

Heart Disease Analysis Project

HarvardX Data Science Capstone Project

Florian Seka

28/05/2019

Contents

1	Introduction	2
1.1	Project Presentation	2
1.2	Purpose and plan	2
1.3	Executive summary	2
1.4	Additionnal information	2
1.5	Raw dataset description	2
1.5.1	Data Source and structure	2
1.5.2	Feature description	3
2	Analysis	3
2.1	Used libraries	3
2.2	Data preparation	4
2.3	Building the training and validation	6
2.4	Feature exploration	6
2.4.1	Continuous features	6
2.4.2	Categorical features	9
2.5	Conclusion - features selected for the learning process	14
2.6	Strategy and methods	14
2.7	Models evaluation	15
2.7.1	Generalized Linear Model	15
2.7.2	K Nearest Neighbours	15
2.7.3	Random Forest	16
2.7.4	Regression Trees	17
2.8	Comparison of methods	18
2.9	Broader Model evaluation	19
3	Final Results	19
3.1	An exploration: approaching the missing data	20
3.1.1	Visualisation of the missing data	20
3.1.2	Imputing the data and training the GLM model	21
3.1.3	Completing the validation data set and predicting the disease presence	21
4	Conclusion	22
5	References and Credits	23
5.1	References	23
5.2	Credits	23

1 Introduction

1.1 Project Presentation

The subject chosen for this capstone project is the Heart Disease Data Set from the UCI Machine learning repository. This dataset gathers data from four different medical institution and provides, for each patient, a number of attributes, as well as a measure of heart disease presence (from “no presence” to different levels of presence). Our objective will be to predict, given a patients set of attributes (medical parameters), the presence or absence of medical disease.

This dataset has also been used in Data Science competitions such as Kagle. However, only the processed data from the Cleveland Clinic Foundation was used. Our study attempts to carry the data analysis on the concatenation of all datasets made available on the UC Irvine Machine Learning Repository.

1.2 Purpose and plan

The purpose of this report is to document the various strategies investigated in the frame of the data science capstone final project, in partial fulfilment of the requirements for the EDX Harvard Data Science Professional Certificate.

In the first part of the report, information is briefly given about the structuring and preparation of the data used to build and evaluate the machine learning algorithms, as well as the different methods that will be developed to achieve the best possible final result. In the second part, the final, optimised, algorithm is selected and tested against the validation dataset.

A final investigation of the data is provided, with regard to the rather important number of missing values in the dataset. A predictive mean matching algorithm is used to complete the datasets in order to allow a prediction.

1.3 Executive summary

After having determined the most important features in the dataset, and tested a variety of machine learning methods, the selected method was the **Genralized Linear Model (GLM)**, achieving an accuracy of about 87.5%. A further analysis also presented in this report extended the number of methods being investigated and confirmed that the GLM was the most suited for this experiment.

1.4 Additionnal information

The analysis was carried out using the R langage in the R Studio developement environment. All scripts and data in support of this project are available for download from a dedicated version control repository:

<https://github.com/fseka/HeartDisease>

1.5 Raw dataset description

1.5.1 Data Source and structure

The data used for this experiment was created by four medical research entities:

- Hungarian Institute of Cardiology.
- University Hospital, Zurich, Switzerland
- University Hospital, Basel, Switzerland
- V.A. Medical Center, Long Beach and Cleveland Clinic Foundation

The data was retrieved from the UCI Machine Learning Repository <https://archive.ics.uci.edu/ml/datasets/heart+disease> and consists of four individual files in the same format and structure:

- processed.cleveland.data
- processed.hungarian.data

- processed.switzerland.data
- processed.va.data

In the data, each row correspond to a patient and each of the 13 columns to an attribute, the 14th column being the rating of heart disease presence (from 0 to 4, 0 corresponding to the absence of disease).

1.5.2 Feature description

The 13 features in the dataset were described in the repository archive and are as follows:

1. **age**: The patient age
2. **sex**: The patient sex
3. **cp**: Chest pain type (from 1 to 4, in 1 increments)
4. **trestbps**: Resting blood pressure (in mm Hg on admission to the hospital)
5. **chol** serum cholestoral in mg/dl
6. **fbs**: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. **restecg**: resting electrocardiographic results
 - i) Value 0: normal
 - ii) Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
 - iii) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. **thalach**: maximum heart rate achieved
9. **exang**: exercise induced angina
10. **oldpeak**: ST depression induced by exercise relative to rest
11. **slope**: the slope of the peak exercise ST segment
 - i) Value 1: upsloping
 - ii) Value 2: flat
 - iii) Value 3: downsloping
12. **ca**: number of major vessels (0-3) colored by flourosopy
13. **thal**: Thallium stress test result : 3 = normal; 6 = fixed defect; 7 = reversable defect

2 Analysis

2.1 Used libraries

The analysis and predictions will rely on the use of dedicated functions available in the following packages.

```
# Cleaning workspace
rm(list=ls())
# Loading the used libraries
library(tidyverse)
library(caret)
library(ggrepel)
library(ggthemes)
library(mice)
library(randomForest)
library(rpart)
library(rpart.plot)
library(ROCR)
library(mgcv)
library(VIM)
```

2.2 Data preparation

The data provided by the UCI repository first needs to be gathered and prepared. In a first step, the individual datasets from each research institutes are appended to one another to build a complete dataset that will be used for the analysis.

```
datasetdir<-file.path(getwd(),"Data")
datacolnames <- c("age","sex","cp","trestbps","chol","fbs","restecg","thalach","exang",
  "oldpeak",
  "slope",
  "ca",
  "thal",
  "target") #defining the column names

clevelanddata<-read.csv(file.path(datasetdir,"processed.cleveland.data"), header=FALSE)
swissdata<-read.csv(file.path(datasetdir,"processed.switzerland.data"), header=FALSE)
vadata<-read.csv(file.path(datasetdir,"processed.va.data"), header=FALSE)
hungariandata<-read.csv(file.path(datasetdir,"processed.hungarian.data"), header=FALSE)

# assign the header names
names(clevelanddata)<-datacolnames
names(swissdata)<-datacolnames
names(vadata)<-datacolnames
names(hungariandata)<-datacolnames

# before concatenating, adding one column to indicate origin of data
clevelanddata<- mutate (clevelanddata, institute="Cleveland")
swissdata<- mutate (swissdata, institute="Switzerland")
vadata<- mutate (vadata, institute="Longbeach")
hungariandata<- mutate (hungariandata, institute="Hungary")

# build a unique dataset concatenating the rows from all previous 4 research institutes
heartdisease<-rbind(clevelanddata,swissdata,vadata,hungariandata)

# We can now remove the individual datasets and variables to free some workspace memory
rm(clevelanddata,swissdata,vadata,hungariandata,datacolnames,datasetdir)
```

In this process, factors are added to the features based on the data description. This will later facilitate the creation and labelling of plots with the ggplot() package.

```
heartdisease$sex<-as.factor(heartdisease$sex)
heartdisease$fbs<-as.factor(as.numeric(as.character(heartdisease$fbs)))
heartdisease$exang<-as.factor(as.numeric(as.character(heartdisease$exang)))

heartdisease$restecg<-as.factor(as.numeric(heartdisease$restecg))
heartdisease$slope<-as.factor(as.numeric(as.character(heartdisease$slope)))
heartdisease$thal<-as.factor(as.numeric(as.character(heartdisease$thal)))
heartdisease$ca<-as.factor(as.numeric(as.character(heartdisease$ca)))
heartdisease$cp<-as.factor(as.numeric(as.character(heartdisease$cp)))

heartdisease$institute<-as.factor(as.character(heartdisease$institute))

levels(heartdisease$sex)[levels(heartdisease$sex)==0] <- "Female"
```

```

levels(heartdisease$sex)[levels(heartdisease$sex)==1] <- "Male"

levels(heartdisease$fbs)[levels(heartdisease$fbs)==0] <- "Fasting_Blood_Sugar_smaller_120"
levels(heartdisease$fbs)[levels(heartdisease$fbs)==1] <- "Fasting_Blood_Sugar_greater_120"

levels(heartdisease$exang)[levels(heartdisease$exang)==0] <- "No_Exercise_Induced_Angina"
levels(heartdisease$exang)[levels(heartdisease$exang)==1] <- "Exercise_Induced_Angina"

levels(heartdisease$restecg)[levels(heartdisease$restecg)==0] <- "REST_ECG_0"
levels(heartdisease$restecg)[levels(heartdisease$restecg)==1] <- "REST_ECG_1"
levels(heartdisease$restecg)[levels(heartdisease$restecg)==2] <- "REST_ECG_2"

levels(heartdisease$cp)[levels(heartdisease$cp)==1] <- "ChestPaintype_1"
levels(heartdisease$cp)[levels(heartdisease$cp)==2] <- "ChestPaintype_2"
levels(heartdisease$cp)[levels(heartdisease$cp)==3] <- "ChestPaintype_3"
levels(heartdisease$cp)[levels(heartdisease$cp)==4] <- "ChestPaintype_4"

levels(heartdisease$slope)[levels(heartdisease$slope)==1] <- "Peak_Exercise_Slope_Upsloping"
levels(heartdisease$slope)[levels(heartdisease$slope)==2] <- "Peak_Exercise_Slope_Flat"
levels(heartdisease$slope)[levels(heartdisease$slope)==3] <- "Peak_Exercise_Slope_Downsloping"

levels(heartdisease$thal)[levels(heartdisease$thal)==3] <- "Thalium_ST_normal"
levels(heartdisease$thal)[levels(heartdisease$thal)==6] <- "Fixed_defect"
levels(heartdisease$thal)[levels(heartdisease$thal)==7] <- "Reversible_defect"

