Cardiac Arrhythmia DataScience Project

Data Analysis and Prediction Algorithms with R

Alejandro A Valenzuela

2021-07-11

Overview/Executive Summary

In this section we describe the dataset and summarizes the goal of the project and key steps that were performed.

This project is part of the Capstone of the Professional Certificate Program of Data Science and it has as scope applying machine learning techniques that go beyond standard linear regression, giving the opportunity to use a publicly available dataset to solve the problem of your choice. The UCI Machine Learning Repository and Kaggle are good places to seek out a dataset.

For this project, the Cardiac Arrhythmia dataset from UCI Machine Learning Repository was selected. As we can guess this is a life/health dataset and as the web page mention, this dataset has the following main information:

- This database contains 279 attributes, 206 of which are linear valued and the rest are nominal.
- Concerning the study of H. Altay Guvenir: "The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. Class 01 refers to 'normal' ECG classes 02 to 15 refers to different classes of arrhythmia and class 16 refers to the rest of unclassified ones. For the time being, there exists a computer program that makes such a classification. However there are differences between the cardiolog's and the programs classification. Taking the cardiolog's as a gold standard we aim to minimise this difference by means of machine learning tools."

Characteristics of dataset

The Arrhythmia dataset is a table witch each row it is collection of health measurement for one patience and the last column says if this persons it is a normal health or has some of the 15 arrhythmia.

```
# dimension
dim(arrhythmia.uci)
```

```
## [1] 452 280
```

```
# Display predictors (columns)
str(arrhythmia.uci, list.len = 25)
```

```
## 'data.frame':    452 obs. of  280 variables:
##  $ V1        : int  75 56 54 55 75 13 40 49 44 50 ...
##  $ V2        : int  0 1 0 0 0 0 1 1 0 1 ...
##  $ V3        : int  190 165 172 175 190 169 160 162 168 167 ...
##  $ V4        : int  80 64 95 94 80 51 52 54 56 67 ...
##  $ V5        : int  91 81 138 100 88 100 77 78 84 89 ...
##  $ V6        : int  193 174 163 202 181 167 129 0 118 130 ...
##  $ V7        : int  371 401 386 380 360 321 377 376 354 383 ...
##  $ V8        : int  174 149 185 179 177 174 133 157 160 156 ...
##  $ V9        : int  121 39 102 143 103 91 77 70 63 73 ...
##  $ V10       : int  -16 25 96 28 -16 107 77 67 61 85 ...
##  $ V11       : chr  "13" "37" "34" "11" ...
##  $ V12       : chr  "64" "-17" "70" "-5" ...
##  $ V13       : chr  "-2" "31" "66" "20" ...
```

```
##  $ V14       : chr  NA NA "23" NA ...
##  $ V15       : chr  "63" "53" "75" "71" ...
##  $ V16       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V17       : int  52 48 40 72 48 36 44 44 40 44 ...
##  $ V18       : int  44 0 80 20 40 48 0 36 0 40 ...
##  $ V19       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V20       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V21       : int  32 24 24 48 28 20 24 24 20 28 ...
##  $ V22       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V23       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V24       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V25       : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

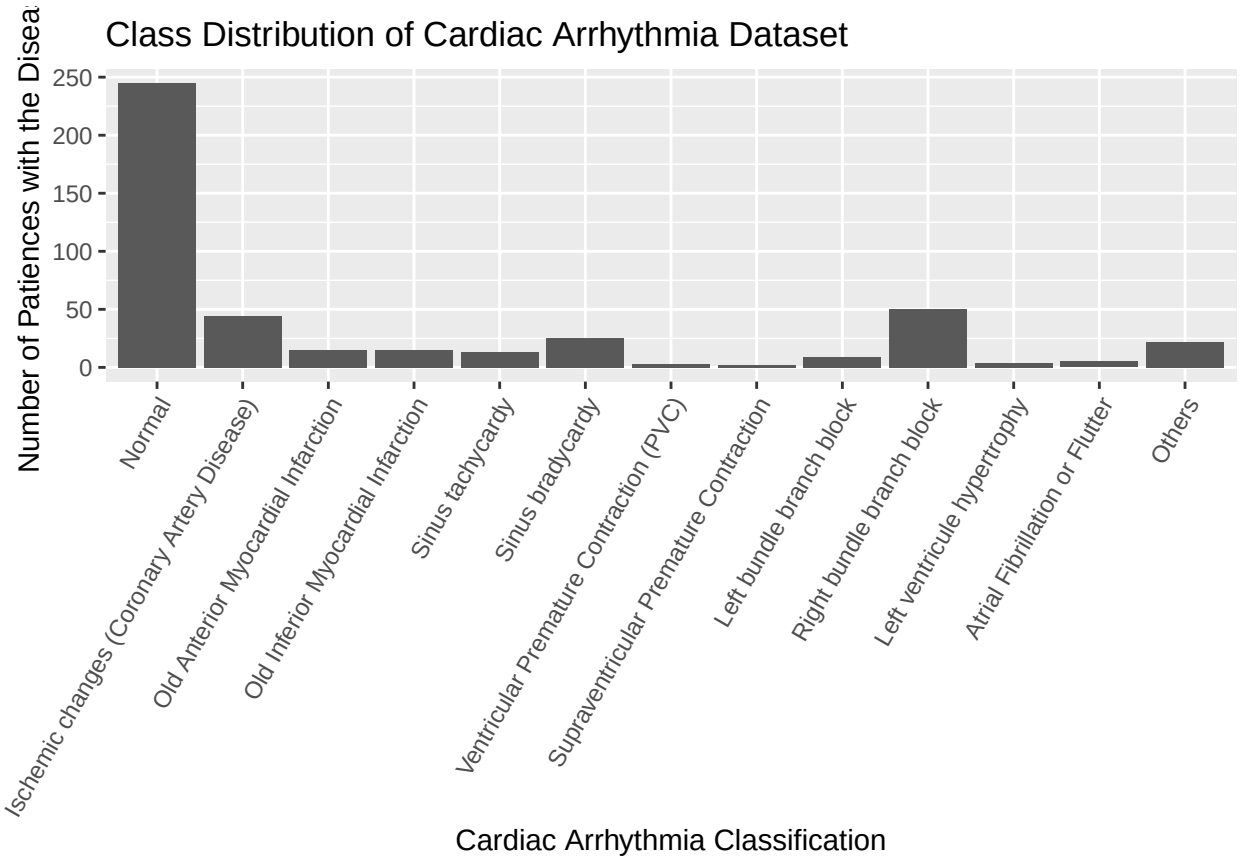
You can see there are few patients but a lot of elements to consider. This elements are displayed as the first view we can see that required some work in setting its proper name (provided in the description of the dataset) but we are not to cover each one because we are talking of 279 predictors.

About the output, the last column give us the Class code. 1 is for "Normal" and all the rest are arrhythmia. First we are going to display de distribution in a table, where you can see that "Normal" take a preponderance percentage and later a graph where we can see how the different arrhythmia are distributed in our dataset

Table 1: Presence of codes in our dataset

Class code	Class name	N Ocurrences	Percentage
1	Normal	245	54.2
2	Ischemic changes (Coronary Artery Disease)	44	9.7
3	Old Anterior Myocardial Infarction	15	3.3
4	Old Inferior Myocardial Infarction	15	3.3
5	Sinus tachycardy	13	2.9
6	Sinus bradycardy	25	5.5
7	Ventricular Premature Contraction (PVC)	3	0.7
8	Supraventricular Premature Contraction	2	0.4
9	Left bundle branch block	9	2.0
10	Right bundle branch block	50	11.1
14	Left ventricule hypertrophy	4	0.9
15	Atrial Fibrillation or Flutter	5	1.1
16	Others	22	4.9

You can see than some arrhythmias type are not present in our dataset (11, 12 and 13).



Approach

The develop of this predicting Cardiac Arrhythmia classification can be explained in the follows steps:

1.- Start analyzing the dataset, cleaning, working on missing values that we found, look for outsider than can be invalid data, review predictors that does not add value and define the outcomes that we are going to predict.

2.- Create the training and validation, or test, dataset.

3.- Works with several classification supervised machines to get a better accuracy in our predictions. Evaluate sensitivity and specificity.

4.- Review the results.

5.- Conclusion.

Methods and Analysis

In this section we explain the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and the modeling approach.

Preparing the data

As we mention,in the description of the dataset is the description of the 279 measurement present in the dataset. The first task it is to set this definition in the name of the column and later set its values according to the one in the description: lineal or nominal, in our case numeric or factor. For that we review the output of the structure created when we read the dataset from UCI.

As a result of this transformation, we have a "well defined" new dataset (only first 25 variables of 280 are showed):

```
## 'data.frame':    452 obs. of  280 variables:
##  $ Age                                                : int  75 56 54 55 75 13 40 49 44 50 ...
##  $ Sex                                                : Factor w/ 2 levels "F","M": 2 1 2 2 2 2 1 1 2 1 ...
##  $ Height                                             : int  190 165 172 175 190 169 160 162 168 167 ...
##  $ Weight                                             : int  80 64 95 94 80 51 52 54 56 67 ...
##  $ QRS duration                                       : int  91 81 138 100 88 100 77 78 84 89 ...
##  $ P-R interval                                       : int  193 174 163 202 181 167 129 0 118 130 ...
##  $ Q-T interval                                       : int  371 401 386 380 360 321 377 376 354 383 ...
##  $ T interval                                         : int  174 149 185 179 177 174 133 157 160 156 ...
##  $ P interval                                         : int  121 39 102 143 103 91 77 70 63 73 ...
##  $ QRS Vector angles                                  : int  -16 25 96 28 -16 107 77 67 61 85 ...
##  $ T Vector angles                                    : int  13 37 34 11 13 66 49 7 69 34 ...
##  $ P Vector angles                                    : int  64 -17 70 -5 61 52 75 8 78 70 ...
##  $ QRST Vector angles                                 : int  -2 31 66 20 3 88 65 51 66 71 ...
##  $ J Vector angles                                    : int  NA NA 23 NA NA NA NA NA 84 NA ...
##  $ Heart rate                                         : int  63 53 75 71 NA 84 70 67 64 63 ...
##  $ Channel DI Q wave Average                          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Channel DI R wave Average                          : int  52 48 40 72 48 36 44 44 40 44 ...
##  $ Channel DI S wave Average                          : int  44 0 80 20 40 48 0 36 0 40 ...
##  $ Channel DI R' wave Average                         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Channel DI S' Average                              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Channel DI Number of intrinsic deflections         : int  32 24 24 48 28 20 24 24 20 28 ...
##  $ Channel DI Existence of ragged R wave              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Channel DI Existence of diphasic derivation of R wave  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Channel DI Existence of ragged P wave              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Channel DI Existence of diphasic derivation of P wave  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##   [list output truncated]
```

Cleaning

The first analysis is to review if all the predictor give information. For that we are going to check if all the values of the dataset are equal, e. gr., it has standard deviation equal to zero. We are going to detect them.

The predictors are:

```
sd.arrhythmia <- apply(arrhythmia,2,sd, na.rm=TRUE)
sd0.list.arrhythmia <-
  names(sd.arrhythmia[sd.arrhythmia == 0 & !is.na(sd.arrhythmia)]) # check is.na

# Identify the column number to be deleted
ix <- which(colnames(arrhythmia) %in% sd0.list.arrhythmia)
variable.names(arrhythmia[ix])
```

```
##  [1] "Channel DI S' Average"
##  [2] "Channel AVL S' wave Average"
##  [3] "Channel AVL Existence of ragged R wave"
##  [4] "Channel AVF Existence of ragged P wave"
##  [5] "Channel V4 Existence of ragged P wave"
##  [6] "Channel V4 Existence of diphasic derivation of P wave"
##  [7] "Channel V5 S' wave Average"
##  [8] "Channel V5 Existence of ragged R wave"
```

```
##  [9] "Channel V5 Existence of ragged P wave"
## [10] "Channel V5 Existence of ragged T wave"
## [11] "Channel V6 S' wave Average"
## [12] "Channel V6 Existence of diphasic derivation of P wave"
## [13] "Channel V6 Existence of ragged T wave"
## [14] "Channel DI S' wave Amplitude"
## [15] "Channel AVL S' wave Amplitude"
## [16] "Channel V5 S' wave Amplitude"
## [17] "Channel V6 S' wave Amplitude"
```

and we delete them from our dataset:

```
arrhythmia <- arrhythmia[,-ix]
```

Cleaning, working with NA

As we mention before, some values are not defined. Initially they are "?" and we change for NA. This table show us which predictor are involve and how many they are:

Table 2: Table of Ocurrence of NA

	Ocurrences
T Vector angles	8
P Vector angles	22
QRST Vector angles	1
J Vector angles	376
Heart rate	1

The algorithms required that all the columns has values. How are we going to proceed with this missing values?

Missing J Vector angles

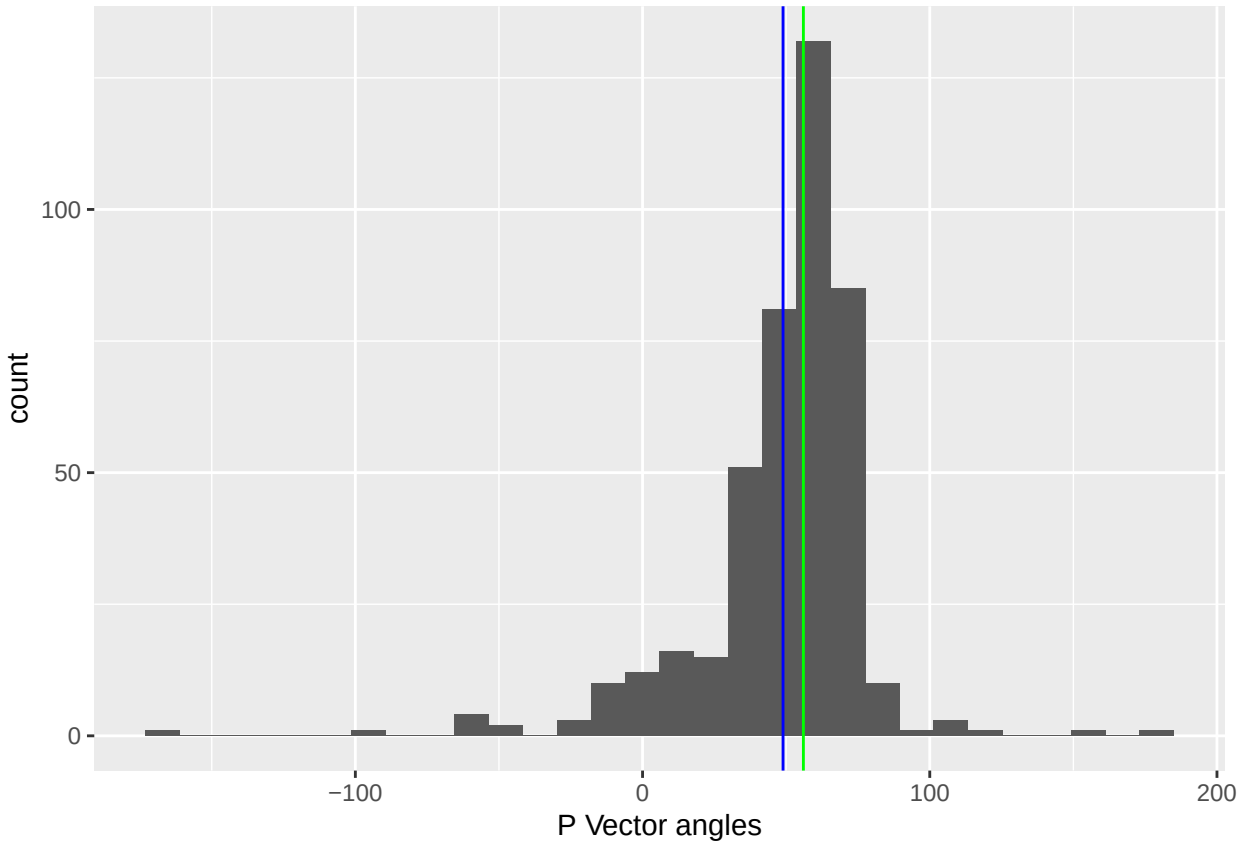
Because we have a big number of this missing values (83.2%) the better solution it is not consider this predictor.

```
arrhythmia <- arrhythmia %>% select(-`J Vector angles`)
```

Missing P Vector angles

This is not as big as the previous one, only 4.9%, let's see how it is distributed

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -170.00   41.00   56.00   48.91   65.00  176.00      22
```



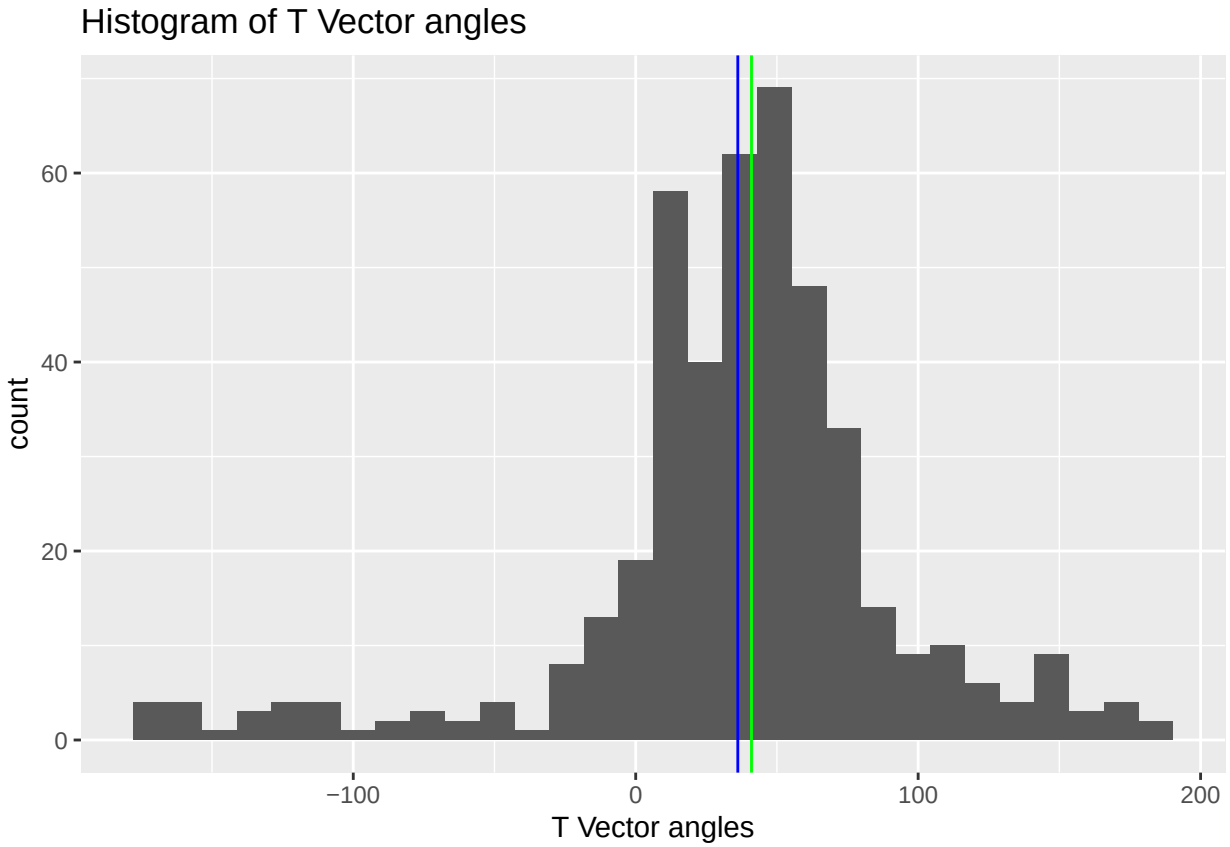
This show in blue the mean and in green the median for this predictors. We are going to use the median which look more independent of the low and high values that look like an outsiders than the mean. Anyway, both values are pretty close but median will be the preference in this and next cases.

```
arrhythmia$`P Vector angles`[is.na(arrhythmia$`P Vector angles`)] <-
  median(arrhythmia$`P Vector angles`,na.rm = TRUE)
```

Missing T Vector angles

This is smaller number of the previous one, only 1.8%, let's see how it is distributed

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -177.00   14.00   41.00   36.15   63.25  179.00       8
```



This show in blue the mean and in green the median for this predictors. We are going to use the median which look more independent of the low values as before.

