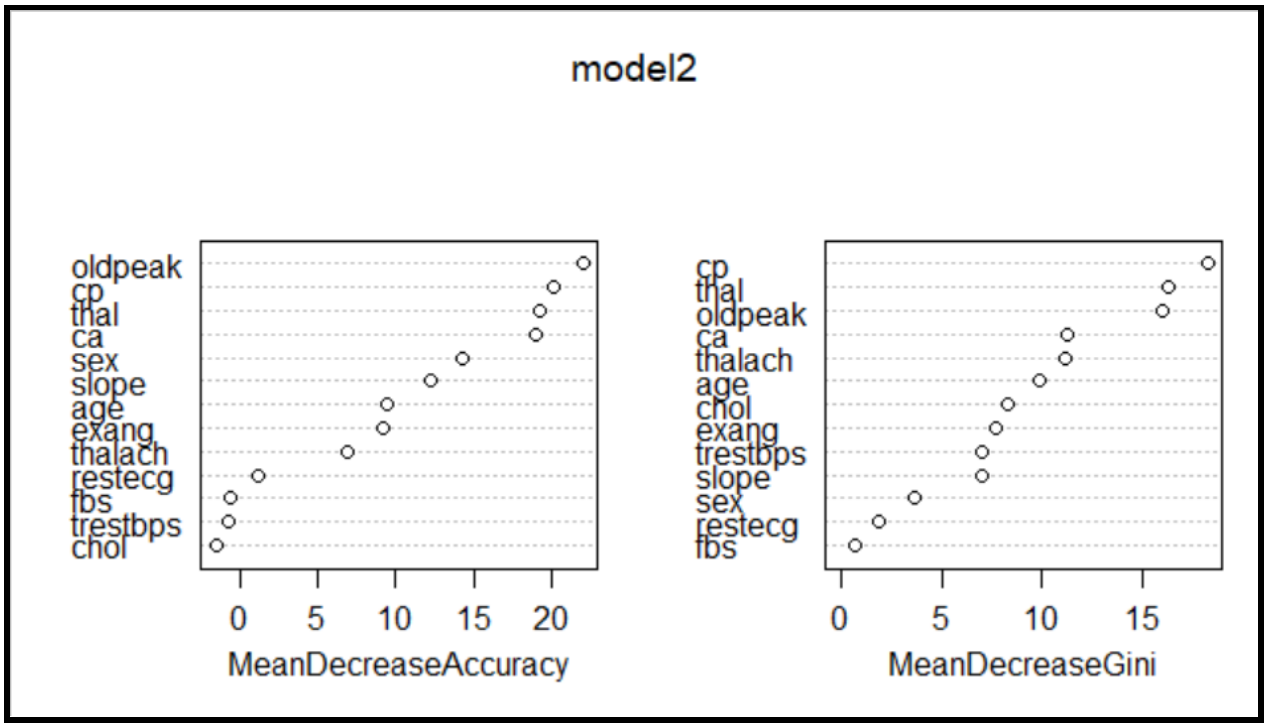
```
> predTrain <- predict(model2, train, type = "class");predTrain
  1   2   3   4   5   6   7   9  10  12  13  14  16  17  18  20  21  22  23  24  25  26  27  28  29  30  31
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
 32  33  34  35  36  37  38  39  40  41  42  44  45  46  47  48  49  51  52  53  54  55  56  57  59  60  61
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
 62  63  64  65  68  69  71  72  73  74  75  76  79  81  82  83  84  85  87  88  89  93  94  96  97  98  99
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
100 103 104 105 106 107 108 109 110 111 113 114 115 116 117 118 119 121 122 124 125 127 128 129 130 131 132
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
134 135 136 137 140 141 142 143 144 145 146 147 149 150 151 152 156 157 158 159 161 162 163 164 166 167 168
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0
169 170 172 173 174 176 177 178 179 181 182 183 184 185 186 188 189 191 192 193 194 195 197 198 199 200 201
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
202 203 204 205 206 207 208 209 210 211 213 214 215 217 219 220 221 222 223 224 225 226 227 228 229 230 232
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
233 234 238 239 240 242 243 244 245 246 247 248 249 250 251 254 255 256 257 258 259 260 261 262 264 266 267
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
268 270 271 272 273 274 277 278 280 282 283 285 287 289 290 292 293 294 295 296 297 298 299 300 302 303
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Levels: 0 1
```

```
> table(predTrain, train$target)

predTrain   0   1
        0 110   0
        1   0 132
```

```
> predValid <- predict(model2, test, type = "class");predValid
  8  11  15  19  43  50  58  66  67  70  77  78  80  86  90  91  92  95 101 102 112 120 123 126 133 138 139
  1   1   1   1   0   1   1   1   1   1   1   1   0   0   1   1   0   1   1   0   1   1   1   1   1   1   0
148 153 154 155 160 165 171 175 180 187 190 196 212 216 218 231 235 236 237 241 252 253 263 265 269 275 276
  1   0   1   1   1   1   1   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
279 281 284 286 288 291 301
  1   0   1   0   1   0   0
Levels: 0 1
```

```
> mean(predValid == test$target)
[1] 0.7868852
> table(predValid,test$target)

predValid  0  1
        0 22  7
        1  6 26
```

```
> importance(model2)
                0           1 MeanDecreaseAccuracy MeanDecreaseGini
age       2.926042  9.17111497            9.3985940        9.8874852
sex       8.533745 11.96452065           14.3066192        3.6804451
cp       16.291657 14.99043830           20.1394299       18.2616521
trestbps -1.274220  0.03010697           -0.7325302        7.0450543
chol     -1.789650 -0.63459020           -1.5101917        8.3309822
fbs      -2.982211  1.89976796           -0.5772116        0.7222282
restecg   1.977869 -0.17042801            1.1605922        1.9172562
thalach   5.774175  4.34235124            6.8830507       11.1898495
exang     8.417915  4.73988313            9.2204669        7.7453285
oldpeak  16.061330 15.21029338           22.0075945       16.0197449
slope    11.598763  5.83190725           12.2352634        7.0228186
ca       10.419983 17.28652924           18.9744693       11.2640207
thal     10.834328 18.01867434           19.2542222       16.3049749
```


model2
oldpeak
cp
thal
ca
sex
slope
age
exang
thalach
restecg
fbs
trestbps
chol
0
5
10
15
20
MeanDecreaseAccuracy
cp
thal
oldpeak
ca
thalach
age
chol
exang
trestbps
slope
sex
restecg
fbs
0
5
10
15
MeanDecreaseGini