```

With regard to the parameter that we will try to predict, we add a **presence** boolean column: the binary heart disease presence column is built based on the target column and is a simplification of the observation. While the target takes integer values from 0 to 4 (0 being the absolute absence of disease), this new value presence will indicate presence (presence=1,2,3or4) or absence (presence=0).

```

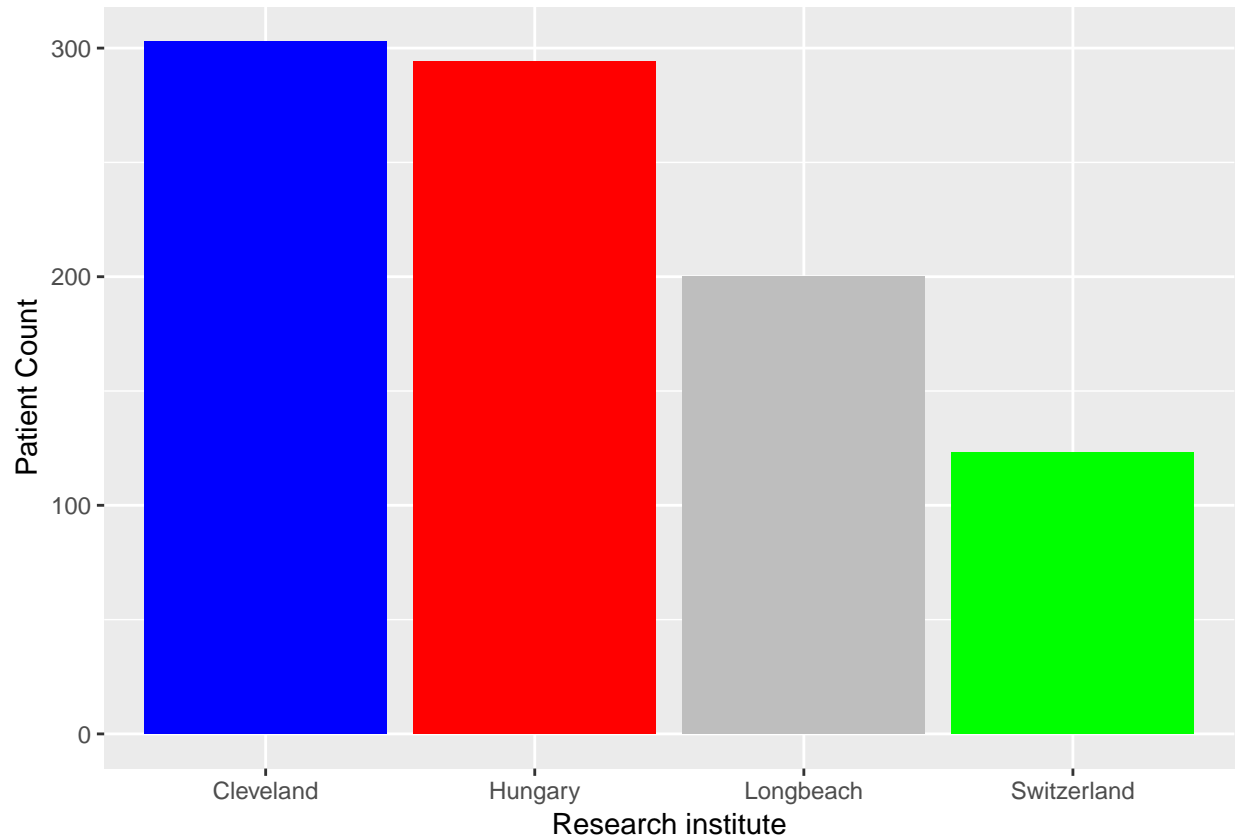
heartdisease <- mutate(heartdisease,presence=1*!(target==0))

# Adding levels for the heart disease presence column
heartdisease$presence<-as.factor(heartdisease$presence)
levels(heartdisease$presence)[levels(heartdisease$presence)==0] <- "Healthy"
levels(heartdisease$presence)[levels(heartdisease$presence)==1] <- "Heart Disease"

# Converting numerical data to numeric format
heartdisease$thalach<-as.numeric(heartdisease$thalach)
heartdisease$trestbps<-as.numeric(heartdisease$trestbps)
heartdisease$chol<-as.numeric(heartdisease$chol)
heartdisease$oldpeak<-as.numeric(heartdisease$oldpeak)

```

We can visualize the distribution of patients depending on the contributing research institute:



There is no predominant quantity of patients from a given institute, which is in favor of having merged all datasets together.

2.3 Building the training and validation

The total count of samples in the dataset published by UCI is:

```
nrow(heartdisease)
```

```
## [1] 920
```

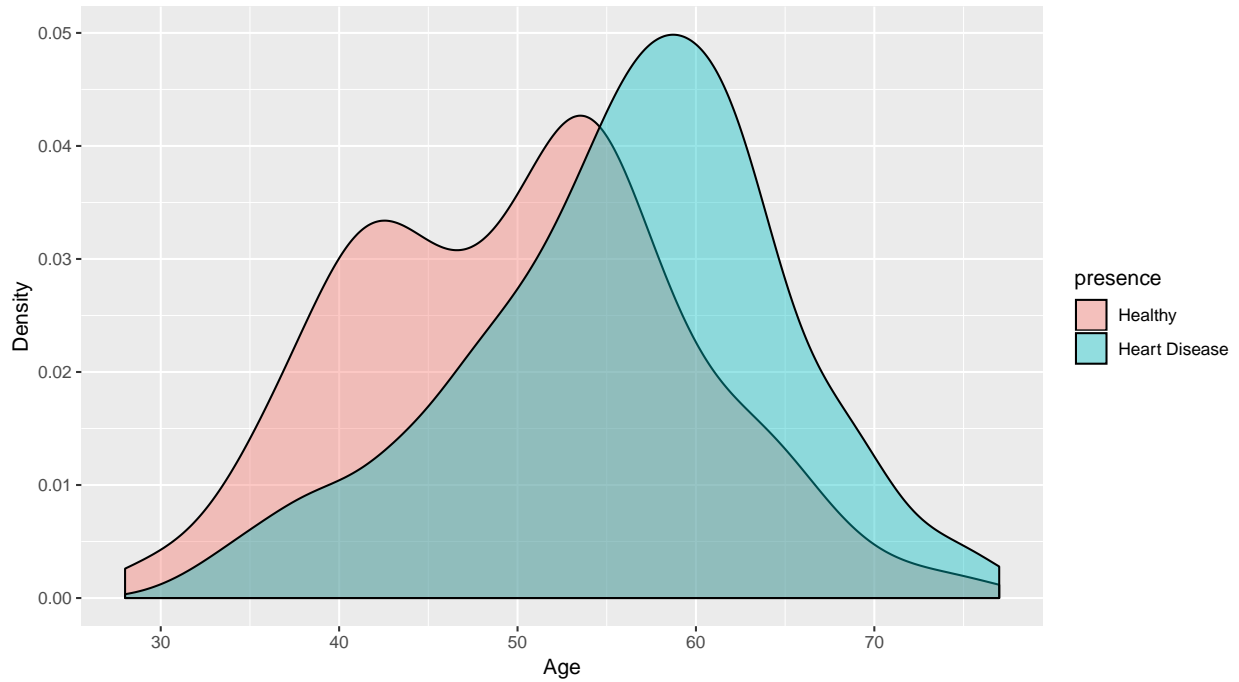
We will divide this data in 80% for the training set and 20% for the final validation. The feature exploration will then be carried out on the training set. Final performance evaluation will be carried out on the validation set.

```
set.seed(2810)
val_index <- createDataPartition(y=heartdisease$presence, times=1, p=0.2, list=FALSE)
hd_trainset <- heartdisease[-val_index,] #defining the heart disease training data set
hd_valset <- heartdisease[val_index,] #defining the heart disease validation data set
```

2.4 Feature exploration

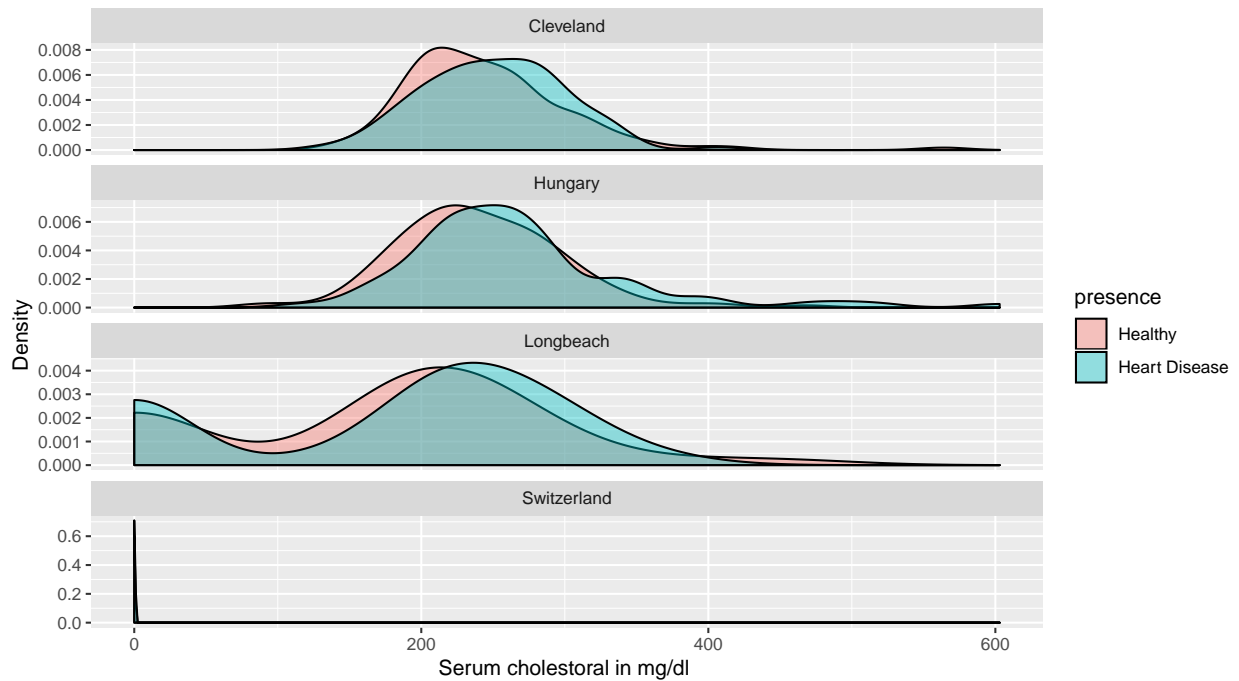
2.4.1 Continuous features

Age:



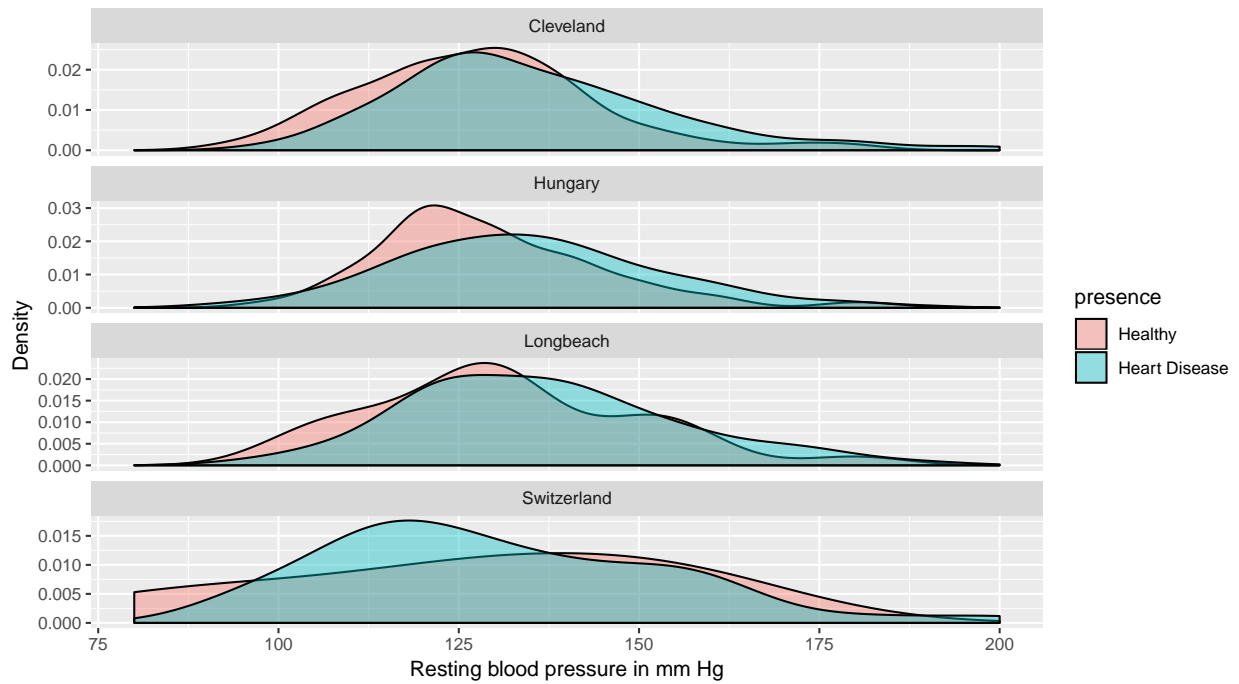
Age appears not to be a significant feature, as the healthy and heart disease distribute evenly over the ages.

Cholesterol:



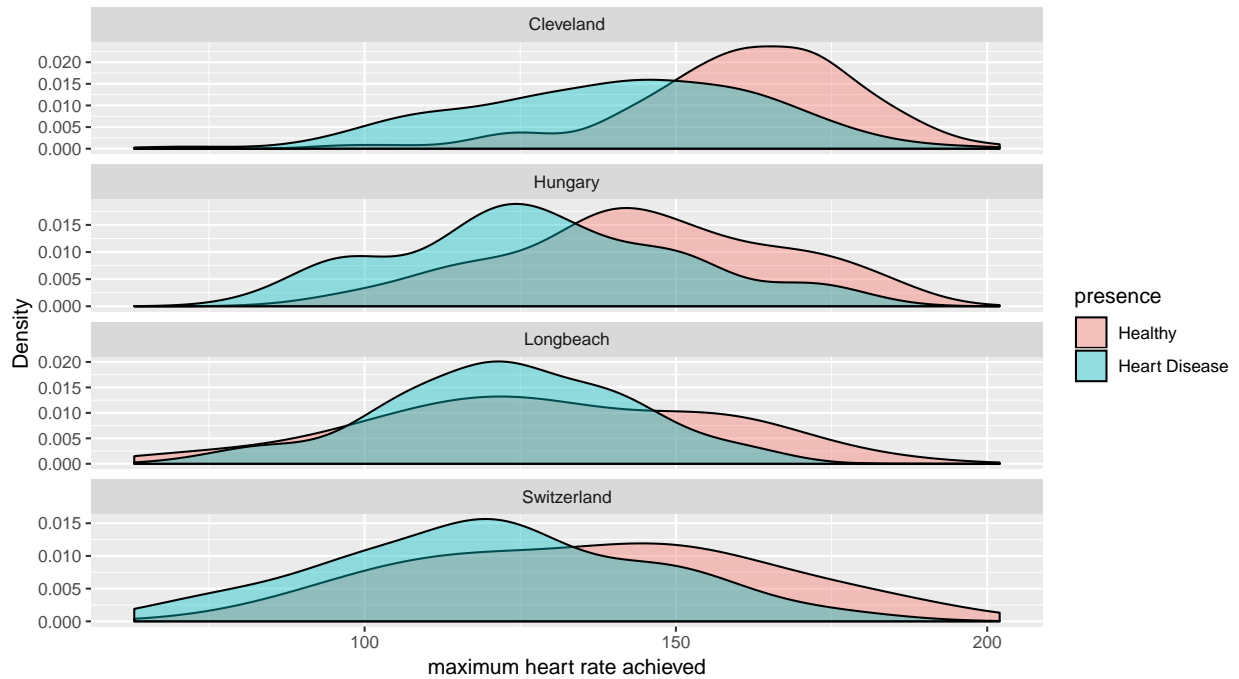
The cholesterol data reveals differences between research institutes, that could be probably explained by input errors. Regardless, the serum cholesterol does not appear to be significantly different between healthy and people suffering from heart disease. This feature will be discarded.

Trestbps Resting blood pressure (in mm Hg on admission to the hospital):



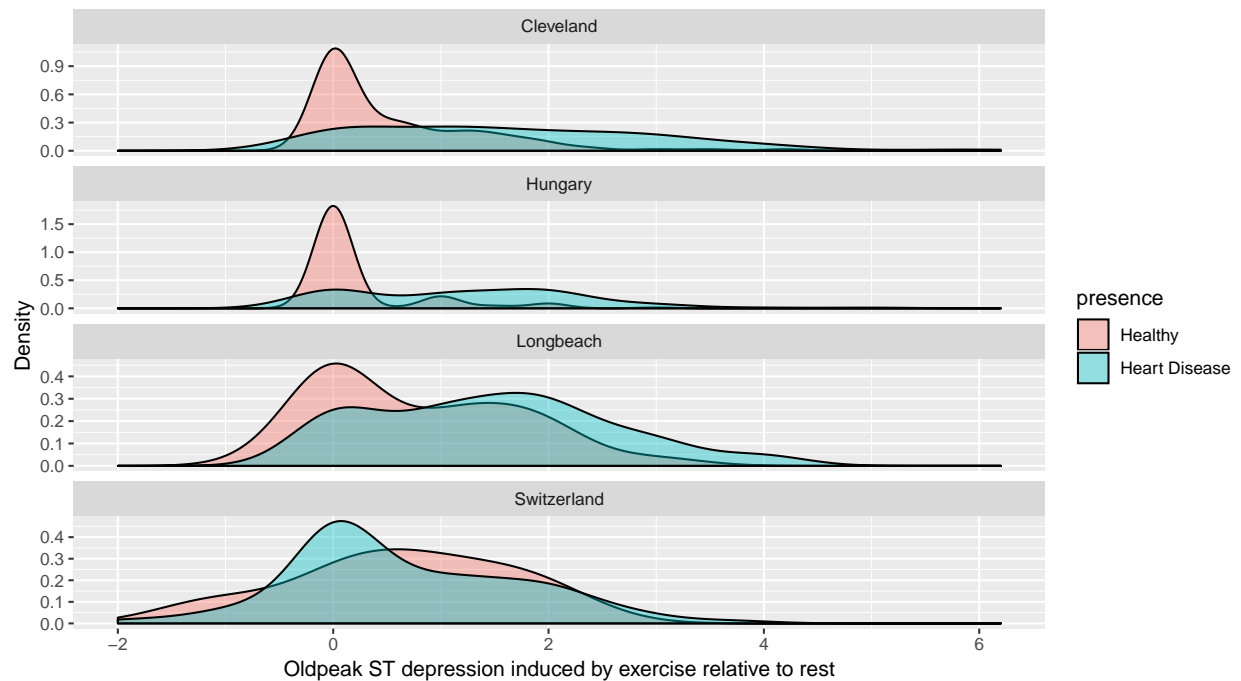
Resting blood pressure appears not to be a significant feature, as the healthy and heart disease distribute evenly over the pressure range.

Thalach maximum heart rate achieved :



The Thalach maximum heart rate reveals some differences, as the healthy/Heart disease populations are shifted from one another. This observation is made for each research institute. This feature is therefore of interest and will be used in the fitting formula.