```
arrhythmia$`T Vector angles`[is.na(arrhythmia$`T Vector angles`)] <-
  median(arrhythmia$`T Vector angles`,na.rm = TRUE)
```

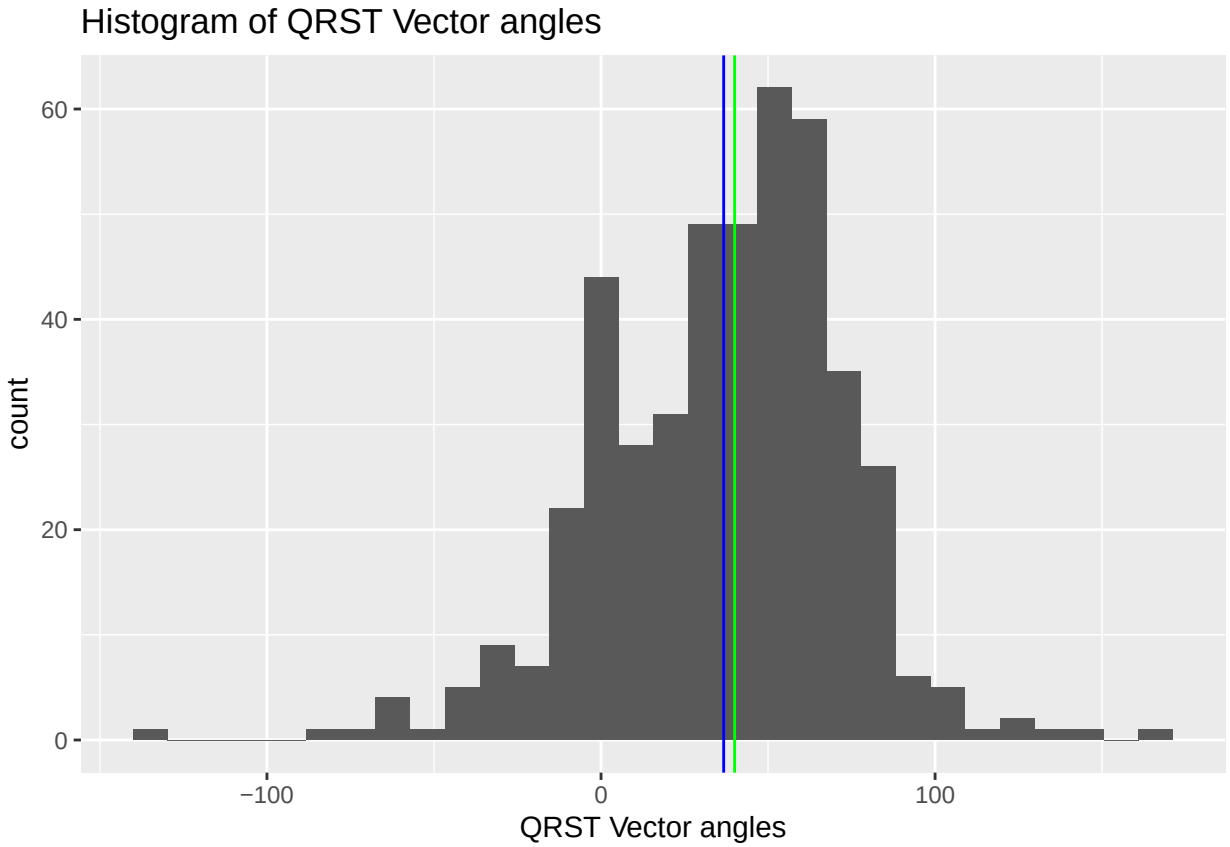
Missing QRST Vector angles

This is only one missing value and represent the 0.2%, let's see some data for this missing:

```
##   Age Sex Height Weight QRS duration P-R interval Q-T interval T interval
## 1  62   F    155     78           90          172          297        209
##   P interval QRS Vector angles
## 1        103                2
```

Let's do the same analysis about the distribution:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -135.00   12.00   40.00   36.72   62.00  166.00       1
```



This show in blue the mean and in green the median for this predictors. We are going to use the median which look more independent of the low and hkgh values as before.

```
arrhythmia$`QRST Vector angles`[is.na(arrhythmia$`QRST Vector angles`)] <-
  median(arrhythmia$`QRST Vector angles`,na.rm = TRUE)
```

Missing Heart rate

This is only one missing value and represent the 0.2%, let's see some data fo this missing:

```
##   Age Sex Height Weight QRS duration P-R interval Q-T interval T interval
## 1  75   M    190     80           88          181          360        177
##   P interval
## 1        103
```

Let's do the same analysis about the distribution:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   44.00   65.00   72.00   74.46   81.00  163.00       1
```

This show in blue the mean and in green the median for this predictors. Again, these values are pretty close. We are going to use the median anyway, getting a little independence of the high value of it.

```
arrhythmia$`Heart rate`[is.na(arrhythmia$`Heart rate`)] <- median(arrhythmia$`Heart rate`,na.rm = TRUE)
```

Correlation

When we use the median in the section abode, we assume that the data is not related with other predictor. Is this true? Now that we have data we can check if this is correct for each predictor we keep.

For this answer we are going to use correlation between the predictors.

For this table the more positive correlated are:

Table 3: Top 10 Positive Correlation between predictors

correlated.1	correlated.2	value
Channel V6 R' wave Average	Channel V6 R' wave Amplitude	0.9977688
Channel V3 S' wave Average	Channel V3 R' wave Amplitude	0.9666644
Channel AVL R' wave Average	Channel AVL R' wave Amplitude	0.9415167
Channel V5 R' wave Average	Channel V5 R' wave Amplitude	0.9334951
Channel AVR R' wave Average	Channel AVR R' wave Amplitude	0.9322929
Channel DIII R wave Amplitude	Channel AVF R wave Amplitude	0.9226571
Channel DIII QRSA	Channel AVF QRSA	0.9172638
Channel DII R wave Amplitude	Channel AVF R wave Amplitude	0.9130322
Channel DI R' wave Average	Channel DI R' wave Amplitude	0.9097042
Channel V5 P wave Amplitude	Channel V6 P wave Amplitude	0.8985656

Table 4: Top 10 Negative Correlation between predictors

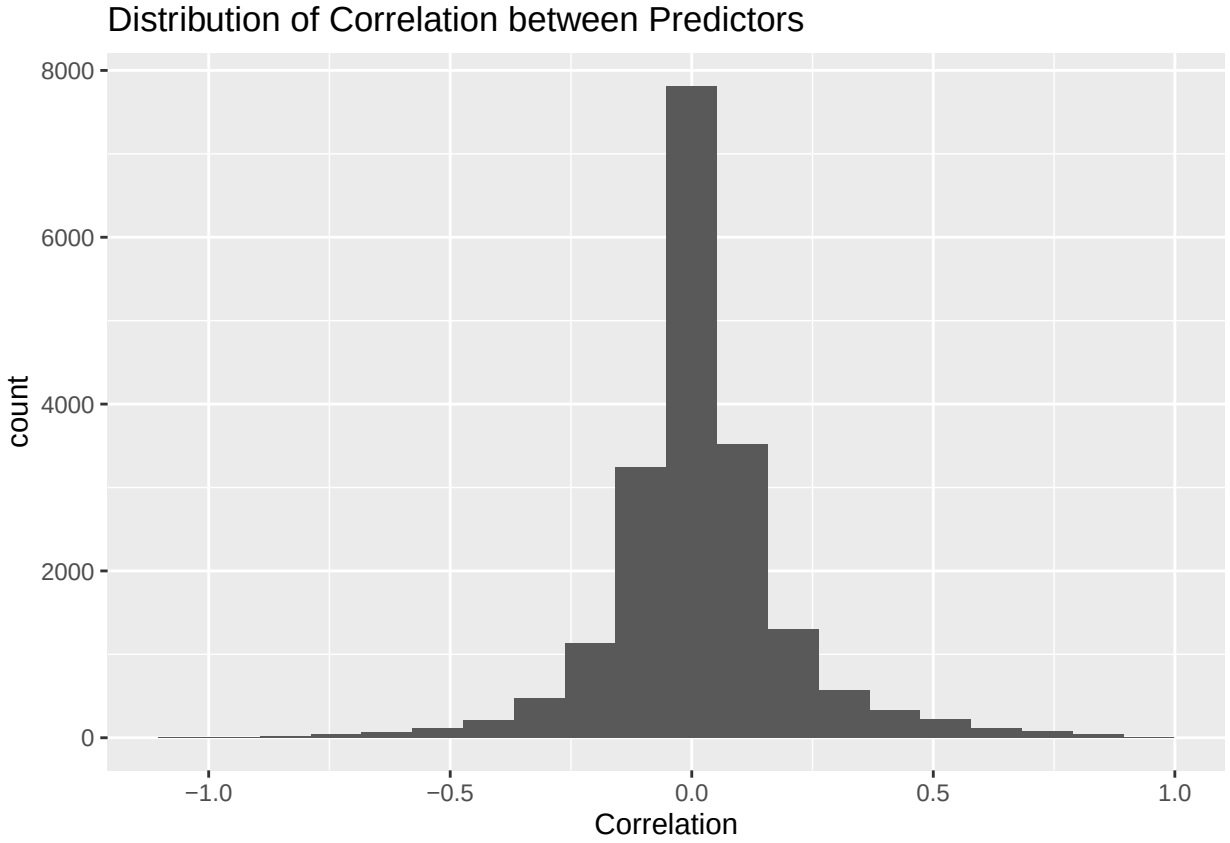
correlated.1	correlated.2	value
Channel AVR S' wave Average	Channel AVR S' wave Amplitude	-1.0000000
Channel V4 S' wave Average	Channel V4 S' wave Amplitude	-0.9992865
Channel AVF S' wave Average	Channel AVF S' wave Amplitude	-0.9673527
Channel V1 S' wave Average	Channel V1 S' wave Amplitude	-0.9577530
Channel V2 S' wave Average	Channel V2 S' wave Amplitude	-0.9532078
Channel DIII S' wave Average	Channel DIII S' wave Amplitude	-0.9405292
Channel DI Q wave Average	Channel DI Q wave Amplitude	-0.9367170
Channel DII Q wave Average	Channel DII Q wave Amplitude	-0.9326229
Channel AVR S wave Average	Channel AVR S wave Amplitude	-0.9175822
Channel DIII QRSA	Channel AVL QRSA	-0.9001125

How many predictors are correlated, positive and negative, more than .9 (high correlated)?

Table 5: Correlation over 0.9 & under -0.9

n
19

This number indicate that the second predictor (or the first if you prefer) does not give us valuable information because the other predictor already give us that information. But how are these distributed?



Only few are high correlated. This is good for our analysis.

Correlation of previous missing values

In the previous predictor that we have missing values, are some others predictor which has complete information are high correlated?

Correlation of P Vector angles Let's start with P Vector angles, where we have 22 missing values.

Table 6: Correlation for P Vector angles

correlated.1	correlated.2	value
P Vector angles	Channel DIII P wave Amplitude	0.7267749
P Vector angles	Channel AVF P wave Amplitude	0.5866874
P Vector angles	Channel DII P wave Amplitude	0.4921735
P Vector angles	Channel V6 P wave Amplitude	0.1834118
P Vector angles	Channel V5 P wave Amplitude	0.1799880

Table 7: Correlation for P Vector angles

correlated.1	correlated.2	value
P Vector angles	Channel AVL P wave Amplitude	-0.6451124
P Vector angles	Channel DI P wave Amplitude	-0.2476149
P Vector angles	Channel DI QRSA	-0.2167978
P Vector angles	Channel V2 P wave Amplitude	-0.2099201
P Vector angles	Channel AVL QRSA	-0.2080543

here we have a value of P Vector angles, Channel DIII P wave Amplitude, 0.726774889752902. If this predictor appear with importance in the final algorithm, let take this in consideration.

Correlation of T Vector angles With T Vector angles, where we have 8 missing values.

Table 8: Correlation for T Vector angles

correlated.1	correlated.2	value
T Vector angles	Channel DIII T wave Amplitude	0.5788952
T Vector angles	Channel AVF T wave Amplitude	0.5276111
T Vector angles	Channel DII T wave Amplitude	0.3605350
T Vector angles	Channel DIII JJ wave Amplitude	0.3128032
T Vector angles	Channel AVF JJ wave Amplitude	0.2691447

Table 9: Correlation for T Vector angles

correlated.1	correlated.2	value
T Vector angles	Channel AVL T wave Amplitude	-0.4253222
T Vector angles	Channel AVL JJ wave Amplitude	-0.2031387
T Vector angles	Channel AVL QRSTA	-0.1853143
T Vector angles	Channel AVF Number of intrinsic deflections	-0.1646638
T Vector angles	Channel DI T wave Amplitude	-0.1610487

The correlation here is lower. Then the assumtion of independant behaivor look valid.

Correlation of QRST Vector angles In the case of QRST Vector angles, with only one value missing, we have

Table 10: Correlation for QRST Vector angles

correlated.1	correlated.2	value
QRST Vector angles	Channel DIII QRSTA	0.8114681
QRST Vector angles	Channel DIII QRSA	0.7294109
QRST Vector angles	Channel AVF QRSTA	0.7138752
QRS Vector angles	QRST Vector angles	0.7086816
QRST Vector angles	Channel AVF QRSA	0.6542389

Table 11: Correlation for QRST Vector angles

correlated.1	correlated.2	value
QRST Vector angles	Channel AVL QRSTA	-0.7672979
QRST Vector angles	Channel AVL QRSA	-0.6923002
QRST Vector angles	Channel AVL R wave Amplitude	-0.6153520
QRST Vector angles	Channel AVL R wave Average	-0.5798941
QRST Vector angles	Channel DI QRSA	-0.4849473

Correlation of Heart rate And the "Heart rate", with only one missing value

Table 12: Correlation for Heart rate

correlated.1	correlated.2	value
Heart rate	Channel AVR R wave Amplitude	0.3107140
Height	Heart rate	0.2865992
Heart rate	Channel AVR QRSTA	0.2398373
Heart rate	Channel V1 R wave Amplitude	0.2372274
Heart rate	Channel AVF R' wave Amplitude	0.2186581

Table 13: Correlation for Heart rate

correlated.1	correlated.2	value
Q-T interval	Heart rate	-0.6547041
Heart rate	Channel DI S wave Amplitude	-0.3532640
Heart rate	Channel V6 QRSA	-0.2492729
Heart rate	Channel V6 S wave Amplitude	-0.2364993
Heart rate	Channel V5 QRSA	-0.2361320

and this predictor has very low correlation with all the rest of the predictor

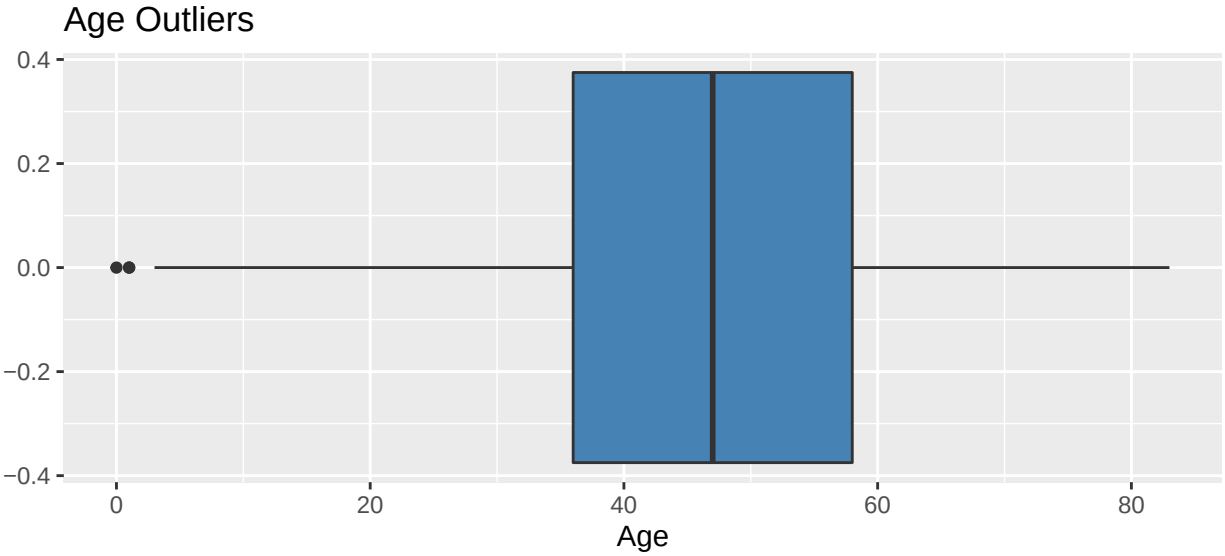
Less Correlation Less correlated predictors are also presented and as we saw in the histogram, they are the most common. For example for the correlation between -0.001 and 0.001 we have:

Table 14: Less Correlated Preditors

correlated.1	correlated.2	value
Weight	Channel V5 R' wave Average	9.06e-05
Channel DI S wave Average	Channel DI P wave Amplitude	-1.10e-06
T interval	Channel DII QRSA	3.40e-06
Channel V2 Q wave Average	Channel AVR R wave Amplitude	-3.59e-05
Channel V6 Q wave Average	Channel AVR R wave Amplitude	-6.80e-05
Channel DIII JJ wave Amplitude	Channel AVR R' wave Amplitude	-6.10e-06
Channel DI Q wave Average	Channel AVR T wave Amplitude	4.49e-05
Channel V2 Number of intrinsic deflections	Channel AVF QRSTA	-4.66e-05
Channel DIII QRSA	Channel V2 S' wave Amplitude	7.89e-05
Channel DIII R' wave Average	Channel V2 P wave Amplitude	-5.20e-06
Channel DII R' wave Amplitude	Channel V3 JJ wave Amplitude	3.10e-06
Channel V2 S wave Amplitude	Channel V3 S' wave Amplitude	7.37e-05
Channel DIII Q wave Average	Channel V5 Q wave Amplitude	9.45e-05
Channel AVF Q wave Amplitude	Channel V5 R' wave Amplitude	3.70e-05

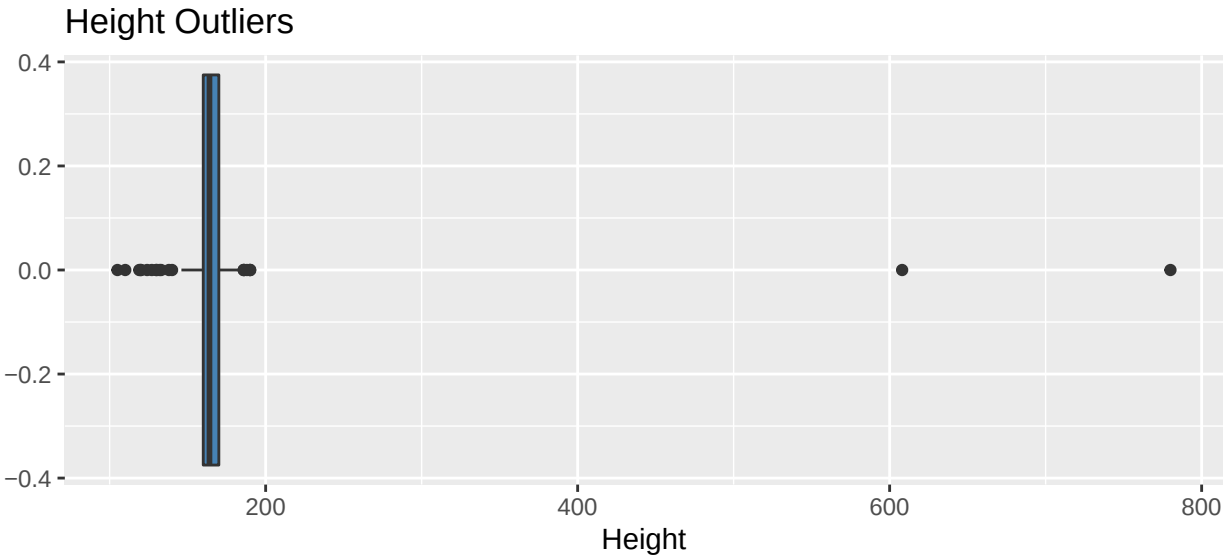
Outliers

Let's take a look to some predictors to see if we can find some outliers. Outliers in statistics are considered as the data values which differ considerably from the bulk of a given data set. These data values lie outside the overall trend, which could be lies in the data. Let's start with some familiar predictors like Age.



It does not show any value we can detect as outlier.

Let's continue with Height. Height are in centimeters.



We see two vales out of expected height of a person:

```
##   Height
## 1    780
## 2    608
```

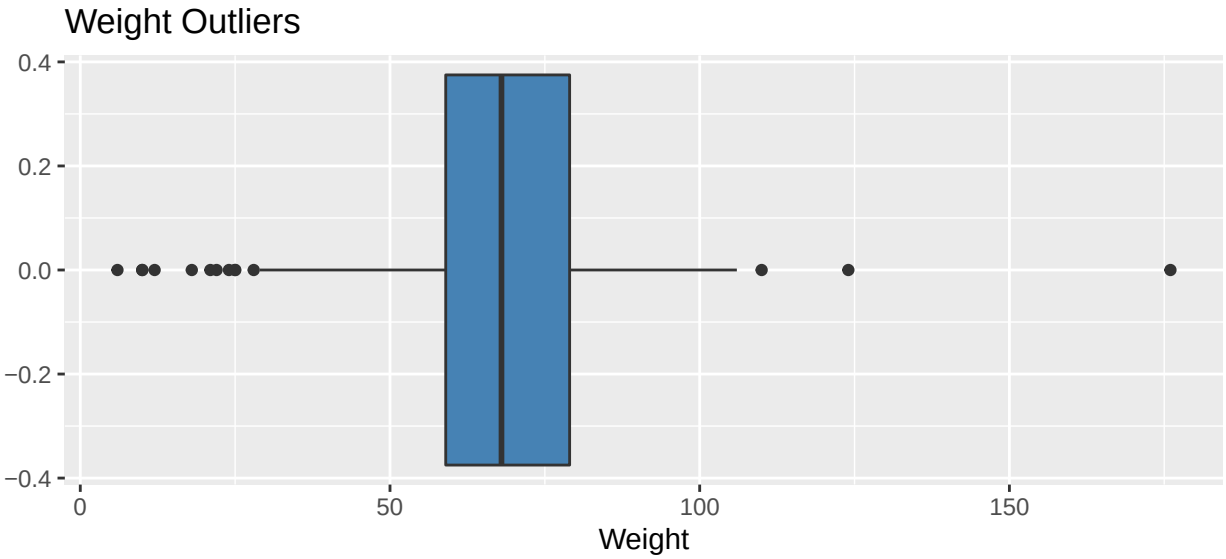
Let's take a look. Can we change for the means o median? Yes, but let's see if we can get more information that can lead us to the right data. Could be a transcription error from US norm and really means 6 feet 8 inches (203 cm) and 7 feet 8 inches (233 cm)? More information we can get from the data:

```
##   Age Weight Height
## 1   1      6    780
## 2   0     10    608
```

They are infants! Then, we can assume, the error is they put the height in mm instead of cm! No other patience less than a year is in the dataset and with one year old there is other one with more weight. For a more precise data maybe a weight-height chat for age can be used. Then we change these values assuming an error in the unit:

```
arrhythmia[arrhythmia$Height == 780,3] <- 78   # Height is column 3
arrhythmia[arrhythmia$Height == 608,3] <- 61   # Height is column 3
```

About the weight in kilograms we have the graph:

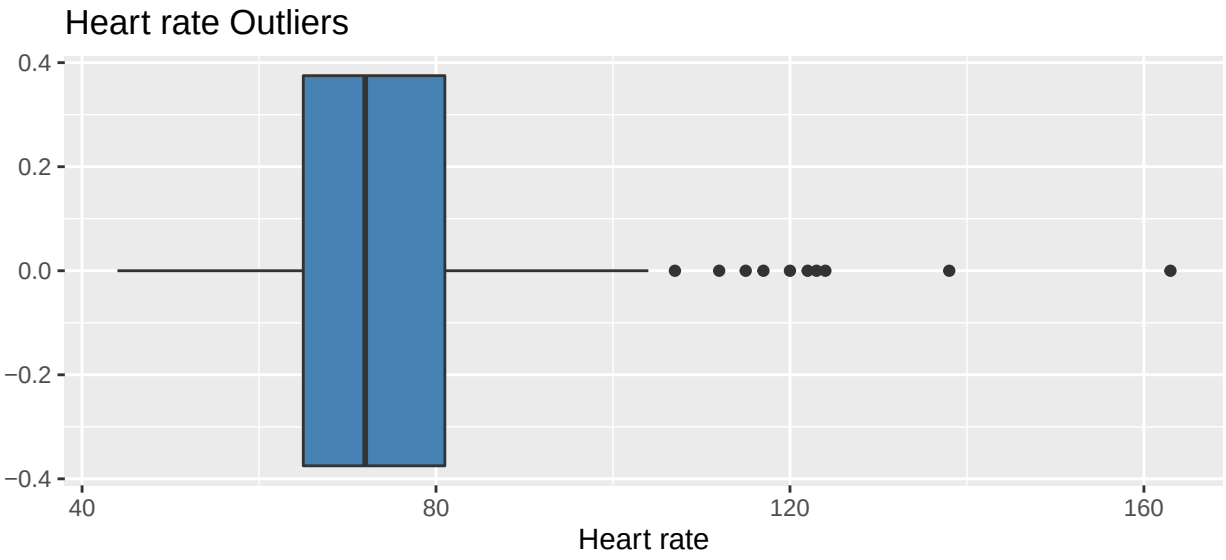


Looking in more detail this patience:

```
##   Age Weight Height
## 1  53    176    169
```

looks plausible and no change will be made.

About the Heart rate in frequency in minutes (Number of heart beats per minute) we have the graph:



In the upper numbers:

```
##   Age Weight Height Heart rate
## 1   0     10     61        163
## 2   1      6     78        138
```

It is look like a possible value (I am not a medical doctor!)

For the predictor above we use our knowledge about the human. A person can not be 7 meter tall, but for the rest, we can review them but for evaluate if it is an outlier and which value can be used to fixed required medical knowledge.

Train and validation dataset

After this analysis and cleaning we are ready to create our training and validation dataset. Because our universe of observation is only 452 we are going to split 80%-20%.

We are going to use "Class code" as an outcome to predict. Because some outcome are too few as we saw before, we need that they are present in the train_set and validation_set. This task should be manager with the createDataPartition() function, but we are going to check anyway.