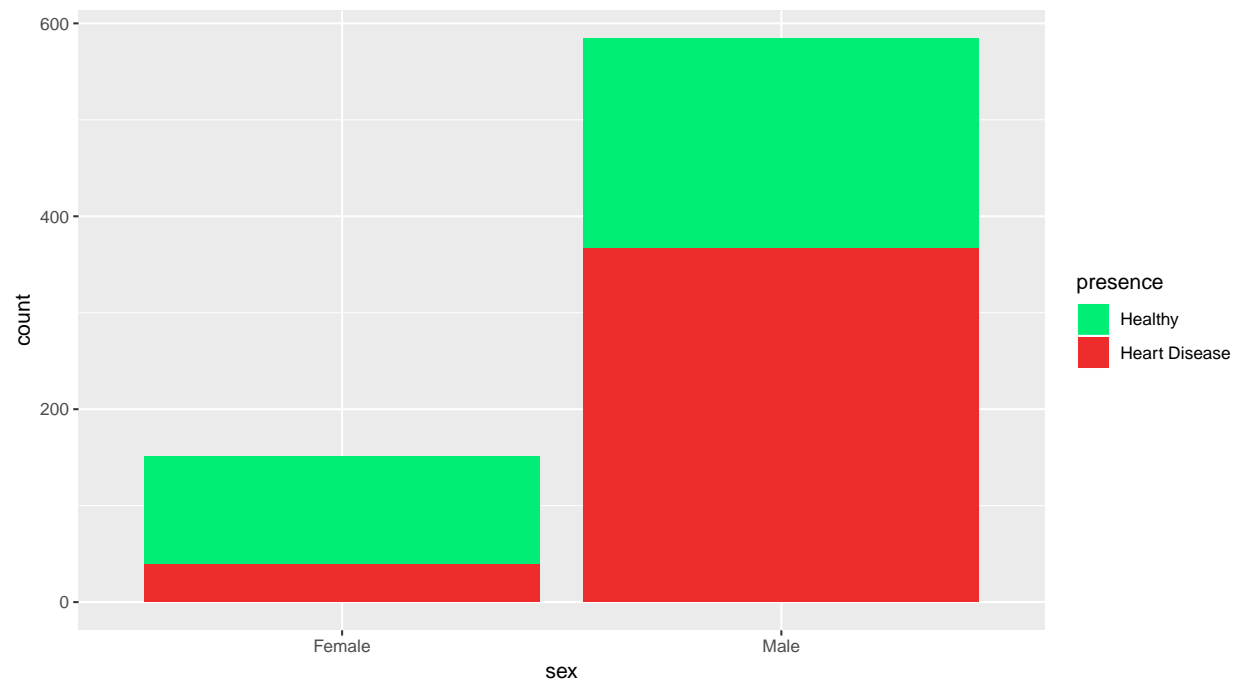
Oldpeak ST depression induced by exercise relative to rest:



The Oldpeak ST depression induced by exercise relative to rest seems not to be a differentiating factor in the data, based on the distributions plotted.

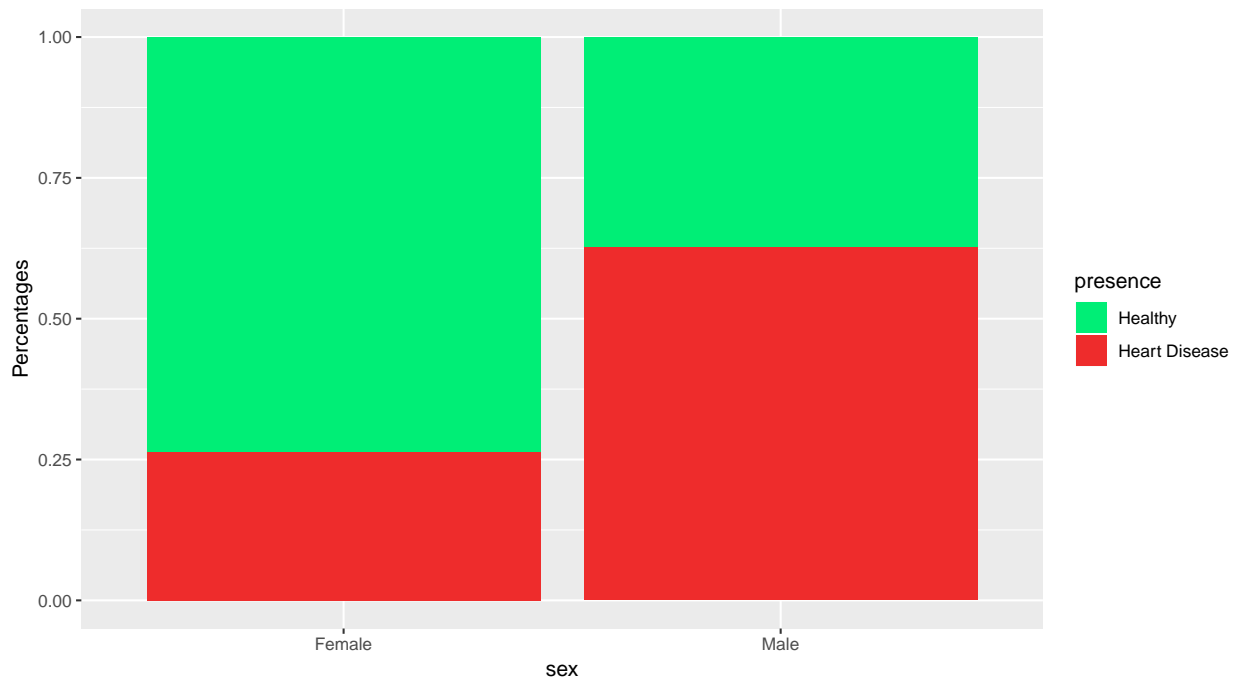
2.4.2 Categorical features

Patient sex:



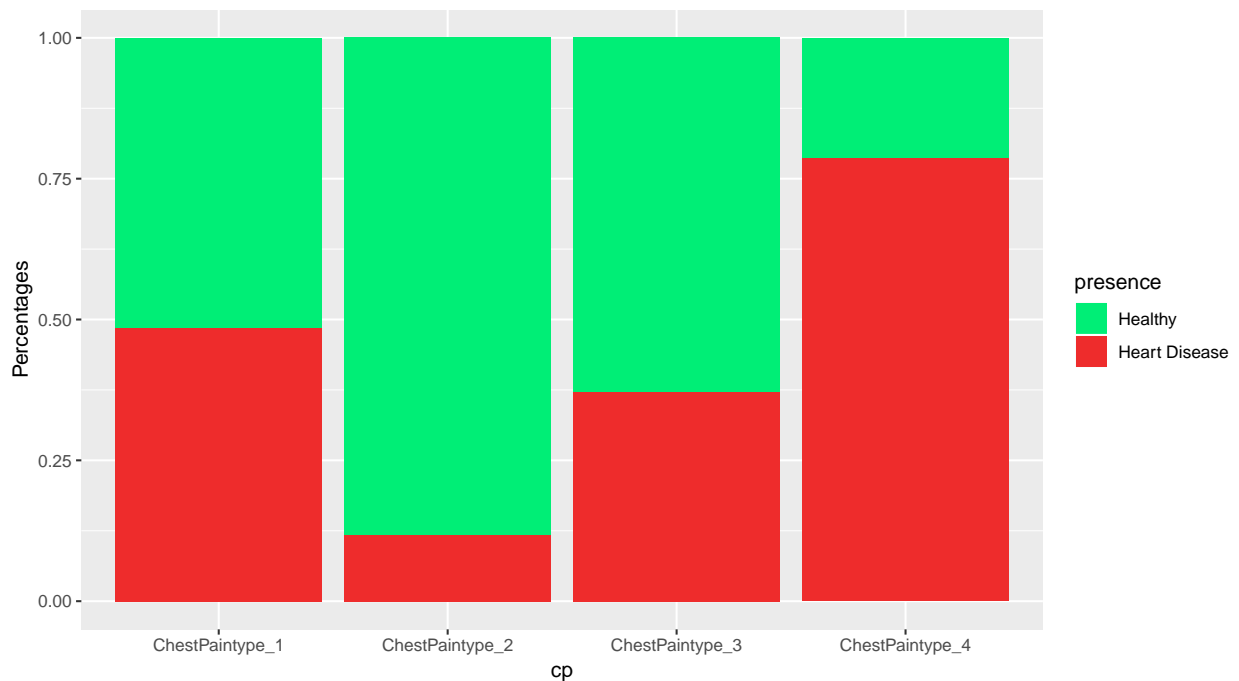
The visualisation tends to show that heart diseases affect more men than women. Modifying the plotting

script to show percentages confirms this observation.



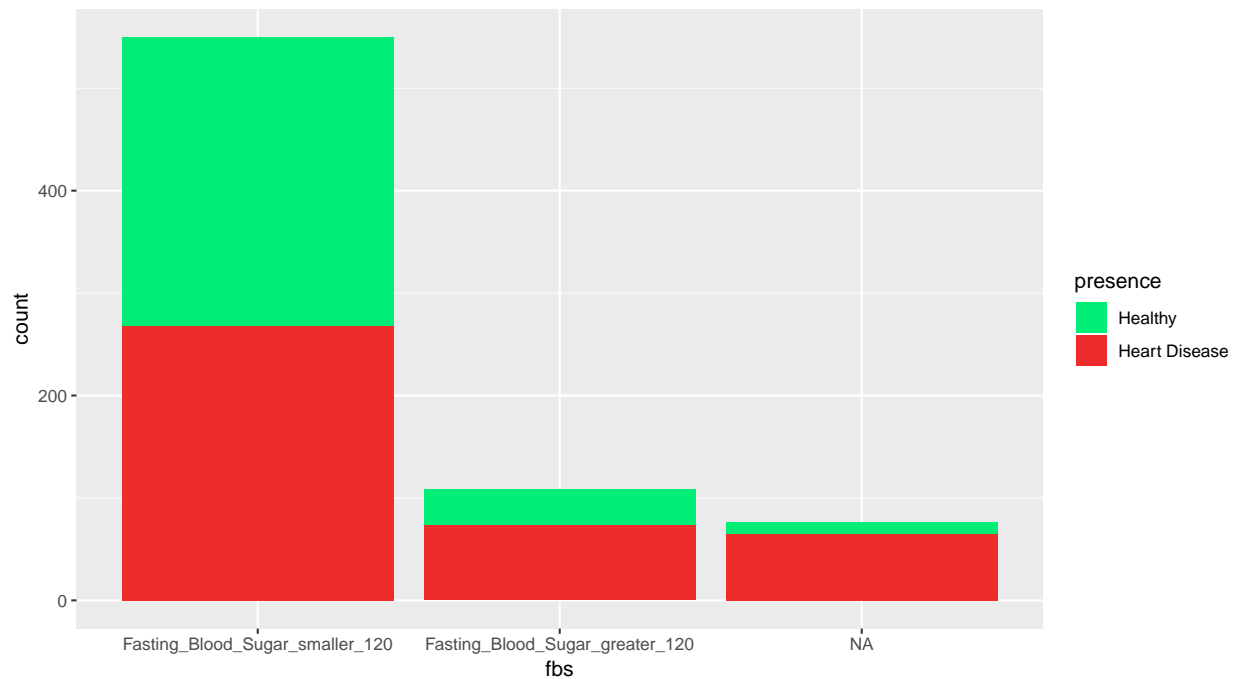
The patient sex is a feature that will be kept in the fitting process, as it appears to be a determining factor.