```
unique(train_set$`Class code`) %>% sort()
```

```
##  [1] 1  2  3  4  5  6  7  8  9  10 14 15 16
## Levels: 1 2 3 4 5 6 7 8 9 10 14 15 16
```

```
unique(validation_set$`Class code`) %>% sort()
```

```
##  [1] 1  2  3  4  5  6  7  8  9  10 14 15 16
## Levels: 1 2 3 4 5 6 7 8 9 10 14 15 16
```

We can see in detail the Class code distribution in both dataset.

check train_set dataset

Table 15: Presence of codes in train_set dataset

Class code	Class name	N Ocurrences	Percentage
1	Normal	196	54.6
2	Ischemic changes (Coronary Artery Disease)	35	9.7
3	Old Anterior Myocardial Infarction	12	3.3
4	Old Inferior Myocardial Infarction	12	3.3
5	Sinus tachycardy	10	2.8
6	Sinus bradycardy	20	5.6
7	Ventricular Premature Contraction (PVC)	2	0.6
8	Supraventricular Premature Contraction	1	0.3
9	Left bundle branch block	7	1.9
10	Right bundle branch block	40	11.1
14	Left ventricule hypertrophy	3	0.8
15	Atrial Fibrillation or Flutter	4	1.1
16	Others	17	4.7

check validation_set dataset

Table 16: Presence of codes in validation_set dataset

Class code	Class name	N Ocurrences	Percentage
1	Normal	49	52.7
2	Ischemic changes (Coronary Artery Disease)	9	9.7
3	Old Anterior Myocardial Infarction	3	3.2
4	Old Inferior Myocardial Infarction	3	3.2
5	Sinus tachycardy	3	3.2
6	Sinus bradycardy	5	5.4
7	Ventricular Premature Contraction (PVC)	1	1.1
8	Supraventricular Premature Contraction	1	1.1
9	Left bundle branch block	2	2.2
10	Right bundle branch block	10	10.8

Class code	Class name	N Ocurrences	Percentage
14	Left ventricule hypertrophy	1	1.1
15	Atrial Fibrillation or Flutter	1	1.1
16	Others	5	5.4

We can see that the percentage of the different classes are not equally distributes in both dataset. Why? Because it is not possible to have the same percentage when there is a few sample in some of them in the initial dataset. For example, Class 8 "Supraventricular Premature Contraction" has only 2 samples, them the function distribute in the 2 dataset and because the total number of samples are different, its percentage is different. This change if we move the partition from 80%-20% to 50%-50% but less samples are left for training propose. This is part of the consideration when we chose this proportion at the beginning.

But this is not the only check we need to perform: we need to see if all the data (predictors) are not equal in this new training dataset. Remember that we check initially in the arrhythmia dataset that this situation does not occurs, but this can be present in this new (and small) dataset. Particularly with factors, if we have one value only we can get the error: "contrasts can be applied only to factors with 2 or more levels" with the train() function.

The predictors that we find with a unique value in the training dataset are:

```
## [1] "Channel DII Existence of diphasic derivation of P wave"
## [2] "Channel DIII Existence of ragged R wave"
## [3] "Channel AVR S' wave Average"
## [4] "Channel V6 Existence of ragged R wave"
## [5] "Channel AVR S' wave Amplitude"
```

We have 5 predictors in this new dataset with equal value, for that we need to delete from train_set and validation_set dataset:

```
train_set <- train_set[,values_count > 1]
validation_set <- validation_set[,values_count > 1]
```

After all this process, we finish with 256 predictors from the initial 279.

In summary, related with the prediction we are going to perform:
* The first prediction is if we can predict is the patience is normal or has one of the 15 arrhythmia.
* The second prediction, as you probably guess, is if we can categorize the right class.

Evaluation metrics

As we mention previously, we are going to implement models for two purposes:

a. **Detection of cardiac arrhythmia**
b. **Classification of cardiac arrhythmia**

In our **first** prediction for detection of cardiac arrhythmia, the data points are classified into two classes: "Normal" & "Arrhythmia". This model only identifies if the patient is normal (class 1) or suffers from any form of arrhythmia (class 2 to 16). Then the output is a binary variable (is a variable that has two possible outcomes). This will possible if all the instances belonging to classes 2 to 16 were merged to one class. The arrhythmia class will be treated as the 'Positive' class.

We are going to see the following evaluation metrics:

- **accuracy** is defined as the overall proportion that is predicted correctly.
- **sensitivity** is defined as the ability of an algorithm to predict a positive outcome when the actual outcome is positive.
- **specificity** is defined as the ability of an algorithm to not predict a positive when the actual outcome is not a positive.

We can summarize in the following way:
High accuracy: $Y = 1 \Rightarrow \hat{Y} = 1$ and $Y = 0 \Rightarrow \hat{Y} = 0$
High sensitivity: $Y = 1 \Rightarrow \hat{Y} = 1$
High specificity: $Y = 0 \Rightarrow \hat{Y} = 0$

We are looking for accuracy and sensitivity. Why? it is much more important to maximize sensitivity over specificity: failing to predict a arrhythmia can put the health/life of the patience in risk. It is better than this patience classified as "Arrhythmia" and make all the exams/procedure and when validate that he or she is ok.

In our **second** prediction model for classification of cardiac arrhythmia classified the patient into one of 16 classes, with class 1 representing "Normal" and classes 2 to 16 representing a condition of cardiac arrhythmia. Here we continue with the accuracy concept but now sensitivity and specificity is for class.

Let's show the standard definition in an binary variable:
True Positive (TN) – This is correctly classified as the class if interest/target.
True Negative (TN) – This is correctly classified as not a class of interest/target.
False Positive (FP) – This is wrongly classified as the class of interest/target.
False Negative (FN) – This is wrongly classified as not a class of interest/target.

	Actually Positive	Actually Negative
Predicted Positive	True Positives (TP)	False Positives (FP)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

An these are the definitions:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$
$$Sensitivity = TP/(TP + FN)$$
$$Specificity = TN/(TN + FP)$$

Whats happened when we have more class? Then we have the same arithmetic for the Accuracy, but we have a value for Sensitivity and Specificity for each class.

	Actually A	Actually B	Actually C
Predicted A	True A (TA)	False A (FA.B)	False A (FA.C)
Predicted B	False B (FB.A)	True B (TB)	False B (FB.C)
Predicted C	False C (FC.A)	False C (FC.B)	True C (TC)

An these are the definitions:

$$Accuracy = (TA + TB + TC)/(TA + TB + TC + FA.B + FA.C + FB.A + FB.C + FC.A + FC.B)$$

$$Sensitivity_A = TA/(TA + FB.A + FC.A)$$
$$Specificity_A = TA/(TA + FA.B + FA.C)$$
$$Sensitivity_B = TB/(TB + FA.B + FC.B)$$
$$Specificity_B = TB/(TB + FB.A + FB.C)$$
$$Sensitivity_C = TC/(TC + FB.C + FA.C)$$
$$Specificity_C = TC/(TC + FA.C + FB.C)$$

First Model. . . just the most common

We are now to create our models. For that we are not to use the validation, we are going to use only train dataset.

The most basic and quick approach it is consider that the most common class (mode), in this case it is the "Normal" class, is the best guess. Then the model is:

$$\hat{y} = y_{mode} + \varepsilon_{i,u}$$

y_{mode} the "true" class for all patience and $\varepsilon_{i,u}$ an independent errors sampled from the same distribution centered at 0.

In the code, for the prediction for the two purposes, we are going to use different dataset to make easier to read the code: the first one, that concern about "Normal" & "Arrhythmia" only, the "Code class" can only has these two values. In the case of Classification of cardiac arrhythmia, the "Code class" has all the values allowed (13). Because it is not a big dataset, this duplicity does not affect the computer resources available fo this project.

First Prediction

Let's works with the first Prediction

```
##  Accuracy
## 0.5268817
```

```
## Sensitivity
##           0
```

```
## Specificity
##           1
```

Because an algorithm that calls everything positive ($Y = 1$ no matter what) has perfect specificity, but worse sensitivity. These are completely opposite if we set as less common. The Accuracy obtained will be our base one.

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##              | Actually
##    Predicted | Arrhythmia |     Normal |  Row Total |
## -------------|------------|------------|------------|
##       Normal |         44 |         49 |         93 |
##              |      0.473 |      0.527 |            |
## -------------|------------|------------|------------|
## Column Total |         44 |         49 |         93 |
## -------------|------------|------------|------------|
##
##
```

For our record, we keep this first model:

Table 19: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Just the most common	0.5268817	0	1

Second Predictions

Second Predictions, that considered all the classes, the accuracy is the same, but now we have sensitivity and specificity by class:

```
##  Accuracy
## 0.5268817
```

```
##           Sensitivity Specificity
## Class: 1            1           0
## Class: 2            0           1
## Class: 3            0           1
## Class: 4            0           1
## Class: 5            0           1
## Class: 6            0           1
## Class: 7            0           1
## Class: 8            0           1
## Class: 9            0           1
## Class: 10           0           1
## Class: 14           0           1
## Class: 15           0           1
## Class: 16           0           1
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##           | Actually
## Predicted |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |    9 |   10 |   14 |   15 |   16 | RowT |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         1 |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
##           | 0.53 | 0.10 | 0.03 | 0.03 | 0.03 | 0.05 | 0.01 | 0.01 | 0.02 | 0.11 | 0.01 | 0.01 | 0.05 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##   ColumnT |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##
##
```

We can see that only class 1 has good sensitivity, as we expected.

For our record, we keep this second model:

Table 20: Prediction All Arrhytmia Classification Summary

Model	Just the most common
Accuracy	0.5268817
Code.1	Class: 1
Sensitivity.1	1
Specificity.1	0
Code.2	Class: 2
Sensitivity.2	0
Specificity.2	1
Code.3	Class: 3
Sensitivity.3	0
Specificity.3	1
Code.4	Class: 4

Table 20: Prediction All Arrhytmia Classification Summary

Sensitivity.4	0
Specificity.4	1
Code.5	Class: 5
Sensitivity.5	0
Specificity.5	1
Code.6	Class: 6
Sensitivity.6	0
Specificity.6	1
Code.7	Class: 7
Sensitivity.7	0
Specificity.7	1
Code.8	Class: 8
Sensitivity.8	0
Specificity.8	1
Code.9	Class: 9
Sensitivity.9	0
Specificity.9	1
Code.10	Class: 10
Sensitivity.10	0
Specificity.10	1
Code.11	Class: 14
Sensitivity.11	0
Specificity.11	1
Code.12	Class: 15
Sensitivity.12	0
Specificity.12	1
Code.13	Class: 16
Sensitivity.13	0
Specificity.13	1

Then this first approach give us a base line to the next ones.

K-Nearest Neighbors Model

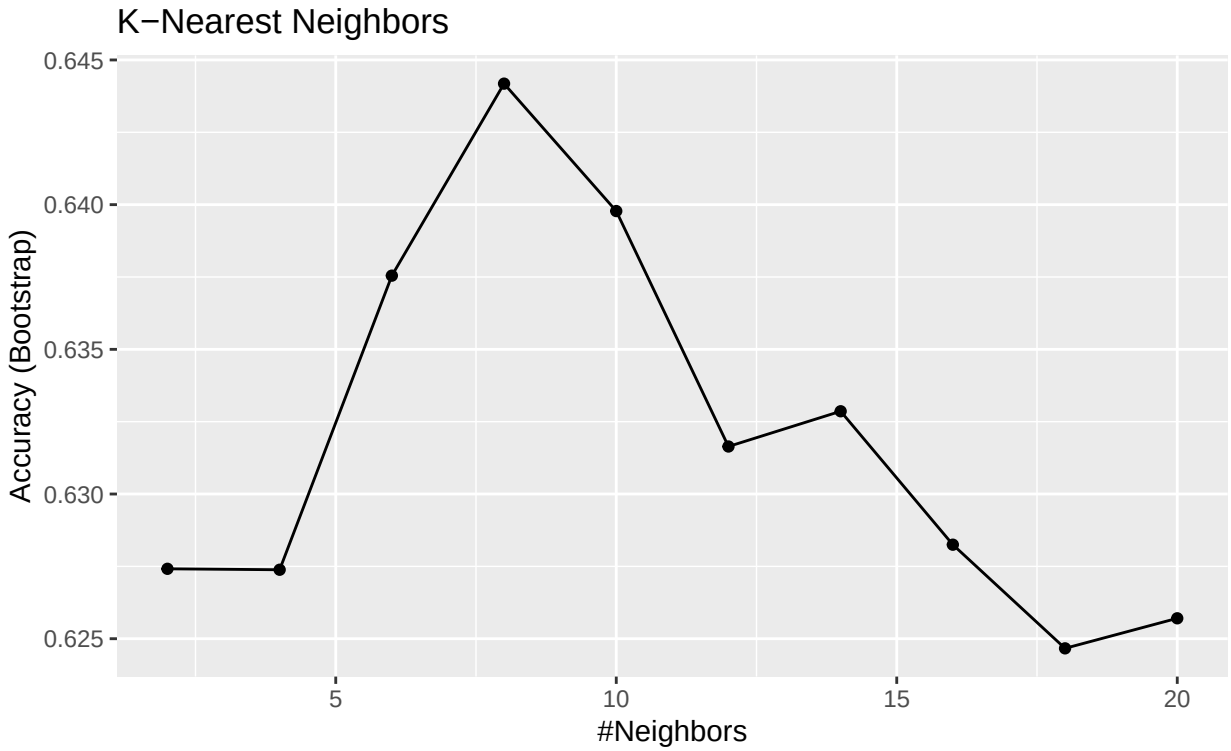
Our first algorithm will be the K-Nearest Neighbors. KNN is a non-parametric method and makes no assumptions.

In base of the concept of distance, this method attempts to find K nearest points of the train data point to the test data point and assigns the class to it on basis of majority for K nearest points. Then the basic question is which value of k give a better solution.

Because we have to few data to train, we use cross-validation in our training process.

First Prediction

Let's works with the first Prediction. In the first graph we are looking for the better k (number of Neighbors) that increase our accuracy. This parameter will be used later, for our prediction.

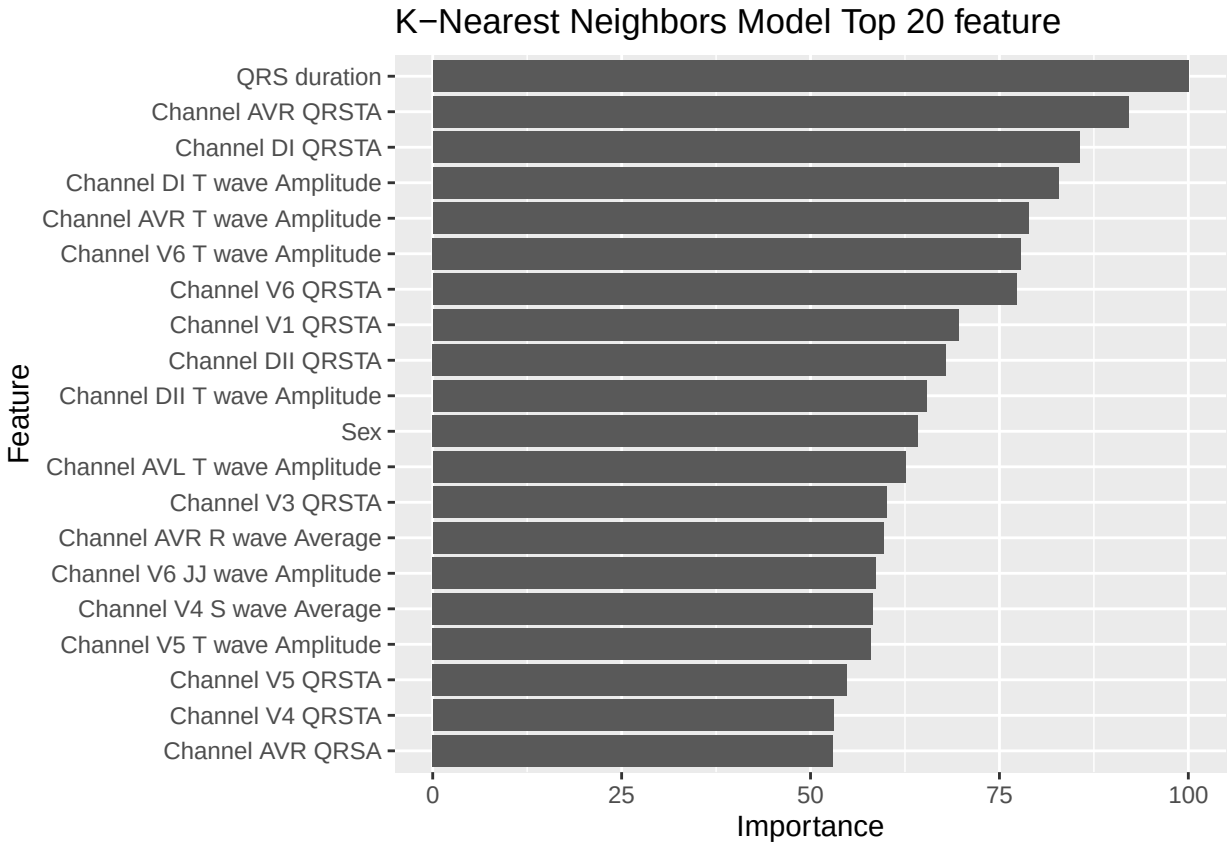



```
## k-Nearest Neighbors
##
## 359 samples
## 256 predictors
##   2 classes: 'Arrhythmia', 'Normal'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    2  0.6274157  0.2280484
##    4  0.6273836  0.2233432
##    6  0.6375462  0.2370438
##    8  0.6441816  0.2469208
##   10  0.6397781  0.2341343
##   12  0.6316420  0.2132738
##   14  0.6328600  0.2129666
##   16  0.6282492  0.2019765
##   18  0.6246673  0.1924420
##   20  0.6257060  0.1920652
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 8.
```

We get k = 8. Now the variables that this model considers more important are:

```
## ROC curve variable importance
##
##   only 20 most important variables shown (out of 256)
##
##                                Importance
## QRS duration                       100.00
## Channel AVR QRSTA                   92.07
## Channel DI QRSTA                    85.58
## Channel DI T wave Amplitude         82.83
## Channel AVR T wave Amplitude        78.83
## Channel V6 T wave Amplitude         77.85
## Channel V6 QRSTA                    77.27
## Channel V1 QRSTA                    69.58
## Channel DII QRSTA                   67.96
## Channel DII T wave Amplitude        65.37
## Sex                                 64.20
## Channel AVL T wave Amplitude        62.56
```

```
## Channel V3 QRSTA                 60.07
## Channel AVR R wave Average       59.63
## Channel V6 JJ wave Amplitude     58.65
## Channel V4 S wave Average        58.28
## Channel V5 T wave Amplitude      58.03
## Channel V5 QRSTA                 54.84
## Channel V4 QRSTA                 53.07
## Channel AVR QRSA                 52.95
```