Chest pain type:

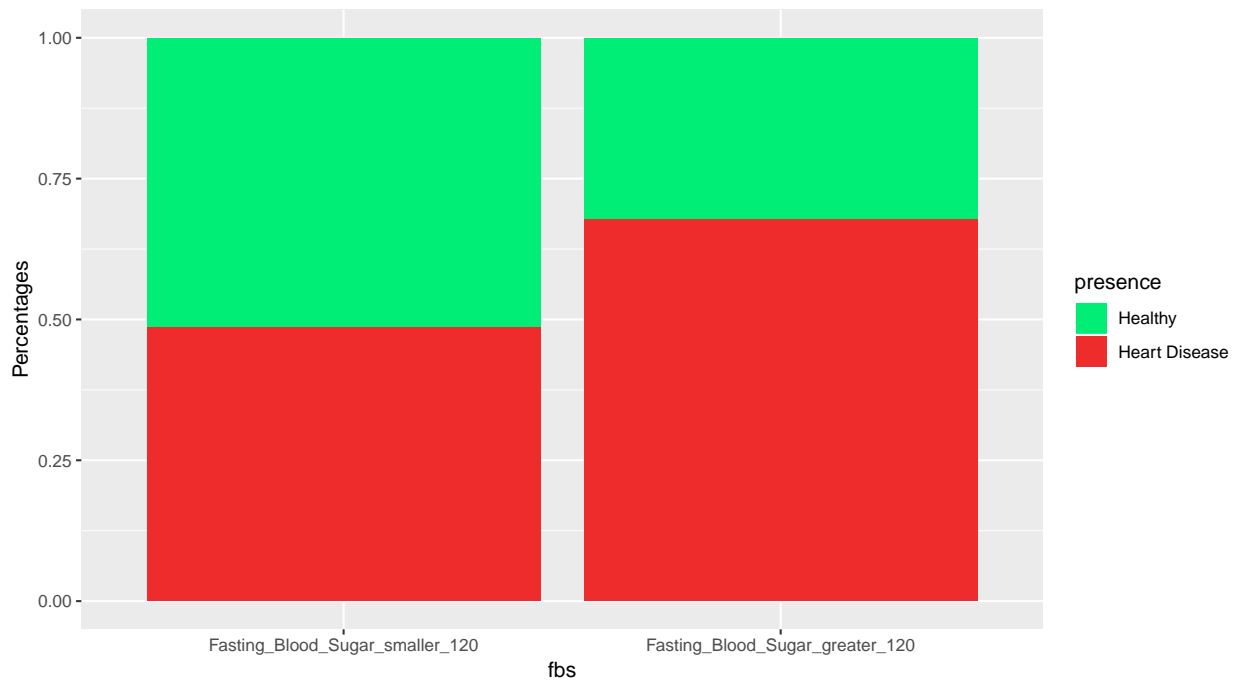


The type of chest pain is an interesting parameter, as the data reveals. Patient declaring a chest pain type of 4 are more likely to suffer from heart disease.

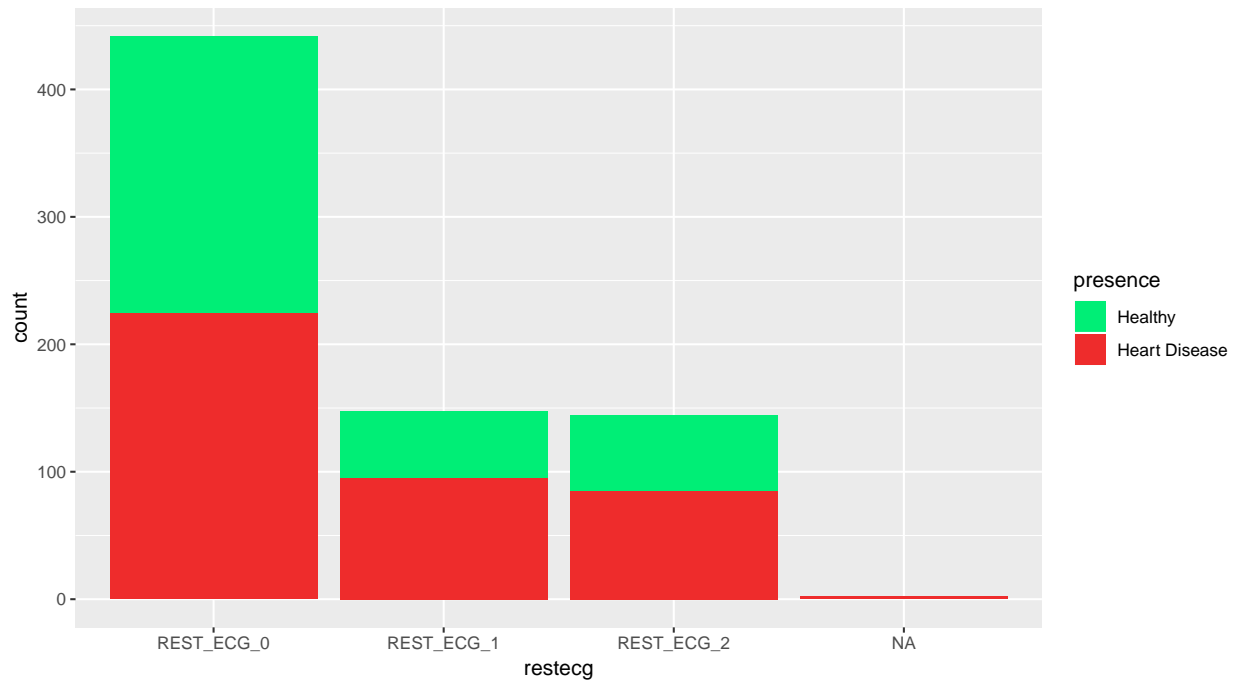
****Fast blood sugar:****



This parameter is a good example of disparity between research institutes, where some of the them have not recorded this parameter. Nevertheless, when discarding the values that are unavailable, and looking at percentages rather than absolute values, we can see that Fast Blood Sugar is an important feature.

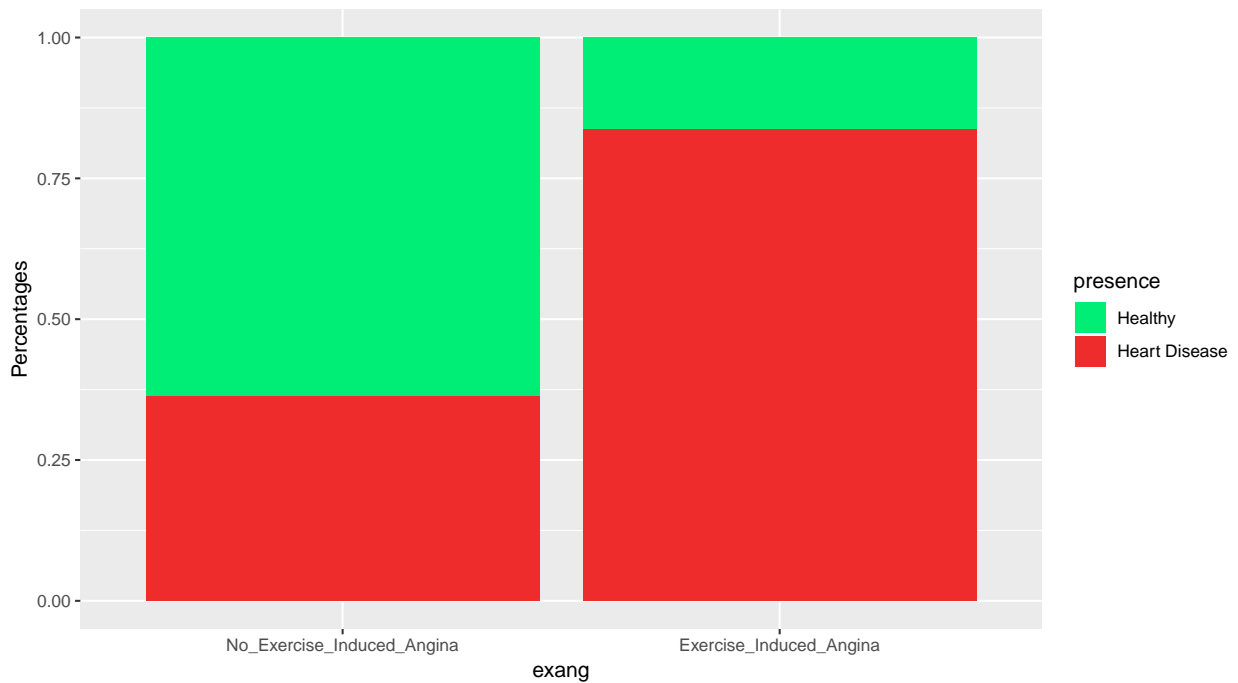


****Rest ECG:****



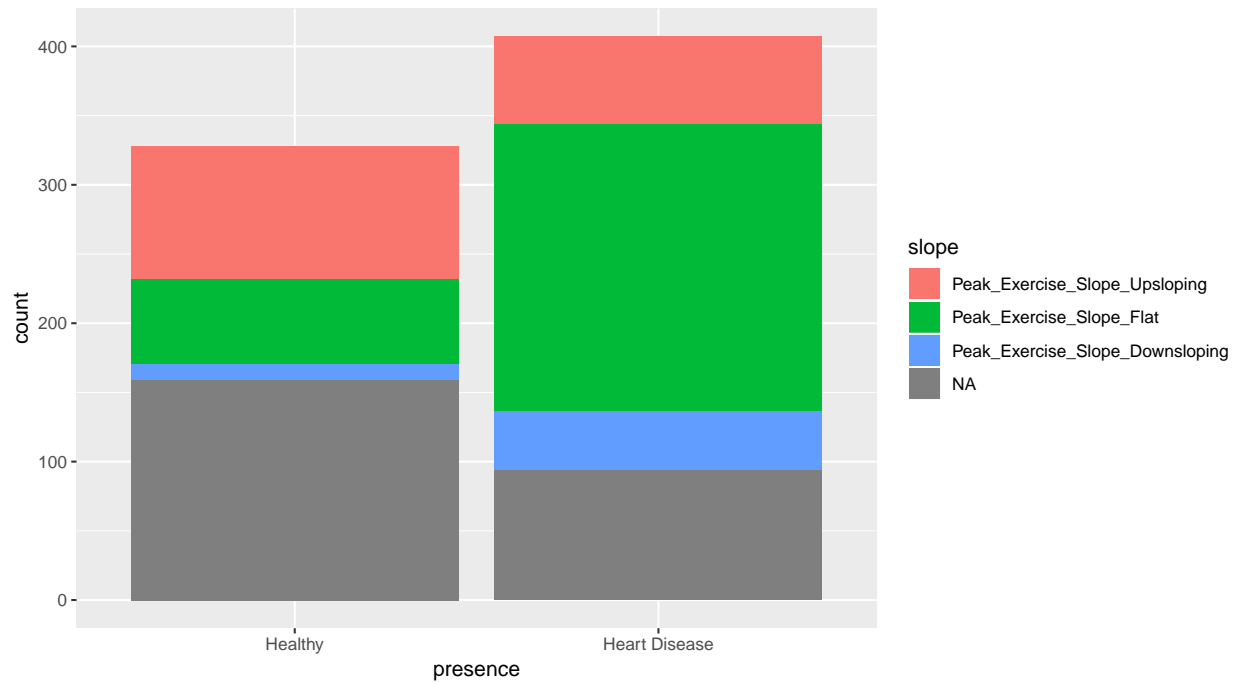
Rest ECG does not appear to be a determining feature, given the equirepartition of no disease/disease among the different levels.