With this model created in base of the train dataset we now take the validation dataset to see how well this apply:

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Arrhythmia Normal
##   Arrhythmia          9      2
##   Normal             35     47
##
##                Accuracy : 0.6022
##                  95% CI : (0.4954, 0.7022)
##     No Information Rate : 0.5269
##     P-Value [Acc > NIR] : 0.0881
##
##                   Kappa : 0.1702
##
##  Mcnemar's Test P-Value : 1.435e-07
##
##             Sensitivity : 0.20455
##             Specificity : 0.95918
##          Pos Pred Value : 0.81818
##          Neg Pred Value : 0.57317
##              Prevalence : 0.47312
##          Detection Rate : 0.09677
##    Detection Prevalence : 0.11828
##       Balanced Accuracy : 0.58186
##
##        'Positive' Class : Arrhythmia
##
```

We see a good improvement in accuracy and sensitivity over the previous approach. Then we see that KNN is able to create a better prediction than just the mode of the patients. We just started with this process, let's see if we can make it better,

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##              | Actually
##    Predicted | Arrhythmia |     Normal |  Row Total |
## -------------|------------|------------|------------|
##   Arrhythmia |          9 |          2 |         11 |
##              |      0.818 |      0.182 |      0.118 |
##              |      0.205 |      0.041 |            |
## -------------|------------|------------|------------|
##       Normal |         35 |         47 |         82 |
##              |      0.427 |      0.573 |      0.882 |
##              |      0.795 |      0.959 |            |
## -------------|------------|------------|------------|
## Column Total |         44 |         49 |         93 |
##              |      0.473 |      0.527 |            |
## -------------|------------|------------|------------|
##
##
```

For our record, we add this model:

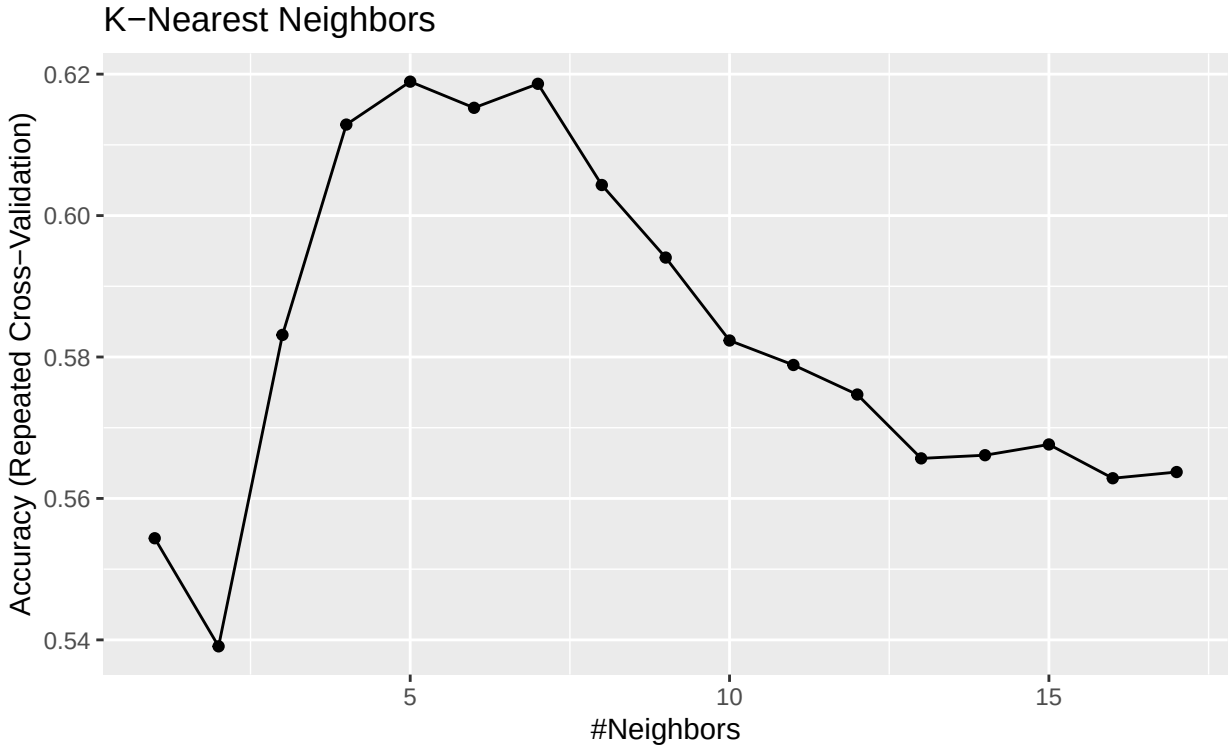
Table 21: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Just the most common	0.5268817	0.0000000	1.0000000
K-Nearest Neighbors	0.6021505	0.2045455	0.9591837

With this first model we have a good improvement. KNN in overall, has a poor sensitivity and very good specificity. If "Arrhythmia" es predicted we have a good chance to be correct, but if we predict "Normal" we have a good chance to be wrong! We need to improve sensitivity.

Second Predictions

Let's works with the second Prediction, that include all the classes and the challenge is to know which of all the arrhythmia classification (or normal) the patient has. Again in the first graph we are looking for the better k (number of Neighbors):

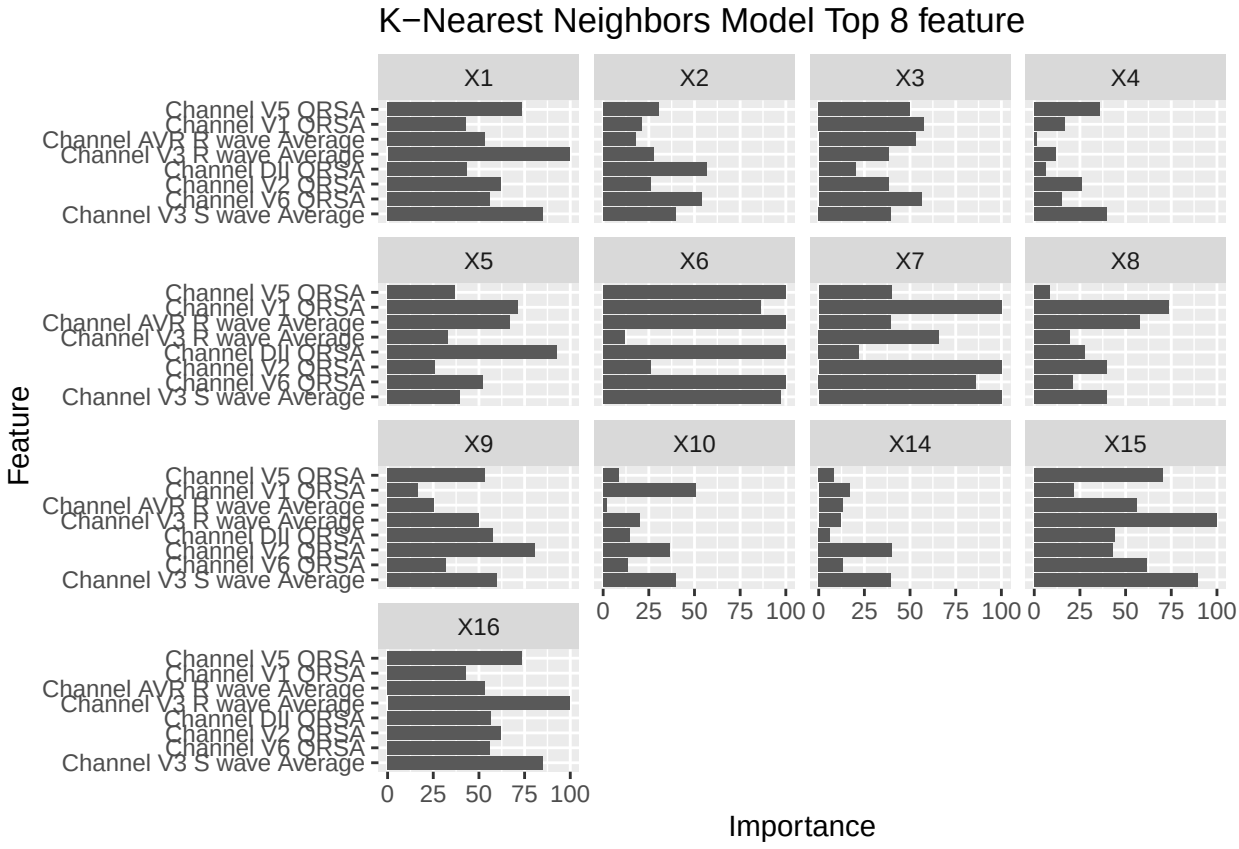



```
## k-Nearest Neighbors
##
## 359 samples
## 256 predictors
##  13 classes: '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '14', '15', '16'
##
## No pre-processing
## Resampling: Cross-Validated (20 fold, repeated 3 times)
## Summary of sample sizes: 340, 340, 342, 341, 340, 341, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.5543723  0.28934803
##    2  0.5390860  0.25116890
##    3  0.5831253  0.25477633
##    4  0.6128689  0.27844147
##    5  0.6189397  0.24731682
##    6  0.6152371  0.22427552
##    7  0.6186334  0.22639690
##    8  0.6043204  0.18541865
##    9  0.5940607  0.14816541
##   10  0.5823344  0.11542488
##   11  0.5788734  0.10724741
##   12  0.5746921  0.08878364
##   13  0.5656645  0.06094346
##   14  0.5661142  0.06124247
##   15  0.5676330  0.05860157
##   16  0.5628563  0.04518964
##   17  0.5637335  0.04340292
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

Now the value of k = 5 is smaller than in the first prediction. Now the variables that this model considers more important are:

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##   only 20 most important variables shown (out of 256)
##
##                                X1    X2    X3    X4   X5     X6     X7    X8
## Channel DII S wave Amplitude  28.83 20.41 25.15  5.98 49.7 100.00  23.25 54.77
## Channel AVF QRSA              23.47 44.56 12.40  3.21 89.3 100.00  31.71 12.30
## Channel AVR QRSA              62.46 38.73 40.41 14.87 87.0 100.00  52.77 45.33
```

```
## Channel DII QRSA              43.62 56.55 20.10  6.25 92.9 100.00  22.01 27.74
## Channel AVF S wave Amplitude  37.50 36.14  1.73  2.93 26.5 100.00  69.02 39.29
## Channel V3 Q wave Average    100.00  2.04  8.16  8.11 49.0   2.04   2.04  2.04
## Channel V6 S wave Amplitude   17.26 26.96 52.09 29.44 64.0 100.00  12.61 42.10
## Channel V6 QRSTA              67.98 85.33 59.04 59.04 59.0 100.00  59.04 59.04
## Channel V3 S wave Average     84.95 39.42 39.42 39.42 39.4  96.94 100.00 39.42
## QRS Vector angles              2.89 45.07 29.49  2.06 90.8 100.00  40.67 11.56
## Channel V3 Q wave Amplitude  100.00  2.04  7.96  8.06 49.0   2.04   2.04  2.04
## Channel V1 S wave Average     64.88  6.97 34.13  6.97 29.6 100.00  21.28 58.24
## Channel V6 QRSA               56.04 54.08 56.28 15.05 52.0 100.00  86.08 20.88
## Channel V3 R wave Average     99.62 27.76 38.11 11.95 32.9  11.95  65.74 19.41
## Channel AVR R wave Average    53.32 18.03 52.91  1.43 67.1 100.00  39.29 57.70
## QRS duration                  51.28 40.66 40.66 40.66 70.4  96.43 100.00 56.34
## QRST Vector angles            11.35 33.12 43.83  3.56 86.2 100.00  19.83 16.59
## Channel V1 JJ wave Amplitude  27.97 27.97 57.19 27.97 28.0  86.22 100.00 27.97
## Channel V2 QRSA               62.03 25.93 38.21 25.93 25.9  25.93 100.00 39.74
## Channel DII QRSTA             63.69 74.45 46.94 46.94 74.2 100.00  46.94 46.94
##                                 X9   X10   X14   X15   X16
## Channel DII S wave Amplitude 42.18 30.87  5.98 31.90 28.83
## Channel AVF QRSA             76.53  3.21  3.21 22.86 44.56
## Channel AVR QRSA             22.79 23.34 11.95 65.95 62.46
## Channel DII QRSA             57.65 14.29  6.25 44.05 56.55
## Channel AVF S wave Amplitude  1.70 46.94  3.54 34.52 37.50
## Channel V3 Q wave Average    31.97 23.47 15.79 100.00 100.00
## Channel V6 S wave Amplitude  38.78 17.60 20.53 25.71 26.96
## Channel V6 QRSTA             59.04 80.87 59.04 59.04 85.33
## Channel V3 S wave Average    60.03 39.42 39.42 89.52 84.95
## QRS Vector angles            94.90 30.61 15.94  3.57 45.07
## Channel V3 Q wave Amplitude  31.29 23.47 15.76 100.00 100.00
## Channel V1 S wave Average     6.97 29.21  6.97 68.33 64.88
## Channel V6 QRSA              31.80 13.32 13.32 61.67 56.04
## Channel V3 R wave Average    50.17 19.90 11.95 100.00 99.62
## Channel AVR R wave Average   25.34  1.79 12.97 55.95 53.32
## QRS duration                 81.46 40.66 45.77 40.66 51.28
## QRST Vector angles           85.20  3.56 21.40 12.38 33.12
## Channel V1 JJ wave Amplitude 38.44 52.42 33.82 27.97  9.61
## Channel V2 QRSA              80.78 36.48 40.22 42.86 62.03
## Channel DII QRSTA            71.77 79.97 46.94 46.94 74.45
```



We noted that the variable are not the same compare with the same model but only looking to predict

Normal/Arrhythmia, some of them appear in other importance position and other are new. This important variables is for each output, that give us more detail that the other measurement that we see in the others mdels used in this analysis.

Now see how well it is the prediction in the validation set:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 14 15 16
##         1  49  8  0  3  3  4  1  1  0  8  1  1  5
##         2   0  1  0  0  0  0  0  0  0  0  0  0  0
##         3   0  0  3  0  0  0  0  0  0  0  0  0  0
##         4   0  0  0  0  0  0  0  0  0  0  0  0  0
##         5   0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   0  0  0  0  0  1  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0  2  0  0  0  0
##         10  0  0  0  0  0  0  0  0  0  2  0  0  0
##         14  0  0  0  0  0  0  0  0  0  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.624
##                  95% CI : (0.517, 0.722)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 0.0381
##
##                   Kappa : 0.274
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity             1.000   0.1111   1.0000   0.0000   0.0000   0.2000
## Specificity             0.205   1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value          0.583   1.0000   1.0000      NaN      NaN   1.0000
## Neg Pred Value          1.000   0.9130   1.0000   0.9677   0.9677   0.9565
## Prevalence              0.527   0.0968   0.0323   0.0323   0.0323   0.0538
## Detection Rate          0.527   0.0108   0.0323   0.0000   0.0000   0.0108
## Detection Prevalence    0.903   0.0108   0.0323   0.0000   0.0000   0.0108
## Balanced Accuracy       0.602   0.5556   1.0000   0.5000   0.5000   0.6000
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 14 Class: 15
## Sensitivity            0.0000   0.0000   1.0000    0.2000    0.0000    0.0000
## Specificity            1.0000   1.0000   1.0000    1.0000    1.0000    1.0000
## Pos Pred Value            NaN      NaN   1.0000    1.0000       NaN       NaN
## Neg Pred Value         0.9892   0.9892   1.0000    0.9121    0.9892    0.9892
## Prevalence             0.0108   0.0108   0.0215    0.1075    0.0108    0.0108
## Detection Rate         0.0000   0.0000   0.0215    0.0215    0.0000    0.0000
## Detection Prevalence   0.0000   0.0000   0.0215    0.0215    0.0000    0.0000
## Balanced Accuracy      0.5000   0.5000   1.0000    0.6000    0.5000    0.5000
##                      Class: 16
## Sensitivity             0.0000
## Specificity             1.0000
## Pos Pred Value             NaN
## Neg Pred Value          0.9462
## Prevalence              0.0538
## Detection Rate          0.0000
## Detection Prevalence    0.0000
## Balanced Accuracy       0.5000
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##           | Actually
## Predicted |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |    9 |   10 |   14 |   15 |   16 | RowT |
```

```
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         1 |   49 |    8 |    0 |    3 |    3 |    4 |    1 |    1 |    0 |    8 |    1 |    1 |    5 |   84 |
##           | 0.58 | 0.10 | 0.00 | 0.04 | 0.04 | 0.05 | 0.01 | 0.01 | 0.00 | 0.10 | 0.01 | 0.01 | 0.06 | 0.90 |
##           | 1.00 | 0.89 | 0.00 | 1.00 | 1.00 | 0.80 | 1.00 | 1.00 | 0.00 | 0.80 | 1.00 | 1.00 | 1.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         2 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    1 |
##           | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 |
##           | 0.00 | 0.11 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         3 |    0 |    0 |    3 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    3 |
##           | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 |
##           | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         6 |    0 |    0 |    0 |    0 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    1 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         9 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    2 |    0 |    0 |    0 |    0 |    2 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##        10 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    2 |    0 |    0 |    0 |    2 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.02 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##   ColumnT |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
##           | 0.53 | 0.10 | 0.03 | 0.03 | 0.03 | 0.05 | 0.01 | 0.01 | 0.02 | 0.11 | 0.01 | 0.01 | 0.05 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##
##
```

As seen above, KNN cannot accurately classify test observations for classes 5 to 8 and between class11 to 16. In total 7 classes with sensitivity = 0 and specificity = 1. Almost completely misclassifies in class 2 that are observations have been classified to class 1. A perfect match in class 3 and 9.

For our record, we keep this result:

Table 22: Prediction All Arrhytmia Classification Summary

Model	Just the most common	K-Nearest Neighbors
Accuracy	0.5268817	0.6236559
Code.1	Class: 1	Class: 1
Sensitivity.1	1	1
Specificity.1	0.0000000	0.2045455
Code.2	Class: 2	Class: 2
Sensitivity.2	0.0000000	0.1111111
Specificity.2	1	1
Code.3	Class: 3	Class: 3
Sensitivity.3	0	1
Specificity.3	1	1
Code.4	Class: 4	Class: 4
Sensitivity.4	0	0
Specificity.4	1	1
Code.5	Class: 5	Class: 5
Sensitivity.5	0	0
Specificity.5	1	1
Code.6	Class: 6	Class: 6
Sensitivity.6	0.0	0.2
Specificity.6	1	1
Code.7	Class: 7	Class: 7
Sensitivity.7	0	0
Specificity.7	1	1
Code.8	Class: 8	Class: 8
Sensitivity.8	0	0
Specificity.8	1	1

Table 22: Prediction All Arrhytmia Classification Summary

Code.9	Class: 9	Class: 9
Sensitivity.9	0	1
Specificity.9	1	1
Code.10	Class: 10	Class: 10
Sensitivity.10	0.0	0.2
Specificity.10	1	1
Code.11	Class: 14	Class: 14
Sensitivity.11	0	0
Specificity.11	1	1
Code.12	Class: 15	Class: 15
Sensitivity.12	0	0
Specificity.12	1	1
Code.13	Class: 16	Class: 16
Sensitivity.13	0	0
Specificity.13	1	1

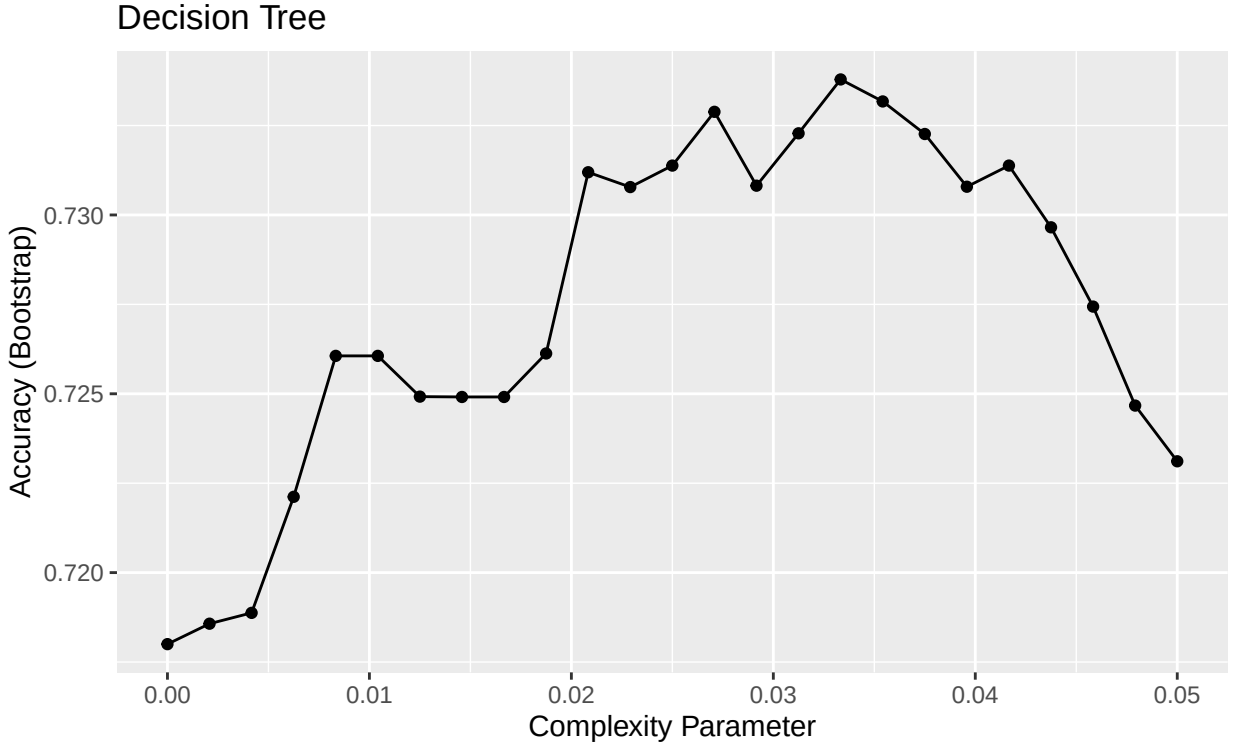
Decision Tree Model

Decision Tree is the most intuitive solution that we get from data, and it s very popular en health, that the system give you a quick rule to answer your question.

For this model we are going to user the rpart package and the cp (complexity parameter) is our parameter to start moving to get a better accuracy.

First Prediction

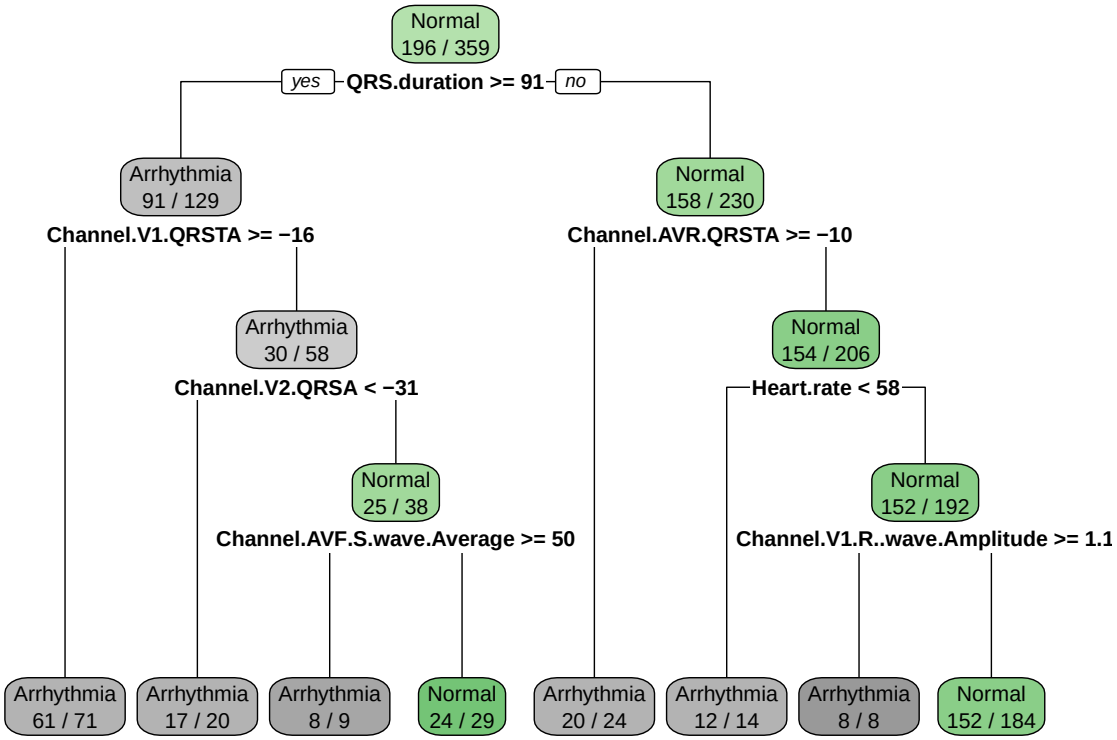
Let's works with the first Prediction and look for the cp (complexity parameter) that increase our accuracy:




```
## CART
```

```
##
## 359 samples
## 256 predictors
##   2 classes: 'Arrhythmia', 'Normal'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.000000000  0.7179999  0.4265427
##   0.002083333  0.7185713  0.4274721
##   0.004166667  0.7188744  0.4276031
##   0.006250000  0.7221191  0.4342277
##   0.008333333  0.7260595  0.4423981
##   0.010416667  0.7260595  0.4425118
##   0.012500000  0.7249201  0.4402098
##   0.014583333  0.7249109  0.4405318
##   0.016666667  0.7249109  0.4405318
##   0.018750000  0.7261269  0.4431310
##   0.020833333  0.7311939  0.4531143
##   0.022916667  0.7307792  0.4524474
##   0.025000000  0.7313806  0.4533381
##   0.027083333  0.7328833  0.4572348
##   0.029166667  0.7308181  0.4524552
##   0.031250000  0.7322824  0.4559502
##   0.033333333  0.7337891  0.4585665
##   0.035416667  0.7331737  0.4573773
##   0.037500000  0.7322633  0.4571489
##   0.039583333  0.7307893  0.4537030
##   0.041666667  0.7313804  0.4550288
##   0.043750000  0.7296543  0.4516924
##   0.045833333  0.7274374  0.4467865
##   0.047916667  0.7246700  0.4410076
##   0.050000000  0.7231124  0.4376607
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03333333.
```

The cp is 0.033 and we get the following decision tree with this:



The variable importance for this method is:

```
## rpart variable importance
```

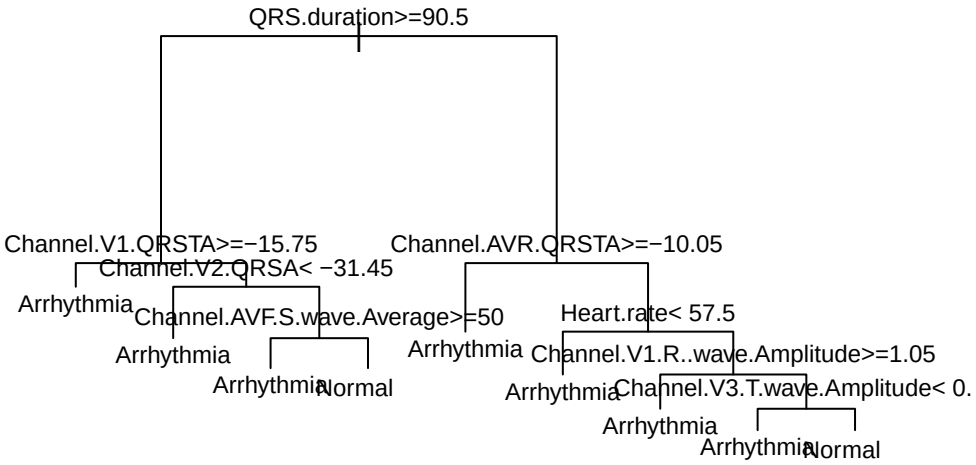
```
##
##   only 20 most important variables shown (out of 256)
##
##                                           Overall
## Channel.V1.Number.of.intrinsic.deflections   100.0
## Channel.AVR.QRSTA                             92.1
## Channel.V6.T.wave.Amplitude                   69.2
## QRS.duration                                  63.6
## Channel.V1.QRSA                               53.8
## Channel.V1.R..wave.Amplitude                  49.4
## Channel.DI.T.wave.Amplitude                   46.6
## Channel.V3.T.wave.Amplitude                   41.7
## Channel.V1.R..wave.Average                    34.5
## Channel.AVF.S.wave.Average                    33.2
## Channel.DI.QRSTA                              31.4
## Heart.rate                                    27.5
## Channel.AVR.T.wave.Amplitude                  27.1
## Channel.V1.QRSTA                              18.7
## Channel.DII.S.wave.Average                    17.5
## Channel.V6.JJ.wave.Amplitude                  17.2
## Height                                        16.9
## Channel.V2.QRSA                               16.9
## Channel.V3.R.wave.Average                     16.0
## Channel.V2.R.wave.Average                     15.6
```

These are predictors are present in the tree too, not all of them because the tree shows less than 20 predictors, and the overall does not show all of them, only 30 are rated, and it is not directly related with the order in the decision tree.

Now let's see how well it is the prediction in the validation set:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   Arrhythmia Normal
##   Arrhythmia         30      8
##   Normal             14     41
##
##                Accuracy : 0.763
##                  95% CI : (0.664, 0.845)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 2.23e-06
##
##                   Kappa : 0.522
##
##  Mcnemar's Test P-Value : 0.286
##
##             Sensitivity : 0.682
##             Specificity : 0.837
##          Pos Pred Value : 0.789
##          Neg Pred Value : 0.745
##              Prevalence : 0.473
##          Detection Rate : 0.323
##    Detection Prevalence : 0.409
##       Balanced Accuracy : 0.759
##
##        'Positive' Class : Arrhythmia
##
```

Let's prune this, using the previous result but applying a new cp value (cp = 0.01) to get a better accuracy and see how it performs. The new decision tree is:




```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Arrhythmia Normal
##   Arrhythmia         32      9
##   Normal             12     40
##
##                Accuracy : 0.774
##                  95% CI : (0.676, 0.854)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 7.38e-07
##
##                   Kappa : 0.545
##
##  Mcnemar's Test P-Value : 0.663
##
##             Sensitivity : 0.727
##             Specificity : 0.816
##          Pos Pred Value : 0.780
##          Neg Pred Value : 0.769
##              Prevalence : 0.473
##          Detection Rate : 0.344
##    Detection Prevalence : 0.441
##       Balanced Accuracy : 0.772
##
##        'Positive' Class : Arrhythmia
##
```

We can not see any improvement, but the decision tree is similar but not equal under the change made by prune. The decision tree graph is different because the library can not manage the output from prune.

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##            | Actually
##  Predicted |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |    9 |   10 |   14 |   15 |   16 | RowT |
## -----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
## Arrhythmia |    9 |    7 |    3 |    3 |    1 |    4 |    0 |    0 |    2 |    9 |    1 |    1 |    1 |   41 |
##            | 0.22 | 0.17 | 0.07 | 0.07 | 0.02 | 0.10 | 0.00 | 0.00 | 0.05 | 0.22 | 0.02 | 0.02 | 0.02 | 0.44 |
##            | 0.18 | 0.78 | 1.00 | 1.00 | 0.33 | 0.80 | 0.00 | 0.00 | 1.00 | 0.90 | 1.00 | 1.00 | 0.20 |      |
## -----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##     Normal |   40 |    2 |    0 |    0 |    2 |    1 |    1 |    1 |    0 |    1 |    0 |    0 |    4 |   52 |
##            | 0.77 | 0.04 | 0.00 | 0.00 | 0.04 | 0.02 | 0.02 | 0.02 | 0.00 | 0.02 | 0.00 | 0.00 | 0.08 | 0.56 |
##            | 0.82 | 0.22 | 0.00 | 0.00 | 0.67 | 0.20 | 1.00 | 1.00 | 0.00 | 0.10 | 0.00 | 0.00 | 0.80 |      |
## -----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##    ColumnT |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
##            | 0.53 | 0.10 | 0.03 | 0.03 | 0.03 | 0.05 | 0.01 | 0.01 | 0.02 | 0.11 | 0.01 | 0.01 | 0.05 |      |
## -----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
```

```
##
##
```

For our record, we keep this result:

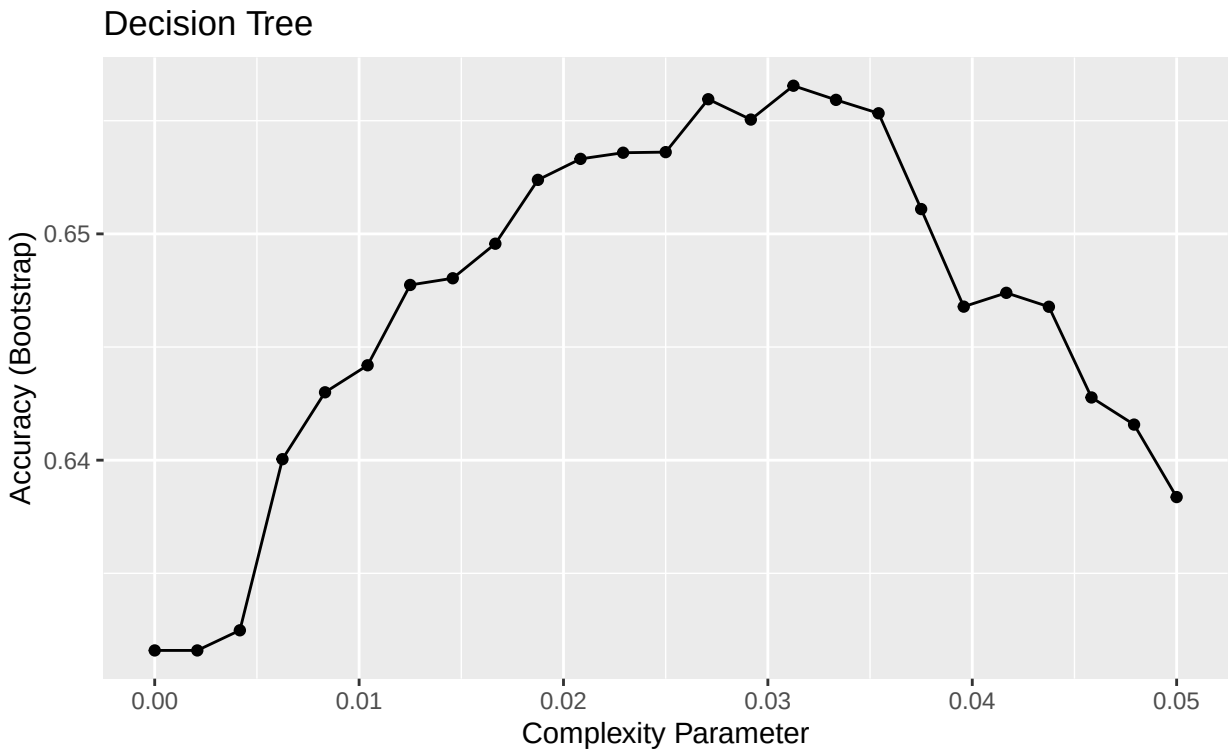
Table 23: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Just the most common	0.5268817	0.0000000	1.0000000
K-Nearest Neighbors	0.6021505	0.2045455	0.9591837
Decision Tree Classifier	0.7741935	0.7272727	0.8163265

We see improvements in Accuracy and Sensitivity and a worse value for Specificity that it is not crucial for this case. The prune applied in this model does not give any improvement in our results.

Second Predictions

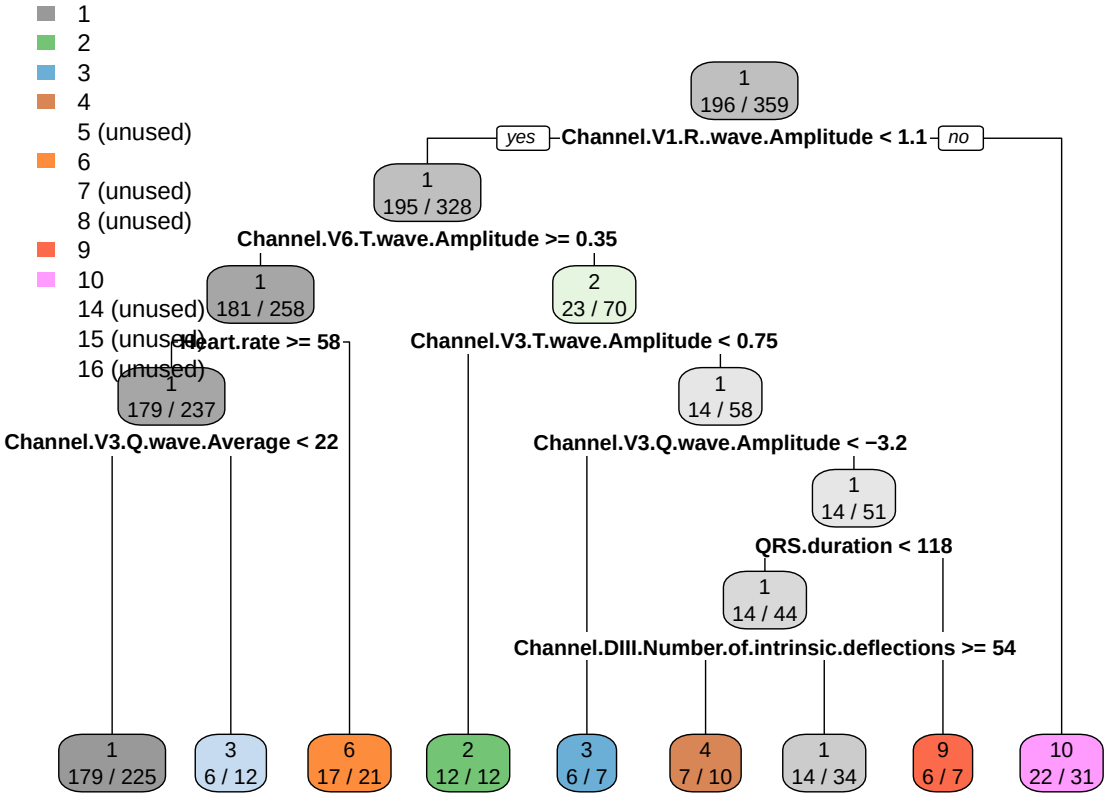
Let's works with the second Prediction with Decision Tree. Again, look for the cp:




```
## CART
##
## 359 samples
## 256 predictors
##  13 classes: '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '14', '15', '16'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.000000000  0.6315934  0.4240513
##   0.002083333  0.6315934  0.4240513
##   0.004166667  0.6324826  0.4251222
##   0.006250000  0.6400443  0.4334626
##   0.008333333  0.6429994  0.4312035
```

```
##   0.010416667  0.6441934  0.4326012
##   0.012500000  0.6477426  0.4354685
##   0.014583333  0.6480389  0.4352358
##   0.016666667  0.6495656  0.4361196
##   0.018750000  0.6523863  0.4388915
##   0.020833333  0.6533138  0.4395617
##   0.022916667  0.6535858  0.4363060
##   0.025000000  0.6536123  0.4327150
##   0.027083333  0.6559501  0.4343033
##   0.029166667  0.6550511  0.4312384
##   0.031250000  0.6565467  0.4294525
##   0.033333333  0.6559214  0.4280107
##   0.035416667  0.6553288  0.4262944
##   0.037500000  0.6510959  0.4228351
##   0.039583333  0.6467841  0.4083274
##   0.041666667  0.6473949  0.4084438
##   0.043750000  0.6467795  0.4070646
##   0.045833333  0.6427714  0.3986225
##   0.047916667  0.6415727  0.3960745
##   0.050000000  0.6383697  0.3853967
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03125.
```

We get a cp = 0.0312 and the following decision tree was created with this train dataset:



Again, let's see the predictors which this model consider that help more:

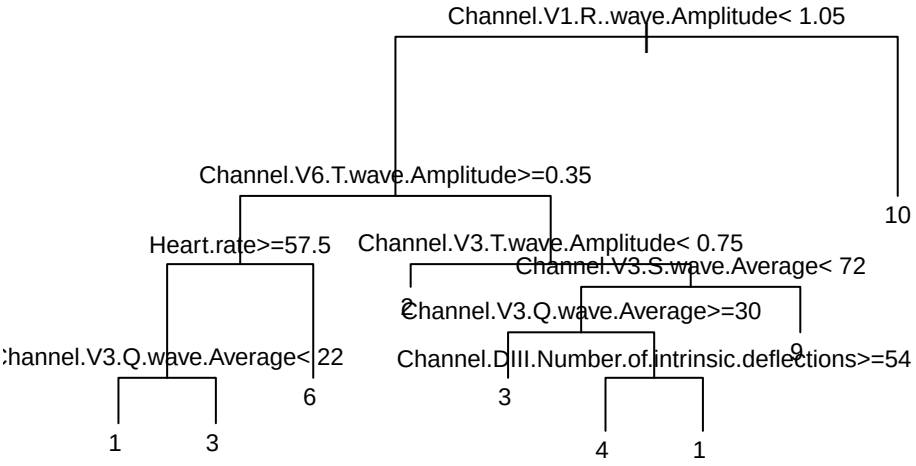
```
## rpart variable importance
##
##   only 20 most important variables shown (out of 256)
##
##                                        Overall
## Heart.rate                              100.00
## Channel.V6.T.wave.Amplitude              56.52
## Channel.V1.QRSA                          52.35
## Channel.V3.Q.wave.Average                46.34
## Channel.V3.Q.wave.Amplitude              35.81
## Channel.V1.R..wave.Amplitude             30.65
## Channel.V3.R.wave.Average                29.65
## Channel.V1.R..wave.Average               27.52
## Channel.AVR.T.wave.Amplitude             26.39
```

```
## Channel.DI.T.wave.Amplitude                 26.05
## Channel.V5.T.wave.Amplitude                 24.18
## Q.T.interval                                11.24
## Channel.V3.T.wave.Amplitude                 11.11
## Channel.V4.T.wave.Amplitude                  9.76
## Channel.V3.S.wave.Amplitude                  8.89
## QRS.duration                                 8.52
## Channel.V5.R.wave.Average                    8.52
## Channel.V6.R.wave.Average                    8.52
## Channel.V3.S.wave.Average                    8.35
## Channel.V3.Number.of.intrinsic.deflections   8.35
```

This trained model, now is applied to the validation set. How well does it perform?

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 14 15 16
##         1  47  4  0  2  3  1  1  1  0  3  1  1  5
##         2   0  4  0  0  0  0  0  0  0  0  0  0  0
##         3   0  1  3  0  0  0  0  0  0  0  0  0  0
##         4   0  0  0  1  0  0  0  0  0  0  0  0  0
##         5   0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   1  0  0  0  0  4  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0  2  0  0  0  0
##         10  1  0  0  0  0  0  0  0  0  7  0  0  0
##         14  0  0  0  0  0  0  0  0  0  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.731
##                  95% CI : (0.629, 0.818)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 4.4e-05
##
##                   Kappa : 0.545
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity             0.959   0.4444   1.0000   0.3333   0.0000   0.8000
## Specificity             0.500   1.0000   0.9889   1.0000   1.0000   0.9886
## Pos Pred Value          0.681   1.0000   0.7500   1.0000      NaN   0.8000
## Neg Pred Value          0.917   0.9438   1.0000   0.9783   0.9677   0.9886
## Prevalence              0.527   0.0968   0.0323   0.0323   0.0323   0.0538
## Detection Rate          0.505   0.0430   0.0323   0.0108   0.0000   0.0430
## Detection Prevalence    0.742   0.0430   0.0430   0.0108   0.0000   0.0538
## Balanced Accuracy       0.730   0.7222   0.9944   0.6667   0.5000   0.8943
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 14 Class: 15
## Sensitivity            0.0000   0.0000   1.0000    0.7000    0.0000    0.0000
## Specificity            1.0000   1.0000   1.0000    0.9880    1.0000    1.0000
## Pos Pred Value            NaN      NaN   1.0000    0.8750       NaN       NaN
## Neg Pred Value         0.9892   0.9892   1.0000    0.9647    0.9892    0.9892
## Prevalence             0.0108   0.0108   0.0215    0.1075    0.0108    0.0108
## Detection Rate         0.0000   0.0000   0.0215    0.0753    0.0000    0.0000
## Detection Prevalence   0.0000   0.0000   0.0215    0.0860    0.0000    0.0000
## Balanced Accuracy      0.5000   0.5000   1.0000    0.8440    0.5000    0.5000
##                      Class: 16
## Sensitivity             0.0000
## Specificity             1.0000
## Pos Pred Value             NaN
## Neg Pred Value          0.9462
## Prevalence              0.0538
## Detection Rate          0.0000
## Detection Prevalence    0.0000
## Balanced Accuracy       0.5000
```

Let's prune this, using the previous result but applying a new cp value (cp = 0.01) to get a better accuracy and see if the prediction improve or not:




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 14 15 16
##         1  47  4  0  2  3  1  1  1  0  3  1  1  5
##         2   0  4  0  0  0  0  0  0  0  0  0  0  0
##         3   0  1  3  0  0  0  0  0  0  0  0  0  0
##         4   0  0  0  1  0  0  0  0  0  0  0  0  0
##         5   0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   1  0  0  0  0  4  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0  2  0  0  0  0
##         10  1  0  0  0  0  0  0  0  0  7  0  0  0
##         14  0  0  0  0  0  0  0  0  0  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.731
##                  95% CI : (0.629, 0.818)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 4.4e-05
##
##                   Kappa : 0.545
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity             0.959   0.4444   1.0000   0.3333   0.0000   0.8000
## Specificity             0.500   1.0000   0.9889   1.0000   1.0000   0.9886
## Pos Pred Value          0.681   1.0000   0.7500   1.0000      NaN   0.8000
## Neg Pred Value          0.917   0.9438   1.0000   0.9783   0.9677   0.9886
## Prevalence              0.527   0.0968   0.0323   0.0323   0.0323   0.0538
## Detection Rate          0.505   0.0430   0.0323   0.0108   0.0000   0.0430
## Detection Prevalence    0.742   0.0430   0.0430   0.0108   0.0000   0.0538
## Balanced Accuracy       0.730   0.7222   0.9944   0.6667   0.5000   0.8943
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 14 Class: 15
## Sensitivity            0.0000   0.0000   1.0000    0.7000    0.0000    0.0000
## Specificity            1.0000   1.0000   1.0000    0.9880    1.0000    1.0000
## Pos Pred Value            NaN      NaN   1.0000    0.8750       NaN       NaN
## Neg Pred Value         0.9892   0.9892   1.0000    0.9647    0.9892    0.9892
## Prevalence             0.0108   0.0108   0.0215    0.1075    0.0108    0.0108
## Detection Rate         0.0000   0.0000   0.0215    0.0753    0.0000    0.0000
## Detection Prevalence   0.0000   0.0000   0.0215    0.0860    0.0000    0.0000
## Balanced Accuracy      0.5000   0.5000   1.0000    0.8440    0.5000    0.5000
##                      Class: 16
## Sensitivity             0.0000
## Specificity             1.0000
## Pos Pred Value             NaN
## Neg Pred Value          0.9462
## Prevalence              0.0538
## Detection Rate          0.0000
## Detection Prevalence    0.0000
## Balanced Accuracy       0.5000
```

We can not see any improvement neither, but the decision tree is similar but not equal under the change made by prune.

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##            | Actually
## Predicted |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |    9 |   10 |   14 |   15 |   16 | RowT |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         1 |   47 |    4 |    0 |    2 |    3 |    1 |    1 |    1 |    0 |    3 |    1 |    1 |    5 |   69 |
##           | 0.68 | 0.06 | 0.00 | 0.03 | 0.04 | 0.01 | 0.01 | 0.01 | 0.00 | 0.04 | 0.01 | 0.01 | 0.07 | 0.74 |
##           | 0.96 | 0.44 | 0.00 | 0.67 | 1.00 | 0.20 | 1.00 | 1.00 | 0.00 | 0.30 | 1.00 | 1.00 | 1.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         2 |    0 |    4 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    4 |
##           | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 |
##           | 0.00 | 0.44 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         3 |    0 |    1 |    3 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    4 |
##           | 0.00 | 0.25 | 0.75 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 |
##           | 0.00 | 0.11 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         4 |    0 |    0 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    1 |
##           | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 |
##           | 0.00 | 0.00 | 0.00 | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         6 |    1 |    0 |    0 |    0 |    0 |    4 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    5 |
##           | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 |
##           | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         9 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    2 |    0 |    0 |    0 |    0 |    2 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##        10 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    7 |    0 |    0 |    0 |    8 |
##           | 0.12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.88 | 0.00 | 0.00 | 0.00 | 0.09 |
##           | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.70 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##   ColumnT |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
##           | 0.53 | 0.10 | 0.03 | 0.03 | 0.03 | 0.05 | 0.01 | 0.01 | 0.02 | 0.11 | 0.01 | 0.01 | 0.05 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##
##
```

For our record, we keep this result:

Table 24: Prediction All Arrhytmia Classification Summary

Model	Just the most common	K-Nearest Neighbors	Decision Tree Classifier
Accuracy	0.5268817	0.6236559	0.7311828
Code.1	Class: 1	Class: 1	Class: 1
Sensitivity.1	1.0000000	1.0000000	0.9591837
Specificity.1	0.0000000	0.2045455	0.5000000
Code.2	Class: 2	Class: 2	Class: 2
Sensitivity.2	0.0000000	0.1111111	0.4444444
Specificity.2	1	1	1
Code.3	Class: 3	Class: 3	Class: 3
Sensitivity.3	0	1	1
Specificity.3	1.0000000	1.0000000	0.9888889
Code.4	Class: 4	Class: 4	Class: 4
Sensitivity.4	0.0000000	0.0000000	0.3333333
Specificity.4	1	1	1
Code.5	Class: 5	Class: 5	Class: 5

Table 24: Prediction All Arrhytmia Classification Summary

Sensitivity.5	0	0	0
Specificity.5	1	1	1
Code.6	Class: 6	Class: 6	Class: 6
Sensitivity.6	0.0	0.2	0.8
Specificity.6	1.0000000	1.0000000	0.9886364
Code.7	Class: 7	Class: 7	Class: 7
Sensitivity.7	0	0	0
Specificity.7	1	1	1
Code.8	Class: 8	Class: 8	Class: 8
Sensitivity.8	0	0	0
Specificity.8	1	1	1
Code.9	Class: 9	Class: 9	Class: 9
Sensitivity.9	0	1	1
Specificity.9	1	1	1
Code.10	Class: 10	Class: 10	Class: 10
Sensitivity.10	0.0	0.2	0.7
Specificity.10	1.0000000	1.0000000	0.9879518
Code.11	Class: 14	Class: 14	Class: 14
Sensitivity.11	0	0	0
Specificity.11	1	1	1
Code.12	Class: 15	Class: 15	Class: 15
Sensitivity.12	0	0	0
Specificity.12	1	1	1
Code.13	Class: 16	Class: 16	Class: 16
Sensitivity.13	0	0	0
Specificity.13	1	1	1

We see a good improvements in Accuracy. The prune applied in this model does not give any improvement in our results. In 5 classes we can not predict any case (Sensitivity =0 & Specificity=1) and Class 9 we have all the cases right!

Random Forests Model

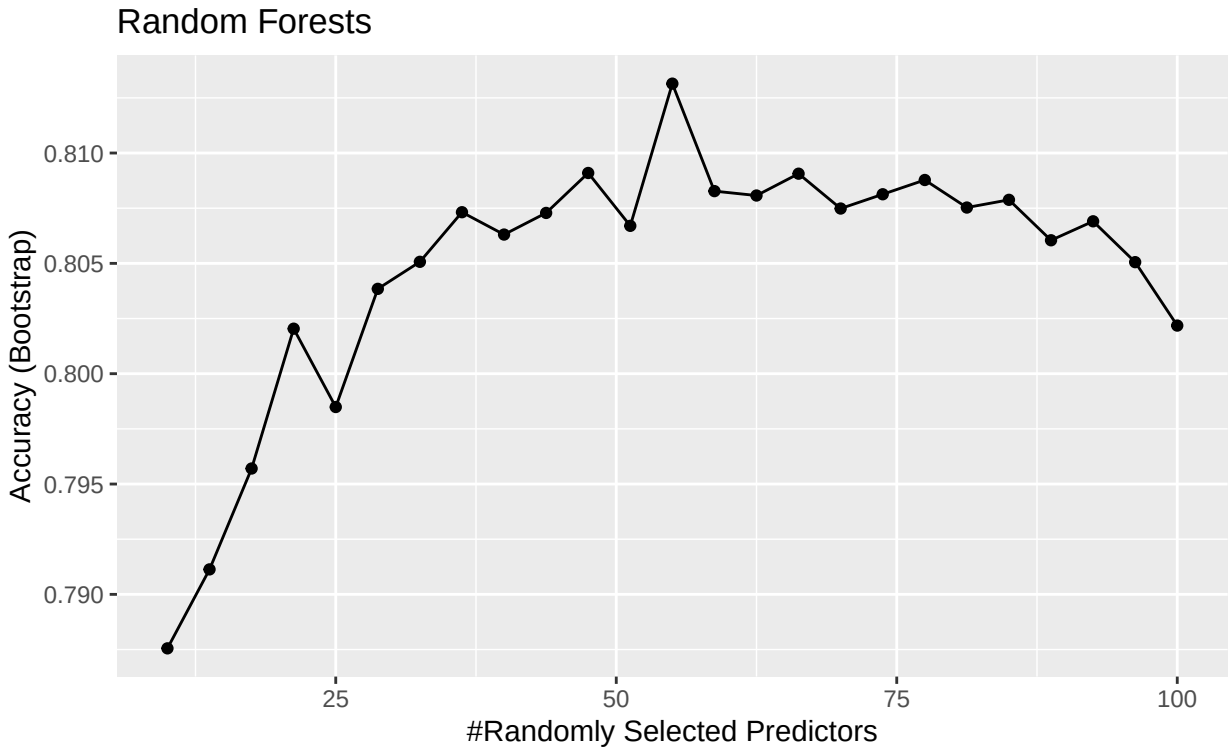
We see one decison tree in the previous section. Applying prune we have a different one but with similar accuracy. The idea behind random forest is that we created multiple trees and the class who get more "votes" among all the tress is the predicted value.

In other words, among the tress, witch prediction is the most popular, wins.

Here we have like parameters the value of mtry that we are trying to use to get the better accuracy.

First Predictions

We start look for mtry to get the better accuracy in this model fro Normal/Arrhythmia prediction:




```
## Random Forest
##
## 359 samples
## 256 predictors
##   2 classes: 'Arrhythmia', 'Normal'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   mtry    Accuracy   Kappa
##    10.00  0.7875539  0.5659593
##    13.75  0.7911314  0.5746985
##    17.50  0.7957031  0.5844496
##    21.25  0.8020399  0.5974888
##    25.00  0.7984903  0.5908309
##    28.75  0.8038453  0.6021540
##    32.50  0.8050707  0.6055532
##    36.25  0.8073178  0.6092809
##    40.00  0.8063051  0.6074431
##    43.75  0.8072858  0.6097125
##    47.50  0.8090962  0.6133771
##    51.25  0.8067009  0.6086614
##    55.00  0.8131449  0.6213730
##    58.75  0.8082740  0.6120482
##    62.50  0.8080747  0.6116935
##    66.25  0.8090640  0.6134446
##    70.00  0.8074850  0.6107539
##    73.75  0.8081312  0.6117353
##    77.50  0.8087783  0.6129704
##    81.25  0.8075298  0.6105329
##    85.00  0.8078797  0.6114404
##    88.75  0.8060468  0.6075933
##    92.50  0.8069058  0.6095287
##    96.25  0.8050554  0.6057375
##   100.00  0.8021813  0.6000891
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 55.
```

the mtry we get is 55. The variable importance for this method are:

```
## rf variable importance
##
```

```
##   only 20 most important variables shown (out of 256)
##
##                                                Overall
## `Heart rate`                                     100.0
## `QRS duration`                                    67.9
## `Channel V1 Number of intrinsic deflections`      45.8
## `Channel V1 R' wave Amplitude`                    43.9
## `Channel V1 QRSA`                                 43.7
## `Channel V6 T wave Amplitude`                     41.5
## `Channel DI T wave Amplitude`                     32.1
## `Channel AVR QRSTA`                               25.4
## `Channel V1 R' wave Average`                      25.1
## `Channel AVR T wave Amplitude`                    24.8
## `Channel DII QRSTA`                               22.5
## `Channel DI QRSTA`                                21.8
## `T interval`                                      17.4
## `Channel V2 S wave Amplitude`                     17.2
## `Q-T interval`                                    15.3
## `Channel V3 QRSTA`                                15.3
## `Channel V6 QRSTA`                                15.1
## `Channel V5 JJ wave Amplitude`                    15.0
## `Channel AVL Number of intrinsic deflections`     14.5
## `Channel V2 R' wave Amplitude`                    13.9
```

Now see how well it is the prediction in the validation set:

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Arrhythmia Normal
##   Arrhythmia         35      5
##   Normal              9     44
##
##                Accuracy : 0.849
##                  95% CI : (0.76, 0.915)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 5.35e-11
##
##                   Kappa : 0.697
##
##  Mcnemar's Test P-Value : 0.423
##
##             Sensitivity : 0.795
##             Specificity : 0.898
##          Pos Pred Value : 0.875
##          Neg Pred Value : 0.830
##              Prevalence : 0.473
##          Detection Rate : 0.376
##    Detection Prevalence : 0.430
##       Balanced Accuracy : 0.847
##
##        'Positive' Class : Arrhythmia
##

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##              | Actually
##    Predicted | Arrhythmia |     Normal |  Row Total |
## -------------|------------|------------|------------|
##   Arrhythmia |         35 |          5 |         40 |
##              |       0.88 |       0.12 |       0.43 |
##              |       0.80 |       0.10 |            |
## -------------|------------|------------|------------|
##       Normal |          9 |         44 |         53 |
##              |       0.17 |       0.83 |       0.57 |
##              |       0.20 |       0.90 |            |
## -------------|------------|------------|------------|
## Column Total |         44 |         49 |         93 |
##              |       0.47 |       0.53 |            |
## -------------|------------|------------|------------|
```

```
##
##
```

For our record, we keep this result:

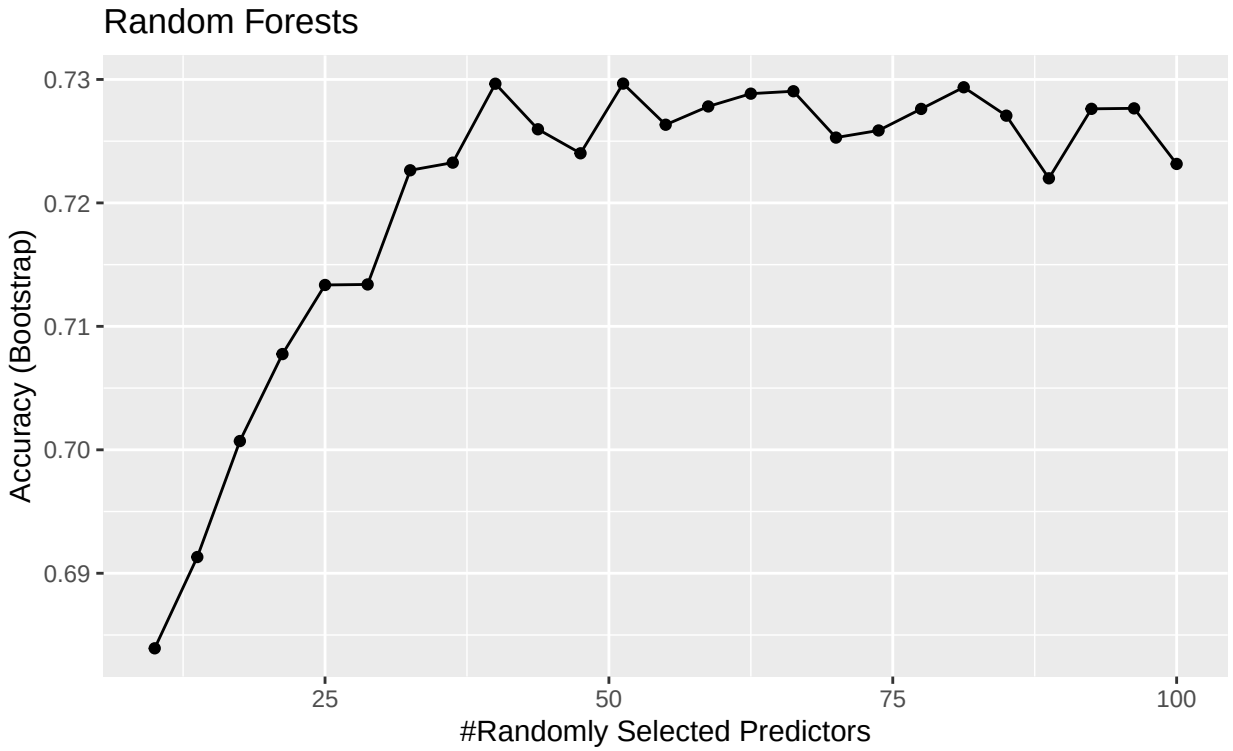
Table 25: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Just the most common	0.5268817	0.0000000	1.0000000
K-Nearest Neighbors	0.6021505	0.2045455	0.9591837
Decision Tree Classifier	0.7741935	0.7272727	0.8163265
Random Forest Classifier	0.8494624	0.7954545	0.8979592

Random Forests get better result over decision tree and previous ones in Accuracy and Sensitivity. Until now if our best predictor.

Second Predictions

Let's start with mtry in random forest, now with all the classes:




```
## Random Forest
##
## 359 samples
## 256 predictors
##  13 classes: '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '14', '15', '16'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   mtry    Accuracy   Kappa
##    10.00  0.6839211  0.4316668
##    13.75  0.6913149  0.4535041
##    17.50  0.7007089  0.4822203
```

```
##    21.25  0.7077561  0.4978313
##    25.00  0.7133526  0.5102739
##    28.75  0.7133953  0.5124898
##    32.50  0.7226439  0.5302680
##    36.25  0.7232619  0.5353494
##    40.00  0.7296540  0.5472532
##    43.75  0.7259659  0.5408133
##    47.50  0.7240193  0.5383796
##    51.25  0.7296644  0.5484682
##    55.00  0.7263317  0.5440184
##    58.75  0.7278143  0.5487114
##    62.50  0.7288498  0.5494765
##    66.25  0.7290454  0.5502940
##    70.00  0.7252924  0.5442965
##    73.75  0.7258682  0.5456165
##    77.50  0.7276140  0.5488429
##    81.25  0.7293669  0.5532431
##    85.00  0.7270665  0.5477733
##    88.75  0.7219938  0.5405467
##    92.50  0.7276208  0.5506111
##    96.25  0.7276598  0.5495607
##   100.00  0.7231572  0.5430519
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 51.25.
```

the mtry we get is 51.25. The variable importance for this method are:

```
## rf variable importance
##
##   only 20 most important variables shown (out of 256)
##
##                                                Overall
## `Heart rate`                                     100.0
## `Channel V1 R' wave Amplitude`                    37.2
## `Channel V1 QRSA`                                 34.7
## `Channel V6 T wave Amplitude`                     33.1
## `Channel V1 Number of intrinsic deflections`      31.8
## `Channel V1 R' wave Average`                      27.5
## `Channel V5 T wave Amplitude`                     26.7
## `QRS duration`                                    21.2
## `Channel AVR T wave Amplitude`                    18.4
## `Channel V4 T wave Amplitude`                     18.3
## `Channel DI T wave Amplitude`                     17.0
## `Q-T interval`                                    16.7
## `Channel V3 Q wave Amplitude`                     13.8
## `Channel V2 R' wave Amplitude`                    13.4
## `Channel AVF Q wave Average`                      12.9
## `Channel V3 Q wave Average`                       12.8
## `Channel V3 R wave Amplitude`                     12.2
## `Channel DII T wave Amplitude`                    11.3
## `Channel AVF Q wave Amplitude`                    11.1
## `Channel V3 R wave Average`                       11.1
```

Now see with the model created using the trining set how well it is the prediction in the validation set:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 14 15 16
##         1  47  1  0  1  3  1  1  1  0  1  0  1  5
##         2   0  7  0  0  0  0  0  0  0  0  1  0  0
##         3   0  0  3  0  0  0  0  0  0  0  0  0  0
##         4   0  1  0  2  0  0  0  0  0  0  0  0  0
##         5   0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   1  0  0  0  0  4  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0  0  2  0  0  0
##         10  1  0  0  0  0  0  0  0  0  0  9  0  0  0
##         14  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.796
##                  95% CI : (0.699, 0.872)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 6.76e-08
##
```

```
##                  Kappa : 0.672
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity             0.959   0.7778   1.0000   0.6667   0.0000   0.8000
## Specificity             0.659   0.9881   1.0000   0.9889   1.0000   0.9886
## Pos Pred Value          0.758   0.8750   1.0000   0.6667      NaN   0.8000
## Neg Pred Value          0.935   0.9765   1.0000   0.9889   0.9677   0.9886
## Prevalence              0.527   0.0968   0.0323   0.0323   0.0323   0.0538
## Detection Rate          0.505   0.0753   0.0323   0.0215   0.0000   0.0430
## Detection Prevalence    0.667   0.0860   0.0323   0.0323   0.0000   0.0538
## Balanced Accuracy       0.809   0.8829   1.0000   0.8278   0.5000   0.8943
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 14 Class: 15
## Sensitivity            0.0000   0.0000   1.0000    0.9000    0.0000    0.0000
## Specificity            1.0000   1.0000   1.0000    0.9880    1.0000    1.0000
## Pos Pred Value            NaN      NaN   1.0000    0.9000       NaN       NaN
## Neg Pred Value         0.9892   0.9892   1.0000    0.9880    0.9892    0.9892
## Prevalence             0.0108   0.0108   0.0215    0.1075    0.0108    0.0108
## Detection Rate         0.0000   0.0000   0.0215    0.0968    0.0000    0.0000
## Detection Prevalence   0.0000   0.0000   0.0215    0.1075    0.0000    0.0000
## Balanced Accuracy      0.5000   0.5000   1.0000    0.9440    0.5000    0.5000
##                      Class: 16
## Sensitivity             0.0000
## Specificity             1.0000
## Pos Pred Value             NaN
## Neg Pred Value          0.9462
## Prevalence              0.0538
## Detection Rate          0.0000
## Detection Prevalence    0.0000
## Balanced Accuracy       0.5000

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##            | Actually
## Predicted |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |    9 |   10 |   14 |   15 |   16 | RowT |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         1 |   47 |    1 |    0 |    1 |    3 |    1 |    1 |    1 |    0 |    1 |    0 |    1 |    5 |   62 |
##           | 0.76 | 0.02 | 0.00 | 0.02 | 0.05 | 0.02 | 0.02 | 0.02 | 0.00 | 0.02 | 0.00 | 0.02 | 0.08 | 0.67 |
##           | 0.96 | 0.11 | 0.00 | 0.33 | 1.00 | 0.20 | 1.00 | 1.00 | 0.00 | 0.10 | 0.00 | 1.00 | 1.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         2 |    0 |    7 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    1 |    0 |    0 |    8 |
##           | 0.00 | 0.88 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.12 | 0.00 | 0.00 | 0.09 |
##           | 0.00 | 0.78 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         3 |    0 |    0 |    3 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    3 |
##           | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 |
##           | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         4 |    0 |    1 |    0 |    2 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    3 |
##           | 0.00 | 0.33 | 0.00 | 0.67 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 |
##           | 0.00 | 0.11 | 0.00 | 0.67 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         6 |    1 |    0 |    0 |    0 |    0 |    4 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    5 |
##           | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 |
##           | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##         9 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    2 |    0 |    0 |    0 |    0 |    2 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
##           | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##        10 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    9 |    0 |    0 |    0 |   10 |
##           | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 | 0.00 | 0.00 | 0.00 | 0.11 |
##           | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 | 0.00 | 0.00 | 0.00 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
##   ColumnT |   49 |    9 |    3 |    3 |    3 |    5 |    1 |    1 |    2 |   10 |    1 |    1 |    5 |   93 |
##           | 0.53 | 0.10 | 0.03 | 0.03 | 0.03 | 0.05 | 0.01 | 0.01 | 0.02 | 0.11 | 0.01 | 0.01 | 0.05 |      |
## ----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
```

##
##

For our record, we keep this result:

Table 26: Prediction All Arrhytmia Classification Summary

Model	Just the most common	K-Nearest Neighbors	Decision Tree Classifier	Random Forest Classifier
Accuracy	0.5268817	0.6236559	0.7311828	0.7956989
Code.1	Class: 1	Class: 1	Class: 1	Class: 1
Sensitivity.1	1.0000000	1.0000000	0.9591837	0.9591837
Specificity.1	0.0000000	0.2045455	0.5000000	0.6590909
Code.2	Class: 2	Class: 2	Class: 2	Class: 2
Sensitivity.2	0.0000000	0.1111111	0.4444444	0.7777778
Specificity.2	1.0000000	1.0000000	1.0000000	0.9880952
Code.3	Class: 3	Class: 3	Class: 3	Class: 3
Sensitivity.3	0	1	1	1
Specificity.3	1.0000000	1.0000000	0.9888889	1.0000000
Code.4	Class: 4	Class: 4	Class: 4	Class: 4
Sensitivity.4	0.0000000	0.0000000	0.3333333	0.6666667
Specificity.4	1.0000000	1.0000000	1.0000000	0.9888889
Code.5	Class: 5	Class: 5	Class: 5	Class: 5
Sensitivity.5	0	0	0	0
Specificity.5	1	1	1	1
Code.6	Class: 6	Class: 6	Class: 6	Class: 6
Sensitivity.6	0.0	0.2	0.8	0.8
Specificity.6	1.0000000	1.0000000	0.9886364	0.9886364
Code.7	Class: 7	Class: 7	Class: 7	Class: 7
Sensitivity.7	0	0	0	0
Specificity.7	1	1	1	1
Code.8	Class: 8	Class: 8	Class: 8	Class: 8
Sensitivity.8	0	0	0	0
Specificity.8	1	1	1	1
Code.9	Class: 9	Class: 9	Class: 9	Class: 9
Sensitivity.9	0	1	1	1
Specificity.9	1	1	1	1
Code.10	Class: 10	Class: 10	Class: 10	Class: 10
Sensitivity.10	0.0	0.2	0.7	0.9
Specificity.10	1.0000000	1.0000000	0.9879518	0.9879518
Code.11	Class: 14	Class: 14	Class: 14	Class: 14
Sensitivity.11	0	0	0	0
Specificity.11	1	1	1	1
Code.12	Class: 15	Class: 15	Class: 15	Class: 15
Sensitivity.12	0	0	0	0
Specificity.12	1	1	1	1
Code.13	Class: 16	Class: 16	Class: 16	Class: 16
Sensitivity.13	0	0	0	0
Specificity.13	1	1	1	1

We see improvements in Accuracy. In 6 classes we can not predict any case (Sensitivity =0 & Specificity=1) and Class 3 and 9 we have all the cases right! One aspect to consider is that this method spend more CPU time over the rest.

Support Vector Machines Model

The Support Vector Machines (SVM), was created as an objective a fast and dependable classification algorithm that performs very well with a limited amount of data to analyze. This create a n-dimensional space, one for each predictor, an try to get the space of the output.

A support vector machine allows you to classify data that's linearly separable, but if it is not linearly separable, you can use the kernel trick to make it work. In this case we let the algorithm to select the kernel and later the parameters cost and gamma.

First Predictions

For this part we are going to use the caret train method that perform better than svm from e1071 library. The last library was useful to determine that the radial kernel is the best solution. In this model, we are not going to use other parameter than the kernel.

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 359 samples
## 256 predictors
##   2 classes: 'Arrhythmia', 'Normal'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 359, 359, 359, 359, 359, 359, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.5232775  0
##   0.50  0.5289918  0
##   1.00  0.5369918  0
##
## Tuning parameter 'sigma' was held constant at a value of 0.003353676
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.003353676 and C = 1.
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Arrhythmia Normal
##   Arrhythmia         28      7
##   Normal             16     42
##
##                Accuracy : 0.753
##                  95% CI : (0.652, 0.836)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 6.37e-06
##
##                   Kappa : 0.499
##
##  Mcnemar's Test P-Value : 0.0953
##
##             Sensitivity : 0.636
##             Specificity : 0.857
##          Pos Pred Value : 0.800
##          Neg Pred Value : 0.724
##              Prevalence : 0.473
##          Detection Rate : 0.301
##    Detection Prevalence : 0.376
##       Balanced Accuracy : 0.747
##
##        'Positive' Class : Arrhythmia
##
```

The values that get the better result are sigma = 0.0034 and C = 1. The SVM-Kernel is radial.

The important variables are:

```
## ROC curve variable importance
##
##   only 20 most important variables shown (out of 256)
##
##                               Importance
## QRS duration                       100.0
## Channel AVR QRSTA                   92.1
## Channel DI QRSTA                    85.6
```

```
## Channel DI T wave Amplitude        82.8
## Channel AVR T wave Amplitude       78.8
## Channel V6 T wave Amplitude        77.8
## Channel V6 QRSTA                   77.3
## Channel V1 QRSTA                   69.6
## Channel DII QRSTA                  68.0
## Channel DII T wave Amplitude       65.4
## Sex                                64.2
## Channel AVL T wave Amplitude       62.6
## Channel V3 QRSTA                   60.1
## Channel AVR R wave Average         59.6
## Channel V6 JJ wave Amplitude       58.6
## Channel V4 S wave Average          58.3
## Channel V5 T wave Amplitude        58.0
## Channel V5 QRSTA                   54.8
## Channel V4 QRSTA                   53.1
## Channel AVR QRSA                   52.9
```

The try to get a tune over the SVM but it fail, for that we keep the original value.

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##              | Actually
##    Predicted | Arrhythmia |     Normal |  Row Total |
## -------------|------------|------------|------------|
##   Arrhythmia |         28 |          7 |         35 |
##              |      0.800 |      0.200 |      0.376 |
##              |      0.636 |      0.143 |            |
## -------------|------------|------------|------------|
##       Normal |         16 |         42 |         58 |
##              |      0.276 |      0.724 |      0.624 |
##              |      0.364 |      0.857 |            |
## -------------|------------|------------|------------|
## Column Total |         44 |         49 |         93 |
##              |      0.473 |      0.527 |            |
## -------------|------------|------------|------------|
##
##
```

For our record, we keep this result:

Table 27: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Just the most common	0.5268817	0.0000000	1.0000000
K-Nearest Neighbors	0.6021505	0.2045455	0.9591837
Decision Tree Classifier	0.7741935	0.7272727	0.8163265
Random Forest Classifier	0.8494624	0.7954545	0.8979592
SVM Classifier	0.7526882	0.6363636	0.8571429

This method did not perform as expected and it is close to KNN and decison tree in result.

Second Predictions

Again, taking in consideration all the classes, we are going to use radial kernel.

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 359 samples
## 256 predictors
##  13 classes: '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '14', '15', '16'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 322, 324, 324, 323, 324, 324, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.5464454  0.00000000
##   0.50  0.5464454  0.00000000
##   1.00  0.5693025  0.07265032
##
## Tuning parameter 'sigma' was held constant at a value of 0.003353676
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.003353676 and C = 1.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5  6  7  8  9 10 14 15 16
##         1  48  7  0  2  3  5  1  1  2  5  1  1  5
##         2   0  2  0  0  0  0  0  0  0  0  0  0  0
##         3   0  0  3  0  0  0  0  0  0  0  0  0  0
##         4   0  0  0  1  0  0  0  0  0  0  0  0  0
##         5   0  0  0  0  0  0  0  0  0  0  0  0  0
##         6   0  0  0  0  0  0  0  0  0  0  0  0  0
##         7   0  0  0  0  0  0  0  0  0  0  0  0  0
##         8   0  0  0  0  0  0  0  0  0  0  0  0  0
##         9   0  0  0  0  0  0  0  0  0  0  0  0  0
##         10  1  0  0  0  0  0  0  0  0  5  0  0  0
##         14  0  0  0  0  0  0  0  0  0  0  0  0  0
##         15  0  0  0  0  0  0  0  0  0  0  0  0  0
##         16  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.634
##                  95% CI : (0.528, 0.732)
##     No Information Rate : 0.527
##     P-Value [Acc > NIR] : 0.0236
##
##                   Kappa : 0.311
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity             0.980   0.2222   1.0000   0.3333   0.0000   0.0000
## Specificity             0.250   1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value          0.593   1.0000   1.0000   1.0000      NaN      NaN
## Neg Pred Value          0.917   0.9231   1.0000   0.9783   0.9677   0.9462
## Prevalence              0.527   0.0968   0.0323   0.0323   0.0323   0.0538
## Detection Rate          0.516   0.0215   0.0323   0.0108   0.0000   0.0000
## Detection Prevalence    0.871   0.0215   0.0323   0.0108   0.0000   0.0000
## Balanced Accuracy       0.615   0.6111   1.0000   0.6667   0.5000   0.5000
##                      Class: 7 Class: 8 Class: 9 Class: 10 Class: 14 Class: 15
## Sensitivity            0.0000   0.0000   0.0000    0.5000    0.0000    0.0000
## Specificity            1.0000   1.0000   1.0000    0.9880    1.0000    1.0000
## Pos Pred Value            NaN      NaN      NaN    0.8333       NaN       NaN
## Neg Pred Value         0.9892   0.9892   0.9785    0.9425    0.9892    0.9892
## Prevalence             0.0108   0.0108   0.0215    0.1075    0.0108    0.0108
## Detection Rate         0.0000   0.0000   0.0000    0.0538    0.0000    0.0000
## Detection Prevalence   0.0000   0.0000   0.0000    0.0645    0.0000    0.0000
## Balanced Accuracy      0.5000   0.5000   0.5000    0.7440    0.5000    0.5000
##                      Class: 16
## Sensitivity             0.0000
## Specificity             1.0000
## Pos Pred Value             NaN
## Neg Pred Value          0.9462
## Prevalence              0.0538
## Detection Rate          0.0000
## Detection Prevalence    0.0000
## Balanced Accuracy       0.5000
```

Note about the code: at the beginning we see that svm from e1071 library works better for multiple output than the caret train method, but a little more investigation shows that with trainControl() we can improve the training with the caret library and then continue using this library as we did with other models.

The SVM-Kernel is radial.

The important variables are:

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##   only 20 most important variables shown (out of 256)
##
##                                  X1    X2    X3    X4   X5     X6     X7    X8
## Channel V6 QRSTA              67.98 85.33 59.04 59.04 59.0 100.00  59.04 59.04
## Channel V2 QRSA               62.03 25.93 38.21 25.93 25.9  25.93 100.00 39.74
## Channel AVR QRSA              62.46 38.73 40.41 14.87 87.0 100.00  52.77 45.33
## Channel V3 Q wave Amplitude  100.00  2.04  7.96  8.06 49.0   2.04   2.04  2.04
## Channel DII S wave Amplitude  28.83 20.41 25.15  5.98 49.7 100.00  23.25 54.77
## Channel AVR R wave Average    53.32 18.03 52.91  1.43 67.1 100.00  39.29 57.70
## Channel DII QRSA              43.62 56.55 20.10  6.25 92.9 100.00  22.01 27.74
## QRST Vector angles            11.35 33.12 43.83  3.56 86.2 100.00  19.83 16.59
## Channel V1 JJ wave Amplitude  27.97 27.97 57.19 27.97 28.0  86.22 100.00 27.97
## QRS Vector angles              2.89 45.07 29.49  2.06 90.8 100.00  40.67 11.56
## Channel DII QRSTA             63.69 74.45 46.94 46.94 74.2 100.00  46.94 46.94
## Channel V1 S wave Average     64.88  6.97 34.13  6.97 29.6 100.00  21.28 58.24
## Channel V6 S wave Amplitude   17.26 26.96 52.09 29.44 64.0 100.00  12.61 42.10
## Channel V6 QRSA               56.04 54.08 56.28 15.05 52.0 100.00  86.08 20.88
## Channel AVF QRSA              23.47 44.56 12.40  3.21 89.3 100.00  31.71 12.30
## Channel AVF S wave Amplitude  37.50 36.14  1.73  2.93 26.5 100.00  69.02 39.29
## QRS duration                  51.28 40.66 40.66 40.66 70.4  96.43 100.00 56.34
## Channel V3 Q wave Average    100.00  2.04  8.16  8.11 49.0   2.04   2.04  2.04
## Channel V3 S wave Average     84.95 39.42 39.42 39.42 39.4  96.94 100.00 39.42
## Channel V5 QRSA               73.68 30.44 49.85 36.02 36.7 100.00  39.87  8.31
##                                  X9   X10   X14    X15    X16
## Channel V6 QRSTA              59.04 80.87 59.04  59.04  85.33
## Channel V2 QRSA               80.78 36.48 40.22  42.86  62.03
## Channel AVR QRSA              22.79 23.34 11.95  65.95  62.46
## Channel V3 Q wave Amplitude   31.29 23.47 15.76 100.00 100.00
## Channel DII S wave Amplitude  42.18 30.87  5.98  31.90  28.83
## Channel AVR R wave Average    25.34  1.79 12.97  55.95  53.32
## Channel DII QRSA              57.65 14.29  6.25  44.05  56.55
## QRST Vector angles            85.20  3.56 21.40  12.38  33.12
## Channel V1 JJ wave Amplitude 38.44 52.42 33.82  27.97   9.61
## QRS Vector angles             94.90 30.61 15.94   3.57  45.07
## Channel DII QRSTA             71.77 79.97 46.94  46.94  74.45
## Channel V1 S wave Average      6.97 29.21  6.97  68.33  64.88
## Channel V6 S wave Amplitude   38.78 17.60 20.53  25.71  26.96
## Channel V6 QRSA               31.80 13.32 13.32  61.67  56.04
## Channel AVF QRSA              76.53  3.21  3.21  22.86  44.56
## Channel AVF S wave Amplitude  1.70 46.94  3.54  34.52  37.50
## QRS duration                  81.46 40.66 45.77  40.66  51.28
## Channel V3 Q wave Average     31.97 23.47 15.79 100.00 100.00
## Channel V3 S wave Average     60.03 39.42 39.42  89.52  84.95
## Channel V5 QRSA               53.23  8.31  8.31  70.24  73.68

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
##
##              | Actually
##    Predicted | Arrhythmia |     Normal |  Row Total |
## -------------|------------|------------|------------|
##            1 |         33 |         48 |         81 |
##              |      0.407 |      0.593 |      0.871 |
##              |      0.750 |      0.980 |            |
## -------------|------------|------------|------------|
##            2 |          2 |          0 |          2 |
##              |      1.000 |      0.000 |      0.022 |
##              |      0.045 |      0.000 |            |
## -------------|------------|------------|------------|
##            3 |          3 |          0 |          3 |
##              |      1.000 |      0.000 |      0.032 |
##              |      0.068 |      0.000 |            |
## -------------|------------|------------|------------|
##            4 |          1 |          0 |          1 |
##              |      1.000 |      0.000 |      0.011 |
##              |      0.023 |      0.000 |            |
```

```
## -------------|------------|------------|------------|
##           10 |          5 |          1 |          6 |
##              |      0.833 |      0.167 |      0.065 |
##              |      0.114 |      0.020 |            |
## -------------|------------|------------|------------|
## Column Total |         44 |         49 |         93 |
##              |      0.473 |      0.527 |            |
## -------------|------------|------------|------------|
##
##
```

For our record, we keep this result:

Table 28: Prediction All Arrhytmia Classification Summary

Model	Just the most common	K-Nearest Neighbors	Decision Tree Classifier	Random Forest Classifier	SVM Classifier
Accuracy	0.5268817	0.6236559	0.7311828	0.7956989	0.6344086
Code.1	Class: 1	Class: 1	Class: 1	Class: 1	Class: 1
Sensitivity.1	1.0000000	1.0000000	0.9591837	0.9591837	0.9795918
Specificity.1	0.0000000	0.2045455	0.5000000	0.6590909	0.2500000
Code.2	Class: 2	Class: 2	Class: 2	Class: 2	Class: 2
Sensitivity.2	0.0000000	0.1111111	0.4444444	0.7777778	0.2222222
Specificity.2	1.0000000	1.0000000	1.0000000	0.9880952	1.0000000
Code.3	Class: 3	Class: 3	Class: 3	Class: 3	Class: 3
Sensitivity.3	0	1	1	1	1
Specificity.3	1.0000000	1.0000000	0.9888889	1.0000000	1.0000000
Code.4	Class: 4	Class: 4	Class: 4	Class: 4	Class: 4
Sensitivity.4	0.0000000	0.0000000	0.3333333	0.6666667	0.3333333
Specificity.4	1.0000000	1.0000000	1.0000000	0.9888889	1.0000000
Code.5	Class: 5	Class: 5	Class: 5	Class: 5	Class: 5
Sensitivity.5	0	0	0	0	0
Specificity.5	1	1	1	1	1
Code.6	Class: 6	Class: 6	Class: 6	Class: 6	Class: 6
Sensitivity.6	0.0	0.2	0.8	0.8	0.0
Specificity.6	1.0000000	1.0000000	0.9886364	0.9886364	1.0000000
Code.7	Class: 7	Class: 7	Class: 7	Class: 7	Class: 7
Sensitivity.7	0	0	0	0	0
Specificity.7	1	1	1	1	1
Code.8	Class: 8	Class: 8	Class: 8	Class: 8	Class: 8
Sensitivity.8	0	0	0	0	0
Specificity.8	1	1	1	1	1
Code.9	Class: 9	Class: 9	Class: 9	Class: 9	Class: 9
Sensitivity.9	0	1	1	1	0
Specificity.9	1	1	1	1	1
Code.10	Class: 10	Class: 10	Class: 10	Class: 10	Class: 10
Sensitivity.10	0.0	0.2	0.7	0.9	0.5
Specificity.10	1.0000000	1.0000000	0.9879518	0.9879518	0.9879518
Code.11	Class: 14	Class: 14	Class: 14	Class: 14	Class: 14
Sensitivity.11	0	0	0	0	0
Specificity.11	1	1	1	1	1
Code.12	Class: 15	Class: 15	Class: 15	Class: 15	Class: 15
Sensitivity.12	0	0	0	0	0
Specificity.12	1	1	1	1	1
Code.13	Class: 16	Class: 16	Class: 16	Class: 16	Class: 16
Sensitivity.13	0	0	0	0	0
Specificity.13	1	1	1	1	1

This method did not perform as expected with multi-variable too and it is close to KNN and decision tree in result as before.

Results

This section presents the modeling results and discusses the model performance.

We start checking the data and see if it has some elements that are leading us to a wrong conclusion. Cleaning the data, taking out some predictors that does not have enough information, taking out or fixing some outsiders was the initial work and see that some arrhythmia type are not present in our dataset (class 11, 12 and 13).

Later we see, despite to have to many predictors, we have several with high correlation.

And the data, some output in the class are not present or they are not equally distributed, except in our first division between "Normal" & "Arrhythmia", then the prediction of certain class are very difficult to established. This lack of flat distribution affect the partition of training and test dataset as well as the algorithm that trying to create the better prediction.