Exercise induced angina:



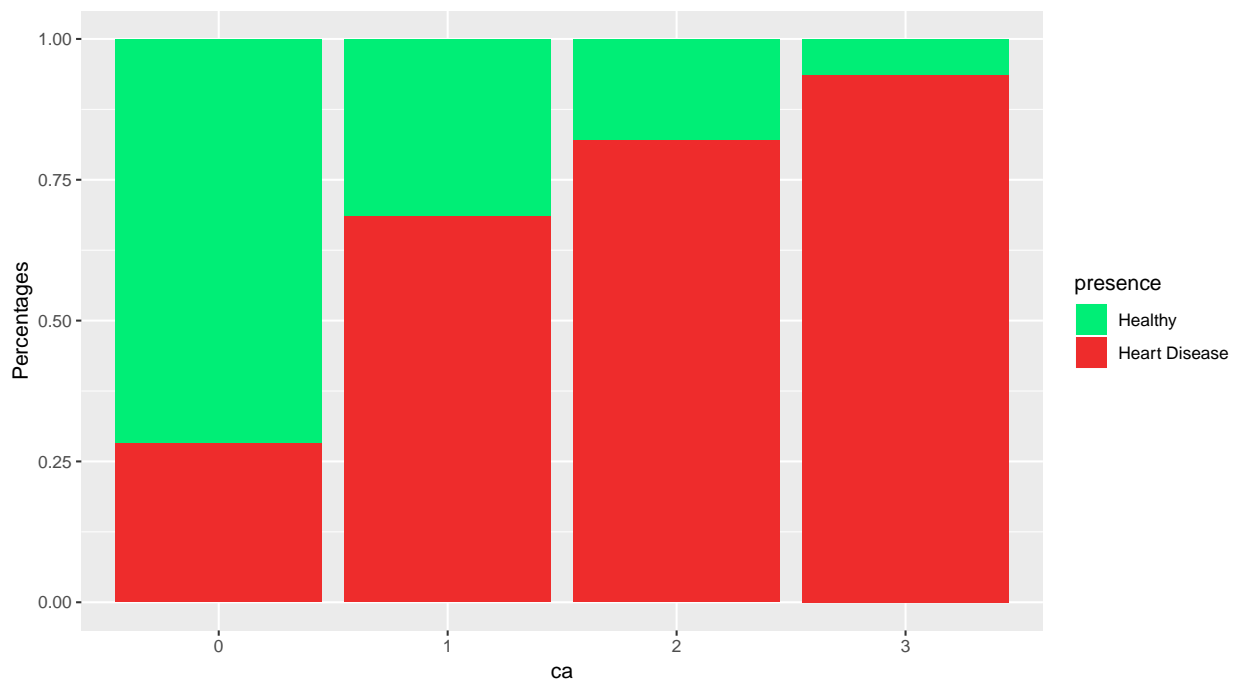
The patients affected by *Exercise induced angina* have a higher likelihood of having a heart disease. We will keep this parameter in the features used for the learning process.

The slope of the peak exercise ST segment:



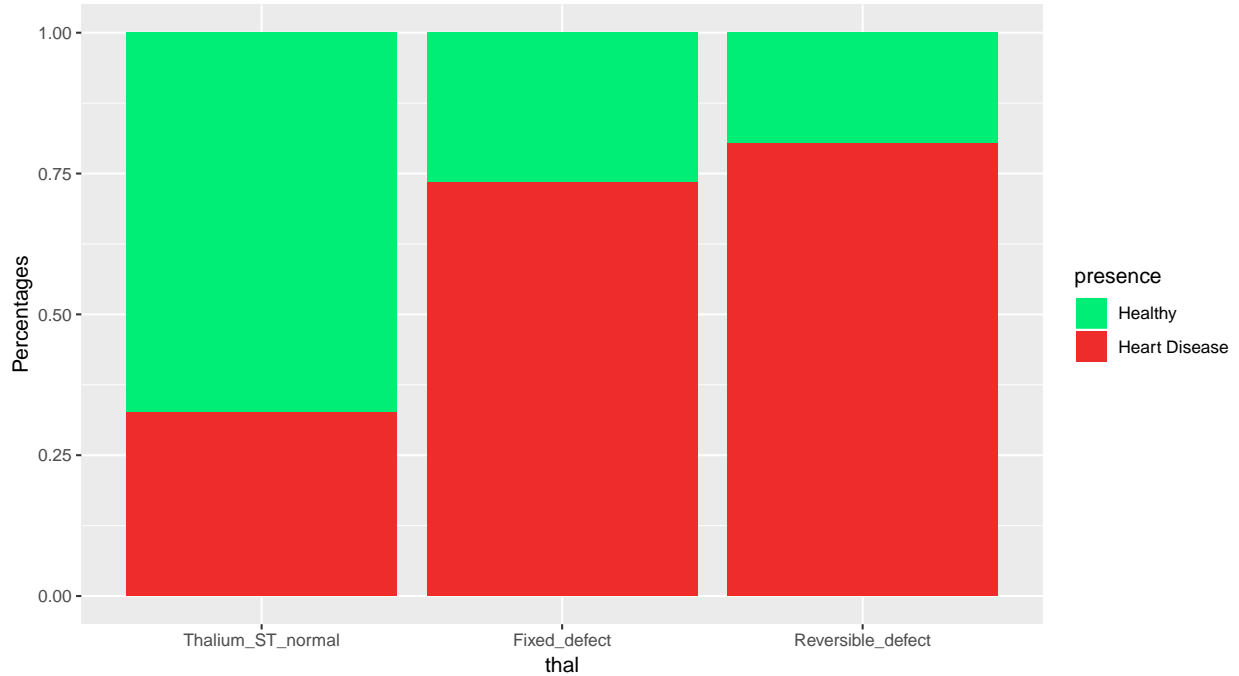
A flat peak exercise slope appears to be an important factor in the determination of a heart disease.

Number of major vessels (0-3) colored by flourosopy:



The plot reveals provides with a significant finding. The higher the number of major vessels colored by fluoroscopy is, the higher is the prevalence of heart disease affected patients. This parameter is definitely a feature to be considered in the analysis.

Thalium stress test result:



The result of the thalium stress test is also a determining feature. The detection of a defect points to a higher proportion of heart disease affected people.

2.5 Conclusion - features selected for the learning process

Based on the data analysis and visualisation provided above, the features that will be kept for the learning process are (in no particular order):

1. **sex**: The patient sex
2. **cp**: Chest pain type (from 1 to 4, in 1 increments)
3. **fbs**: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
4. **exang**: exercise induced angina
5. **slope**: the slope of the peak exercise ST segment
6. **ca**: number of major vessels (0-3) colored by fluoroscopy
7. **thal**: Thallium stress test result : 3 = normal; 6 = fixed defect; 7 = reversible defect

2.6 Strategy and methods

The approach will be to test four different methods and pick the one providing the best performance. The methods used will be: * GLM: Generalized Linear Model * KNN: K-Nearest Neighbors * RF: random forest * RT: regression trees

With regard to performance, we will predominantly consider the receiver operating characteristic AUC (area under the curve). But accuracy will also be looked at, with the selection of decision threshold (the methods output probabilities in the prediction and it can be interesting to explore setting the decision threshold to a different value than the immediate and typical 0.5).

In a second step, we will explore further machine learning techniques to fit the data. The objective is to see if other methods are able to further improve the results already achieved.