In the Analysis, we divided the question of the prediction in two:

a. **Detection of cardiac arrhythmia**
b. **Classification of cardiac arrhythmia**

Detection of cardiac arrhythmia

As a summary of **Detection of cardiac arrhythmia** we have (it was shown above):

Table 29: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
K-Nearest Neighbors	0.6021505	0.2045455	0.9591837
Decision Tree Classifier	0.7741935	0.7272727	0.8163265
Random Forest Classifier	0.8494624	0.7954545	0.8979592
SVM Classifier	0.7526882	0.6363636	0.8571429

Then with this, we get a accuracy of 84.9% with random forest, e. gr., our prediction of "Normal" & "Arrhythmia" is correct 84.9% of the time and if the patience actually has Arrhythmia with the sensitivity of 79.5%, this is the percentage of patience detected correctly (1 - sensitivity are patients with arrhythmia detected as normal). Remember, the arrhythmia class was treated as the 'Positive' class.

Variable Importance of Predictors

Which predictors help to get this percentage? Analyzing first the random forest which has the better results, we have:

```
##                                           Overall
## `Heart rate`                            100.00000
## `QRS duration`                           67.94778
## `Channel V1 Number of intrinsic deflections`   45.83778
## `Channel V1 R' wave Amplitude`           43.90221
## `Channel V1 QRSA`                        43.65389
## `Channel V6 T wave Amplitude`            41.53877
## `Channel DI T wave Amplitude`            32.12417
## `Channel AVR QRSTA`                      25.44268
## `Channel V1 R' wave Average`             25.07239
## `Channel AVR T wave Amplitude`           24.77490
## `Channel DII QRSTA`                      22.49002
## `Channel DI QRSTA`                       21.82767
## `T interval`                             17.36649
## `Channel V2 S wave Amplitude`            17.18325
## `Q-T interval`                           15.30938
## `Channel V3 QRSTA`                       15.28164
## `Channel V6 QRSTA`                       15.13620
## `Channel V5 JJ wave Amplitude`           15.03668
## `Channel AVL Number of intrinsic deflections`  14.49221
```

```
## `Channel V2 R' wave Amplitude`                13.93557
```

Reviewing all the models we used, except the decision tree that does not rank all the predictors, only that it use to create the tree, we have the top among them:

```
##                                           varimp_p1_mean
## QRS duration                                    89.31593
## Channel AVR QRSTA                               69.86377
## Channel DI T wave Amplitude                     65.92913
## Channel V6 T wave Amplitude                     65.74340
## Channel DI QRSTA                                64.32617
## Channel AVR T wave Amplitude                    60.80873
## Channel V6 QRSTA                                56.56103
## Channel DII QRSTA                               52.80346
## Channel V1 Number of intrinsic deflections      50.48361
## Channel V1 QRSTA                                50.11736
## Channel DII T wave Amplitude                    47.05064
## Channel AVL T wave Amplitude                    45.93153
## Channel V3 QRSTA                                45.13778
## Sex                                             44.10569
## Channel V5 T wave Amplitude                     42.55722
## Channel V1 R' wave Amplitude                    42.33868
## Channel V6 JJ wave Amplitude                    41.53819
## Channel AVR R wave Average                      41.39632
## Channel V4 S wave Average                       40.24883
## Channel V5 JJ wave Amplitude                    39.81938
```

Let's take a look to the principal predictors

Heart rate Predictor

Appear like the most relevant for random forest. Appear in the decision tree too but not in the top 20 in rest of the models.

```
##      rowname        .
## 1 Heart rate 8.827219
```

```
##      rowname Arrhythmia   Normal
## 1 Heart rate   8.827219 8.827219
```

We saw in the Analysis that Heart rate does not have correlation with other predictors. Then look that random forest took a good decision in give importance to this predictor.

QRS duration Predictor

The second Predictor from random forest (QRS duration) appear in others models too (actually in first place).

Channel V1 Number of intrinsic deflections & Channel V1 R' wave Amplitude Predictors

These predictors are present in the others models too.

Channel V1 QRSA Predictor

This predictor appear only in the Top20 of random forest. Is this something only used for this model? Let's see its position in the other models:

```
##           rowname        .
## 1 Channel V1 QRSA 28.85701
```

```
##           rowname Arrhythmia   Normal
## 1 Channel V1 QRSA   28.85701 28.85701
```

It has importance in knn. Let's check its correlation:

Table 30: Correlation for Channel V1 QRSA

correlated.1	correlated.2	value
Channel V1 QRSA	Channel V2 QRSA	0.7380082
Channel V1 QRSA	Channel V1 QRSTA	0.5629033
Channel AVR QRSA	Channel V1 QRSA	0.5608476
Channel V1 R wave Average	Channel V1 QRSA	0.5509871
Channel DI JJ wave Amplitude	Channel V1 QRSA	0.5236234

Table 31: Correlation for Channel V1 QRSA

correlated.1	correlated.2	value
Channel V1 JJ wave Amplitude	Channel V1 QRSA	-0.6501790
Channel V1 QRSA	Channel V2 JJ wave Amplitude	-0.5805285
Channel V1 T wave Amplitude	Channel V1 QRSA	-0.5637906
Channel DI QRSA	Channel V1 QRSA	-0.5570893
Channel V6 R wave Average	Channel V1 QRSA	-0.5495769

This predictor has some correlation with "Channel V1 QRSTA" which it is in the others models. Under this, could be that the others models indirectly use this type or information. Then Channel V1 QRSA predictors is used as a important predictor in more than random forest model.

Why is important all this review? Because we have too many predictors but not all of them give us the same "quality" of information to our predictions. Predictors with high correlation "enter in a competition" with other.

Classification of cardiac arrhythmia

As a summary of **Classification of cardiac arrhythmia** we have (it was shown above too):

Table 32: Prediction All Arrhytmia Classification Summary

Model	K-Nearest Neighbors	Decision Tree Classifier	Random Forest Classifier	SVM Classifier
Accuracy	0.6236559	0.7311828	0.7956989	0.6344086
Code.1	Class: 1	Class: 1	Class: 1	Class: 1
Sensitivity.1	1.0000000	0.9591837	0.9591837	0.9795918
Specificity.1	0.2045455	0.5000000	0.6590909	0.2500000
Code.2	Class: 2	Class: 2	Class: 2	Class: 2
Sensitivity.2	0.1111111	0.4444444	0.7777778	0.2222222
Specificity.2	1.0000000	1.0000000	0.9880952	1.0000000
Code.3	Class: 3	Class: 3	Class: 3	Class: 3
Sensitivity.3	1	1	1	1
Specificity.3	1.0000000	0.9888889	1.0000000	1.0000000
Code.4	Class: 4	Class: 4	Class: 4	Class: 4
Sensitivity.4	0.0000000	0.3333333	0.6666667	0.3333333
Specificity.4	1.0000000	1.0000000	0.9888889	1.0000000
Code.5	Class: 5	Class: 5	Class: 5	Class: 5
Sensitivity.5	0	0	0	0
Specificity.5	1	1	1	1
Code.6	Class: 6	Class: 6	Class: 6	Class: 6
Sensitivity.6	0.2	0.8	0.8	0.0

Table 32: Prediction All Arrhytmia Classification Summary

Specificity.6	1.0000000	0.9886364	0.9886364	1.0000000
Code.7	Class: 7	Class: 7	Class: 7	Class: 7
Sensitivity.7	0	0	0	0
Specificity.7	1	1	1	1
Code.8	Class: 8	Class: 8	Class: 8	Class: 8
Sensitivity.8	0	0	0	0
Specificity.8	1	1	1	1
Code.9	Class: 9	Class: 9	Class: 9	Class: 9
Sensitivity.9	1	1	1	0
Specificity.9	1	1	1	1
Code.10	Class: 10	Class: 10	Class: 10	Class: 10
Sensitivity.10	0.2	0.7	0.9	0.5
Specificity.10	1.0000000	0.9879518	0.9879518	0.9879518
Code.11	Class: 14	Class: 14	Class: 14	Class: 14
Sensitivity.11	0	0	0	0
Specificity.11	1	1	1	1
Code.12	Class: 15	Class: 15	Class: 15	Class: 15
Sensitivity.12	0	0	0	0
Specificity.12	1	1	1	1
Code.13	Class: 16	Class: 16	Class: 16	Class: 16
Sensitivity.13	0	0	0	0
Specificity.13	1	1	1	1

Here, we get a accuracy of 79.6%, e. gr., we have this percentage of be correct in the kind of arrhythmia, where “normal” is one class and the more frequent one. This value es lower than the first prediction and we expected because we do not have too many patients and the distribution is is not equal among the classes.

In sensitivity for class 1 (“Normal”) is quite good and better than in the first prediction. And the better result is 98% is in the SVM model. Is this good? Here the Arrhythmia class is not the ‘Positive’ class as before, then the concept of sensitivity change in relation to the previous prediction. Here we are considering the sensitivity and specificity in each class, and the result has to be analyzed in this environment.

We see some classes with sensitivity = 0 and specificity = 1. This means that all the prediction for these classes are wrong! This happened with all the models for these classes:
* Class: 7
* Class: 8
* Class: 14
* Class: 15
* Class: 16

and in the case of Class 5, the situation it is the same, except for decision tree shows a little better numbers.

Table 33: Presence of codes in our dataset

Class code	Class name	N Ocurrences	Percentage
5	Sinus tachycardy	13	2.9
7	Ventricular Premature Contraction (PVC)	3	0.7
8	Supraventricular Premature Contraction	2	0.4
14	Left ventricule hypertrophy	4	0.9
15	Atrial Fibrillation or Flutter	5	1.1
16	Others	22	4.9

The numbers of occurrences maybe be explains for few patients with these arrhythmia in the dataset, except for class 5 and 16 which has more that class 9 and this last one can be resolved for some methods. These few occurrences should be related that no predictor has a high correlation with this classes, particularly for Class 16, as its same say "Others" maybe is a group of not homogeneous class and for that no model can detect it.

Class 9, in the other hand, in 3 models has a perfect score: sensitivity = 1 and specificity = 1. Too good to be true? The only point to see it is the low numbers of patients of this disease, only the 2%. Similar situation is with Class 3, but it combines the perfect score and the worse score depending the model. Because the random forest give it the perfect score, we can have a good classification for this class.

Looking the rest of the classes, we see:

Table 34: Prediction in remaining Arrhytmia Classification

Accuracy	0.5268817	0.6236559	0.7311828	0.7956989	0.6344086
Code.1	Class: 1	Class: 1	Class: 1	Class: 1	Class: 1
Sensitivity.1	1.0000000	1.0000000	0.9591837	0.9591837	0.9795918
Specificity.1	0.0000000	0.2045455	0.5000000	0.6590909	0.2500000
Code.2	Class: 2	Class: 2	Class: 2	Class: 2	Class: 2
Sensitivity.2	0.0000000	0.1111111	0.4444444	0.7777778	0.2222222
Specificity.2	1.0000000	1.0000000	1.0000000	0.9880952	1.0000000
Code.4	Class: 4	Class: 4	Class: 4	Class: 4	Class: 4
Sensitivity.4	0.0000000	0.0000000	0.3333333	0.6666667	0.3333333
Specificity.4	1.0000000	1.0000000	1.0000000	0.9888889	1.0000000
Code.6	Class: 6	Class: 6	Class: 6	Class: 6	Class: 6
Sensitivity.6	0.0	0.2	0.8	0.8	0.0
Specificity.6	1.0000000	1.0000000	0.9886364	0.9886364	1.0000000
Code.10	Class: 10	Class: 10	Class: 10	Class: 10	Class: 10
Sensitivity.10	0.0	0.2	0.7	0.9	0.5
Specificity.10	1.0000000	1.0000000	0.9879518	0.9879518	0.9879518

we have a "realistic" results, not to good to be true or too bad.

Like a summary, we have considering the whole dataset for Ocurrences and percentage:

Table 35: Result by class codes in our validation dataset

Class code	Class name	N Ocurrences	Percentage	Prediction
1	Normal	245	54.2	Predictable
2	Ischemic changes (Coronary Artery Disease)	44	9.7	Predictable
3	Old Anterior Myocardial Infarction	15	3.3	Predictable
4	Old Inferior Myocardial Infarction	15	3.3	Predictable
5	Sinus tachycardy	13	2.9	No Predictable
6	Sinus bradycardy	25	5.5	Predictable
7	Ventricular Premature Contraction (PVC)	3	0.7	No Predictable
8	Supraventricular Premature Contraction	2	0.4	No Predictable
9	Left bundle branch block	9	2.0	Predictable
10	Right bundle branch block	50	11.1	Predictable
14	Left ventricule hypertrophy	4	0.9	No Predictable
15	Atrial Fibrillation or Flutter	5	1.1	No Predictable
16	Others	22	4.9	No Predictable

Variable Importance of Predictors

We have the same question again, now for the complete classification. Which predictors help to get this percentage? Are these the same the first prediction? Analyzing first the random forest which has the better results, we have:

```
##                                           Overall
## `Heart rate`                            100.00000
## `Channel V1 R' wave Amplitude`           37.20654
## `Channel V1 QRSA`                        34.73682
## `Channel V6 T wave Amplitude`            33.07979
## `Channel V1 Number of intrinsic deflections` 31.84650
## `Channel V1 R' wave Average`             27.49445
## `Channel V5 T wave Amplitude`            26.73103
## `QRS duration`                           21.20552
## `Channel AVR T wave Amplitude`           18.41807
## `Channel V4 T wave Amplitude`            18.28262
## `Channel DI T wave Amplitude`            16.97207
## `Q-T interval`                           16.70657
## `Channel V3 Q wave Amplitude`            13.84523
## `Channel V2 R' wave Amplitude`           13.41139
## `Channel AVF Q wave Average`             12.87968
## `Channel V3 Q wave Average`              12.75818
## `Channel V3 R wave Amplitude`            12.17041
## `Channel DII T wave Amplitude`           11.29119
## `Channel AVF Q wave Amplitude`           11.14968
## `Channel V3 R wave Average`              11.11864
```

Comparing the Variable Importance for the random forest in the first predition with this one, the 6th first are present in the Top20 and the "Channel V5 T wave Amplitude" in the 7th position that it is not. Then a very good similarity we can see in both predictions.

Reviewing all the models we used, except the decision tree that does not rank all the predictors, only that it use to create the tree, we have the top among them:

```
##                                         varimp_p2_mean
## Channel V6 T wave Amplitude                  66.49030
## Channel V5 T wave Amplitude                  63.98637
## Channel AVR T wave Amplitude                 59.55084
## Heart rate                                   58.81628
## Channel DI T wave Amplitude                  58.62710
## Channel DII T wave Amplitude                 57.78705
## Channel V3 QRSA                              52.97570
## Channel V4 T wave Amplitude                  52.40301
## Channel V1 Number of intrinsic deflections   52.04656
## Channel V3 R wave Average                    52.01721
## Channel V3 S wave Average                    51.57096
## Channel V3 QRSTA                             50.81369
## Channel V3 S wave Amplitude                  49.83688
## Channel V5 JJ wave Amplitude                 49.47028
## Channel V3 Q wave Amplitude                  49.32486
## Channel V6 QRSTA                             49.14026
## Channel V3 Q wave Average                    48.98726
## Channel V3 R wave Amplitude                  48.85468
## Channel AVR QRSTA                            48.31628
## Channel V2 R wave Amplitude                  47.75639
```

Let's take a look to the principal predictors

Heart rate Predictor

Appear like the most relevant for random forest. In this opportunity it is appear in the decision tree **and** in the top 20 in rest of the models. If we like to see more detail, the big change was made by KNN model:

```
##      rowname        .
## 1 Heart rate 50.02028
```

```
##      rowname       X1       X2       X3       X4      X5       X6       X7
## 1 Heart rate 26.42857 58.71599 98.46939 98.39286 37.2449 28.57143 26.42857
##         X8       X9      X10      X14      X15      X16
## 1 26.42857 66.83673 71.17347 26.42857 26.42857 58.71599
```

Channel V1 R' wave Amplitude Predictor

The second Predictor from random forest (Channel V1 R' wave Amplitude) does not appear in the Top20 of others models. Does si ti appear with some importance?

```
##                         rowname        .
## 1 Channel V1 R' wave Amplitude 11.99303
```

```
##                         rowname       X1       X2       X3       X4       X5
## 1 Channel V1 R' wave Amplitude 3.571429 3.571429 6.785714 2.346939 3.571429
##         X6       X7       X8       X9      X10      X14      X15      X16
## 1 3.571429 3.571429 58.34184 30.61224 22.06633 8.613445 5.714286 3.571429
```

The SVM give it some importance, KNN not.

Channel V1 QRSA Predictor

This predictor is not present in the op20 of others models

In this case, the variables importance that has random forest it is less similar to the one created by the others models compared with the previos prediction. This is because the predction for the other models (knn, svm) cahnge more when we create the fitting for the Noraml/Arrhythmia than we we works with all classes.

Conclusion

This section that gives a brief summary of the report, its limitations and future work.

The first thing to note is that we have a big numbers of predictors (large input dimensionality) while the number of observations is really low. This big numbers of predictors has some degree of correlation among them. This is a challenge to the algorithms we use, where random forest look like has some advantages.

Some approach artificially expand the size of the dataset predictors by multiplying/averaging/(other function) some observations. This can work in some cases, required knowledge to be made properly. We are not to apply this here, besides the last point, we already have to many predictors for our algorithm.

For the low number of patients (row in our dataset) we used bootstrapping in our training.

Despite of this low numbers of rows we still divide the dataset in train and validation, because we do not want to brake this rule of independency of the data in the validation set

The initial task of cleaning the data, using this real data, shows its importance, detecting data incomplete, erroneous data, that can lead us to a bad conclusion. More knowledge of physiology can be useful in detect more erroneous data in all the predictors.

In this classifier problem, when we look between "Normal" & "Arrhythmia", with random forest we get a good result in accuracy and sensitivity:

Table 36: Prediction Normal/Arrhytmia Summary

Model	Accuracy	Sensitivity	Specificity
Random Forest Classifier	0.8494624	0.7954545	0.8979592

then we can feel we have a good result with a dataset with only 452 patients.

When we work with all the different classification in the arrhythmia, its 15 class, we notices that the distribution of this class are not equal, some of them are not even present in the dataset and the classification of "Others" was no predictable, maybe because under this name several different diseases. This characteristics of the dataset can not be resolved properly with the model used, and not model can do it.

In the second classifier problem, when we look among all the classification, random forest we get a good result:

Table 37: Prediction All Arrhytmia Classification Summary

Model	Random Forest Classifier
Accuracy	0.7956989
Code.1	Class: 1
Sensitivity.1	0.9591837
Specificity.1	0.6590909
Code.2	Class: 2
Sensitivity.2	0.7777778
Specificity.2	0.9880952
Code.3	Class: 3
Sensitivity.3	1
Specificity.3	1
Code.4	Class: 4
Sensitivity.4	0.6666667
Specificity.4	0.9888889
Code.5	Class: 5
Sensitivity.5	0

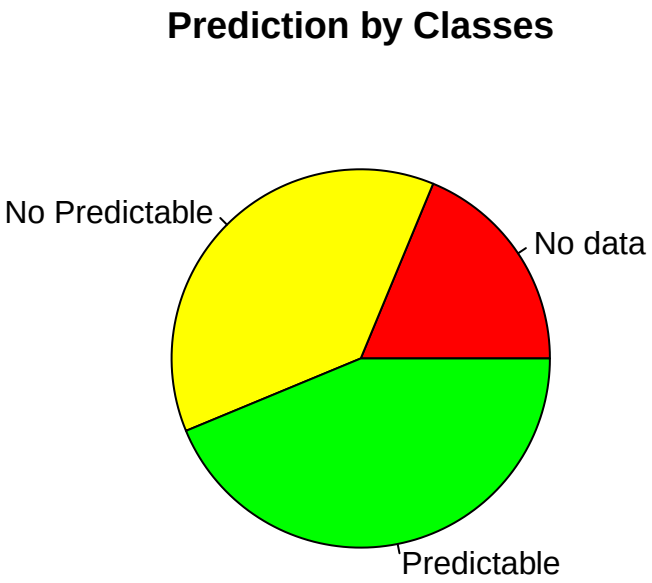
Table 37: Prediction All Arrhytmia Classification Summary

Specificity.5	1
Code.6	Class: 6
Sensitivity.6	0.8
Specificity.6	0.9886364
Code.7	Class: 7
Sensitivity.7	0
Specificity.7	1
Code.8	Class: 8
Sensitivity.8	0
Specificity.8	1
Code.9	Class: 9
Sensitivity.9	1
Specificity.9	1
Code.10	Class: 10
Sensitivity.10	0.9
Specificity.10	0.9879518
Code.11	Class: 14
Sensitivity.11	0
Specificity.11	1
Code.12	Class: 15
Sensitivity.12	0
Specificity.12	1
Code.13	Class: 16
Sensitivity.13	0
Specificity.13	1

5 points less in accuracy compared with the first prediction.

As a summary of the 16 classes (class 1 is normal) we have:
No data in dataset: 3
No prediction can be construct with the models used: 6
Classes can be predicted with our models: 7

In a pie graph:



Can this be improved? Yes, if we have more data that cover our lack of patients with some arrhythmia. Check if the "Others" can be divided in some way and then look for a rule to detect it.

Other approach that we did not take is trying to create another category in base of group of arrhythmia, for example, all the arrhythmia related with Myocardial, or Synus, because I need medical knowledge that I do not have.

We use some classical algorithm as KNN, decision tree and random forest and SVM not as popular as the first ones but more models are available and can be implemented and some others training method can be used in these models. I selected them considered better for classification task but I know more options are available in this everyday growing up area.

The original problem was that some software can not detect the arrhythmia properly. With this work we can see we can create some level of confidence in the case of "Normal" & "Arrhythmia" and for all the classes but for a more robust solution more data will be necessary to train our model.