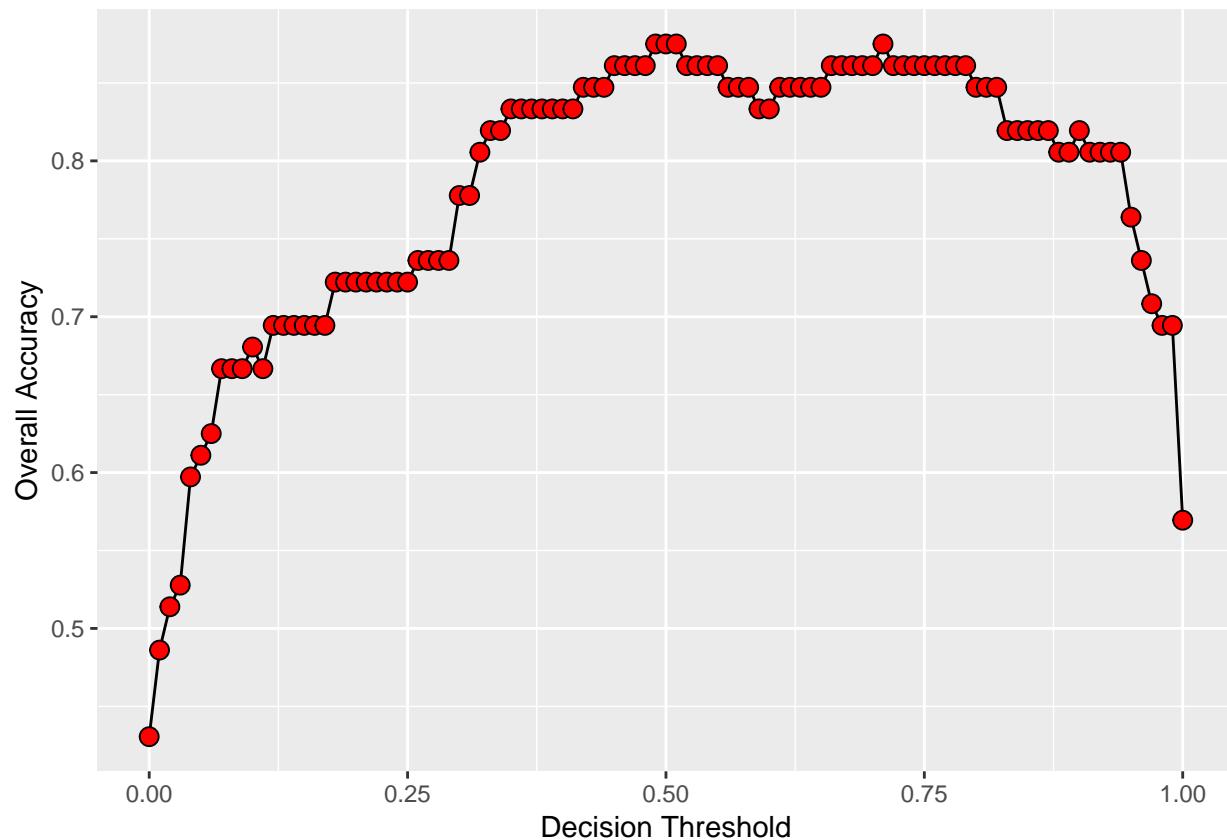
All methods will be run with the same training control. The cross validation, with a fitting sampled set representing 90% of the training set. Cross validation will be done on 10 resampling iterations.

2.7 Models evaluation

2.7.1 Generalized Linear Model

```
control_glm <- trainControl(method = "cv", number = 10, p = .9)
train_glm <- train(presence ~ sex+cp+fbs+exang+slope+ca+thal, method = "glm",
  family=binomial,
  data = hd_trainset, trControl = control_glm ,
  na.action = na.omit)
glm_results <- predict(train_glm, hd_valset, type = "prob")
```

It can be interesting to see how accuracy evolves depending on the threshold chosen to separate positive (heart disease) from negative (healthy) populations. The data confirms that the traditional value of 0.5 is a good choice:



2.7.2 K Nearest Neighbours

```
control_knn <- trainControl(method = "cv", number = 10, p = .9)
train_knn <- train(presence ~ sex+cp+fbs+exang+slope+ca+thal,
  method = "knn",
  data = hd_trainset,
  trControl = control_knn ,
  tuneGrid=data.frame(k=seq(1,20,1)),
  na.action = na.omit)
knn_results <- predict(train_knn, hd_valset, type = "prob")
```

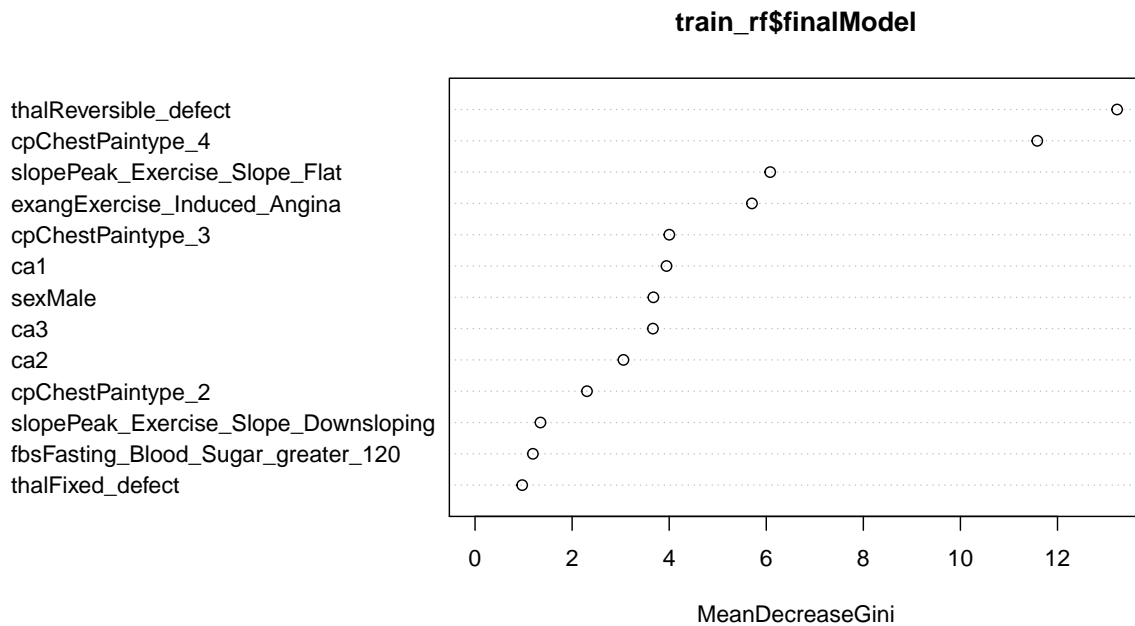
2.7.3 Random Forest

```
control_rf <- trainControl(method = "cv", number = 10, p = .9)
train_rf <- train(presence ~ sex+cp+fbs+exang+slope+ca+thal ,
                 method = "rf",
                 data = hd_trainset,
                 na.action = na.omit,
                 trControl=control_rf)

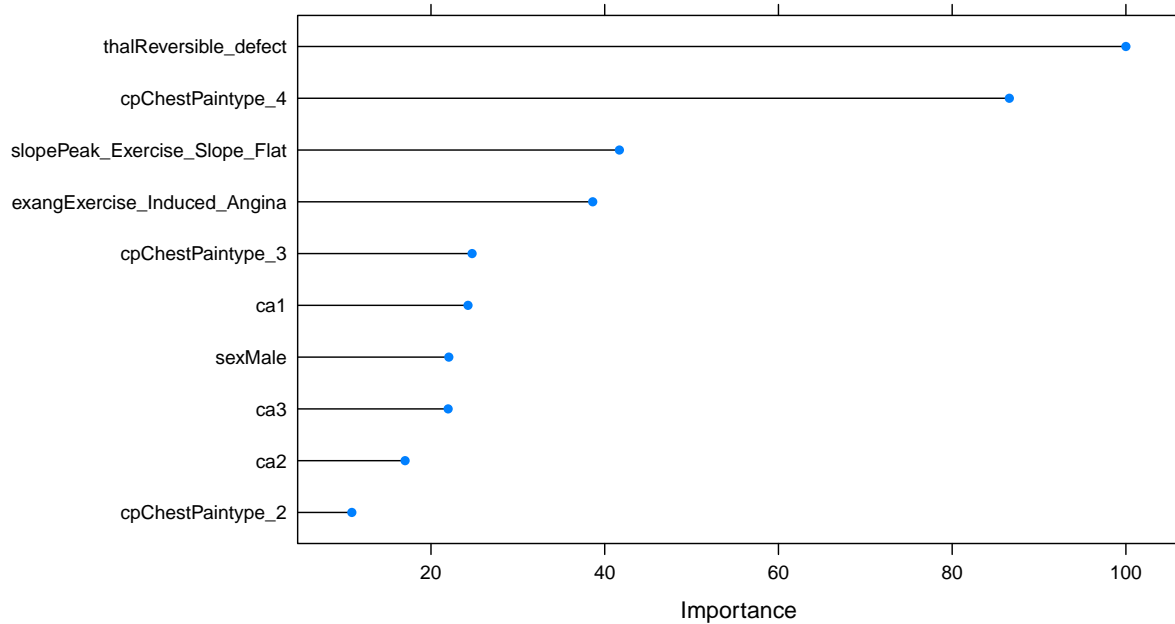
# predicting
rf_results<-predict(train_rf,hd_valset,type ="prob")
```

The variable importance function provides interesting insights in the significant features that predominantly determine the presence/absence of disease:

```
varImpPlot(train_rf$finalModel)
```



```
# plotting the most important features and associated value, in decreasing importance
plot(varImp(train_rf), top = 10)
```

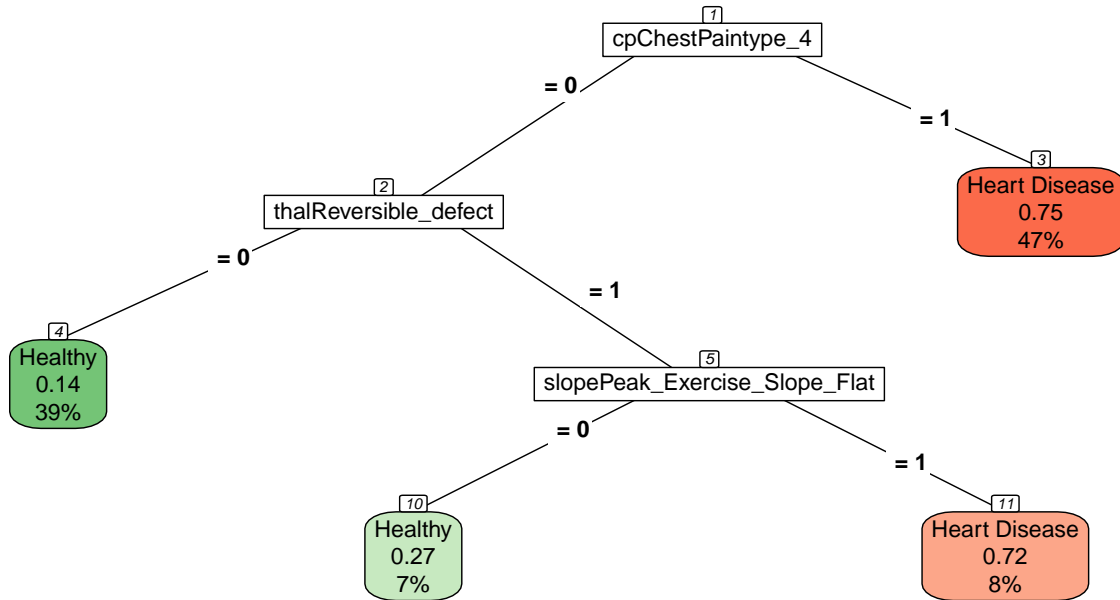
The plots show that Chest Pain (type 4) and a Thallium Stress test with reversible defect are the most important factors in the prediction of heart disease. This is in line with our initial assumptions, when selecting the training parameters.

2.7.4 Regression Trees

```
control_rt <- trainControl(method = "cv", number = 10, p = .9)
train_rt <- train(presence ~ sex+cp+fbs+exang+slope+ca+thal ,
  method = "rpart",
  data = hd_trainset,
  trControl = control_rt,
  na.action = na.omit)
rt_results<-predict(train_rt,hd_valset,type ="prob")
```

The resulting decision tree can be drawn using the rpart.plot package:

```
rpart.plot(train_rt$finalModel,
  type=5,
  fallen.leaves = FALSE,
  box.palette = "GnRd",
  nn=TRUE)
```



2.8 Comparison of methods

The ROCR package can be used to compare these four methods. In particular, ROR AUC and accuracy can be compared.

Providing following ROC AUCs:

Method	ROCAUC
glm	0.9067663
knn	0.9130606
rf	0.8992919
rt	0.7773407

And with the following accuracies:

Method	Accuracy
glm	0.8750000
knn	0.8472222
rf	0.8611111
rt	0.7638889

The accuracy provided by the regression tree algorithm is much lower than the other approaches. While the KNN methods yields a better ROC AUC, the accuracy of the GLM method is much better for a ROC AUC that is not significantly smaller. The preferred method is therefore GLM. The next section explores additional methods to evaluate if the GLM results can be further improved.

2.9 Broader Model evaluation

In this section, we explore a group of learning methods to assess how GLM's performance compares. Using the `lapply()` function, it is possible to run the `train` function on a list of methods:

- `glm`: generalized linear model
- `lda`: Linear Discriminant Analysis
- `gamLoess`, `gam`: generalized additive models
- `qda`: Quadratic Discriminant Analysis
- `knn`: K-nearest neighbours
- `kknn`: Weighted k-Nearest Neighbors
- `rf`: Random Forerst
- `svmRadial`, `svmRadialCost`, `svmRadialSigma` : Radial Support Vector Machines
- `svmLinear`: Linear Support Vector Machines
- `wsrf`: Weighted Subspace Random Forest
- `Rborist`: Random forest
- `avNNet`: Neural Networks
- `mlp`: multilayer perceptron
- `monmlp`: Multi-Layer Perceptron Neural Network with Optional Monotonicity
- `adaboost`: Adaboost Algorithm
- `gbm`: Generalized Boosted Regression
- naive bayes method

For the purpose of consistency, the same train control parameters will be used.

3 Final Results

The result from the training can be summarized in the following table which provides the method's accuracy over the training data set:

Method	Accuracy over training set
mlp	0.8411
avNNet	0.8320
svmRadialCost	0.8288
gam	0.8229
adaboost	0.8202
wsrf	0.8192
lda	0.8188
glm	0.8186
gamLoess	0.8154
rf	0.8154
svmRadial	0.8153
monmlp	0.8146
gbm	0.8146
naive_bayes	0.8111
svmRadialSigma	0.8065
svmLinear	0.8059
knn	0.7968
qda	0.7789
kknn	0.7662
Rborist	0.5243

Here are the achieved accuracies over the validation test set:

With regard to the training set, it is interesting to see that a number of methods provide similar results, with our selected method GLM not scoring the highest figure. The Rborist method does not provide good results at all.

Method	Accuracy over validation set
glm	0.8750000
gamLoess	0.8750000
gam	0.8750000
lda	0.8611111
svmRadialSigma	0.8611111
gbm	0.8611111
rf	0.8472222
svmRadialCost	0.8472222
svmRadial	0.8333333
svmLinear	0.8333333
naive_bayes	0.8333333
qda	0.8194444
knn	0.8194444
mlp	0.8194444
wsrf	0.8055556
adaboost	0.8055556
avNNet	0.7638889
monmlp	0.7638889
kknn	0.7500000
Rborist	0.5694444

When the models are run over the validation data set, we see that the gamLoess and gam method provide similar results to the GLM approach (however it should be noted that these algorithms took more time to run).

This exploration confirms that the GLM method is well suited for this data set experiment.

3.1 An exploration: approaching the missing data

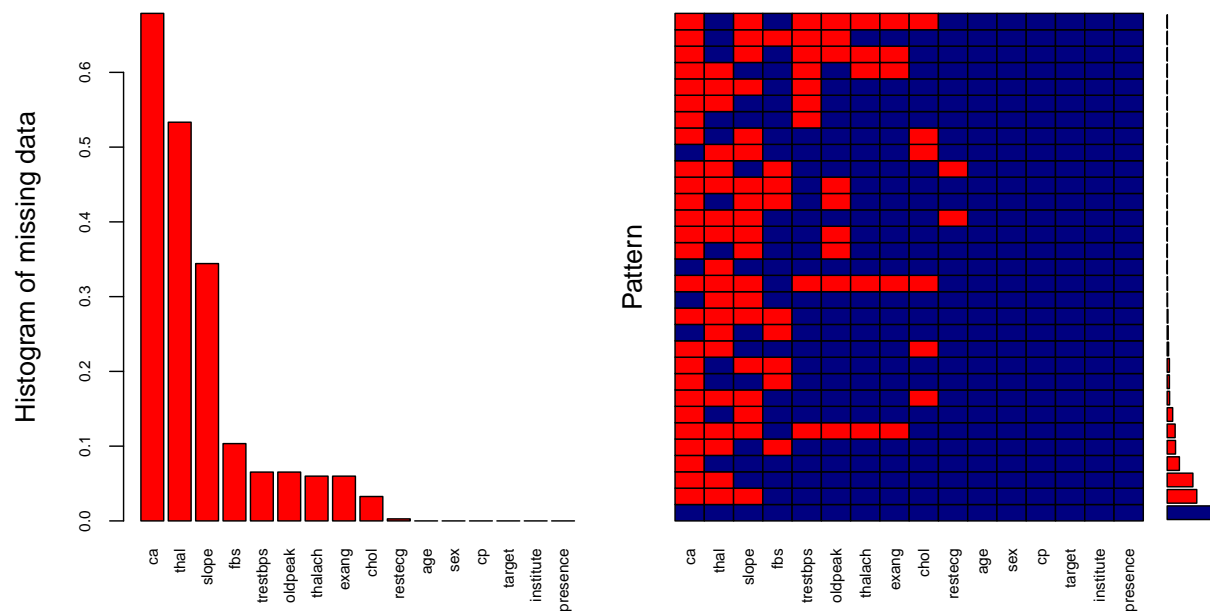
Up to now, we have been careful to discard from the training and validation data, all rows from the Heart Disease dataset that contained “NAs”. In this section, we will explore a data imputation method and assess the obtained accuracy. The MICE package will be used to complete the datasets. For the purpose of this exploration, only the selected method (GLM) will be used to train the model with the imputed data.

A reasonable assumption in using the MICE package is to consider the data is missing at random.

3.1.1 Visualisation of the missing data

Before training the model, it is possible to visualize the missing data, providing insights on which features in the datasets are particularly scarce. We will use a function provided by the VIM package.

```
aggr_plot <- aggr(hd_trainset, col=c('navyblue','red'), numbers=TRUE,
                 sortVars=TRUE, labels=names(hd_trainset), cex.axis=.7,
                 gap=3, ylab=c("Histogram of missing data","Pattern"))
```



3.1.2 Imputing the data and training the GLM model

```
## Imputing the missing data
impResult<-mice(data = hd_trainset, m = 5, method = "pmm", maxit = 50, seed = 500)
completedTrainData <- complete(impResult,1)

## Training the model
control_train <- trainControl(method = "cv", number = 10, p = .9)
models <- c("glm")

fits <- lapply(models, function(model){
  print(model)
  train(presence ~ sex+cp+fbs+exang+slope+ca+thal, method = model, data = completedTrainData,
        trControl = control_train)
})
names(fits) <- models
```

3.1.3 Completing the validation data set and predicting the disease presence

Now that the model is trained, the validation data set needs first to be completed as well, as this data sets also misses data. We will use the same approach as for the training data set, relying on the MICE data set with the same parameters.

```
impResult<-mice(data = hd_valset, m = 5, method = "pmm", maxit = 50, seed = 500)
completedValData <- complete(impResult,1)
```

The final prediction can then be carried out and accuracy calculated:

```
pred <- sapply(fits, function(object)
  predict(object, newdata = completedValData))
dim(pred)
```

```
acc <- colMeans(pred == hd_valset$presence)
```

This provides an accuracy of 0.854 over the validation test set.

This exploration is interesting and shows that data imputing has degraded to the overall accuracy of the GLM model, but not to an extent that would render this approach unacceptable. Further investigations could be to tune the imputing method (we used the **Predictive mean matching** univariate imputation method). A drawback of data imputation in this case is that it requires to be carried out on the validation data as well before predicting the presence or absence of heart disease.

4 Conclusion

After an exploratory data analysis, a number of features in the data set were determined to be significant to predict the presence/absence of heart disease over patients. A variety of methods were investigated and the method selected was the Generalized Linear Model, achieving a prediction accuracy of about **87.5%** over the validation data set.

The exploratory data analysis revealed a large number of incomplete patient features. A further investigation was carried out to explore ways to handle the missing data. The imputation method used, in conjunction with the selected GLM method, led to achieve a prediction accuracy of about 85%.

5 References and Credits

5.1 References

- **Data Analysis and Prediction Algorithms with R**, Rafael A. Irizarry, <https://rafalab.github.io/dsbook/>
- **Missing Data Analysis with mice**, Firouzeh Noghrehchi, <https://web.maths.unsw.edu.au/~dwarton/missingDataLab.html>

5.2 Credits

The authors of the databases used for this project have requested that any publications resulting from the use of the data include the names of the principal investigator responsible for the data collection at each institution. They would be:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.