Fetal Health Rate - Analysis

Methods

In this section, all the methods for the multilinear regression analysis over the white wine are presented in order. This dataset is available for anyone on-line. The data information were collected based on the red and white variants of the Portuguese "Vinho Verde" wine. The dataset contains objective tests data such as:

- Fixed Acidity, Volatile Acidity, Citric Acid, Residual Sugar, Chlorides, Free Sulfur Dioxide, Total Sulfur Dioxide, Density, pH, Sulphates, Alcohol.

and sensory data such as:

- Quality.

This DATASET is highly versatile. Indeed, not only the DATASET source recommend multiple linear regression but also related papers introduce some other statistical methods to analyze, extract and present data from this DATASET such as classification. Nevertheless, our aim will be to implement multiple linear regression using the quality as the output variable.

In each method, I decided to use it as numerical from 1 to 3, and as characters *normal*, *suspect*, and *pathologic*. First, I take a quick look into the data:

Data Cleaning and Enhancement

In this data all variables are abbreviated for better analysis process table 1, and grouped in two sections to perform the statistical analysis: explanatory and response variables.

Abbreviations	Description
LB	FHR baseline (beats per minute)
AC	Number of accelerations per second
FM	Number of fetal movements per second
UC	Number of uterine contractions per second
DL	Number of light decelerations per second
DS	Number of severe decelerations per second
DP	Number of prolonged decelerations per second
ASTV	percentage of time with abnormal short term variability
MSTV	mean value of short term variability
ALTV	percentage of time with abnormal long term variability
MLTV	mean value of long term variability
Width	width of FHR histogram
Min	minimum of FHR histogram
Max	Maximum of FHR histogram
Nmax	Number of histogram peaks
Nzeros	Number of histogram zeros
Mode	histogram mode
Mean	histogram mean
Median	histogram median
Variance	histogram variance
Tendency	histogram tendency

Abbreviations	Description
CLASS	FHR pattern class code (1 to 10)

CLASS as the response variable, and the rest as explanatory variables. The data might not be transformed into numbers or characters due to the fact each process might be using the response variable differently.

Statistics

To start building any model, the variable analysis is crucial. Let's take a look into the data. Before, I remove non relevant variables such as:

- Width, Min, Max, Nmax, Nzeros, Mode, Mean, Median, Variance and Tendency.

LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	ALTV	MLTV	CLASS
120	0.000	0	0.000	0.000	0	0	73	0.5	43	2.4	2
132	0.006	0	0.006	0.003	0	0	17	2.1	0	10.4	1

The usual way to analyze the data is statistics and correlations.

These variables are based on other variables from the original data which might interfere with the analysis.

```
##                     LB       AC       FM       UC       DL       DS       DP
## nbr.val        2126.000 2126.000 2126.000 2126.000 2126.000 2126.000 2126.000
## nbr.null          0.000  894.000 1311.000  332.000 1231.000 2119.000 1948.000
## nbr.na            0.000    0.000    0.000    0.000    0.000    0.000    0.000
## min             106.000    0.000    0.000    0.000    0.000    0.000    0.000
## max             160.000    0.019    0.481    0.015    0.015    0.001    0.005
## range            54.000    0.019    0.481    0.015    0.015    0.001    0.005
## sum          283404.000    6.757   20.156    9.283    4.017    0.007    0.337
## median          133.000    0.002    0.000    0.004    0.000    0.000    0.000
## mean            133.304    0.003    0.009    0.004    0.002    0.000    0.000
## SE.mean           0.213    0.000    0.001    0.000    0.000    0.000    0.000
## CI.mean.0.95      0.419    0.000    0.002    0.000    0.000    0.000    0.000
## var              96.842    0.000    0.002    0.000    0.000    0.000    0.000
## std.dev           9.841    0.004    0.047    0.003    0.003    0.000    0.001
## coef.var          0.074    1.216    4.922    0.675    1.567   17.403    3.722
##                    ASTV     MSTV      ALTV      MLTV    CLASS
## nbr.val        2126.000 2126.000  2126.000  2126.000 2126.000
## nbr.null          0.000    0.000  1240.000   137.000    0.000
## nbr.na            0.000    0.000     0.000     0.000    0.000
## min              12.000    0.200     0.000     0.000    1.000
## max              87.000    7.000    91.000    50.700    3.000
## range            75.000    6.800    91.000    50.700    2.000
## sum           99901.000 2833.500 20934.000 17406.900 2773.000
## median           49.000    1.200     0.000     7.400    1.000
## mean             46.990    1.333     9.847     8.188    1.304
## SE.mean           0.373    0.019     0.399     0.122    0.013
## CI.mean.0.95      0.731    0.038     0.782     0.239    0.026
## var             295.593    0.780   338.445    31.677    0.377
## std.dev          17.193    0.883    18.397     5.628    0.614
## coef.var          0.366    0.663     1.868     0.687    0.471
```

The analysis results are:

- There is no missing values in the data.
- In general there are 2126 observations.
- All the variables do not have minimum variables except from the tendency. Nevertheless, that variable is not a basis variable but one resulting from others.
- The minimum baby fetal heart beats base is 106 per minute, and the maximum 160 per minute. According to Johns Hopkins Medicine the fetal health rate average is in between 110 and 160. So, the fetal health max beats base is 160, and matches with the literature. On the other hand, the minimum data fetal heart beat base min value is 106 which is not in the average range. This case is named fetal **bradyarrhythmia**.
- All the explanatory variables are continuous.

Additionally, implementing basic grouping selection in the data shows that there is a high (77.85%) of cases where the fetal state is normal, (13.88%) wealthy suspect fetal, and unfortunately (8.28%) fetal in pathological state. This implies that the data is not uniformly grouped which might not allow some statistical methods fit as we expect to.

```
## [1] 14
```

Lastly, I performed duplication analysis which might guide to a biased analysis because of data collection error. Nevertheless, the number of repetitions is not significant (14 out of 2126) which shows that there is not evidence to affirm that the data repetitions is because of mismanagement of the cardiograms.

Correlation

The "GGally" package describes a widely complete correlation, density and frequency table all in one included in the same plot. For that reason, we select this rather than other packages.

```
## Loading required package: ggplot2
```

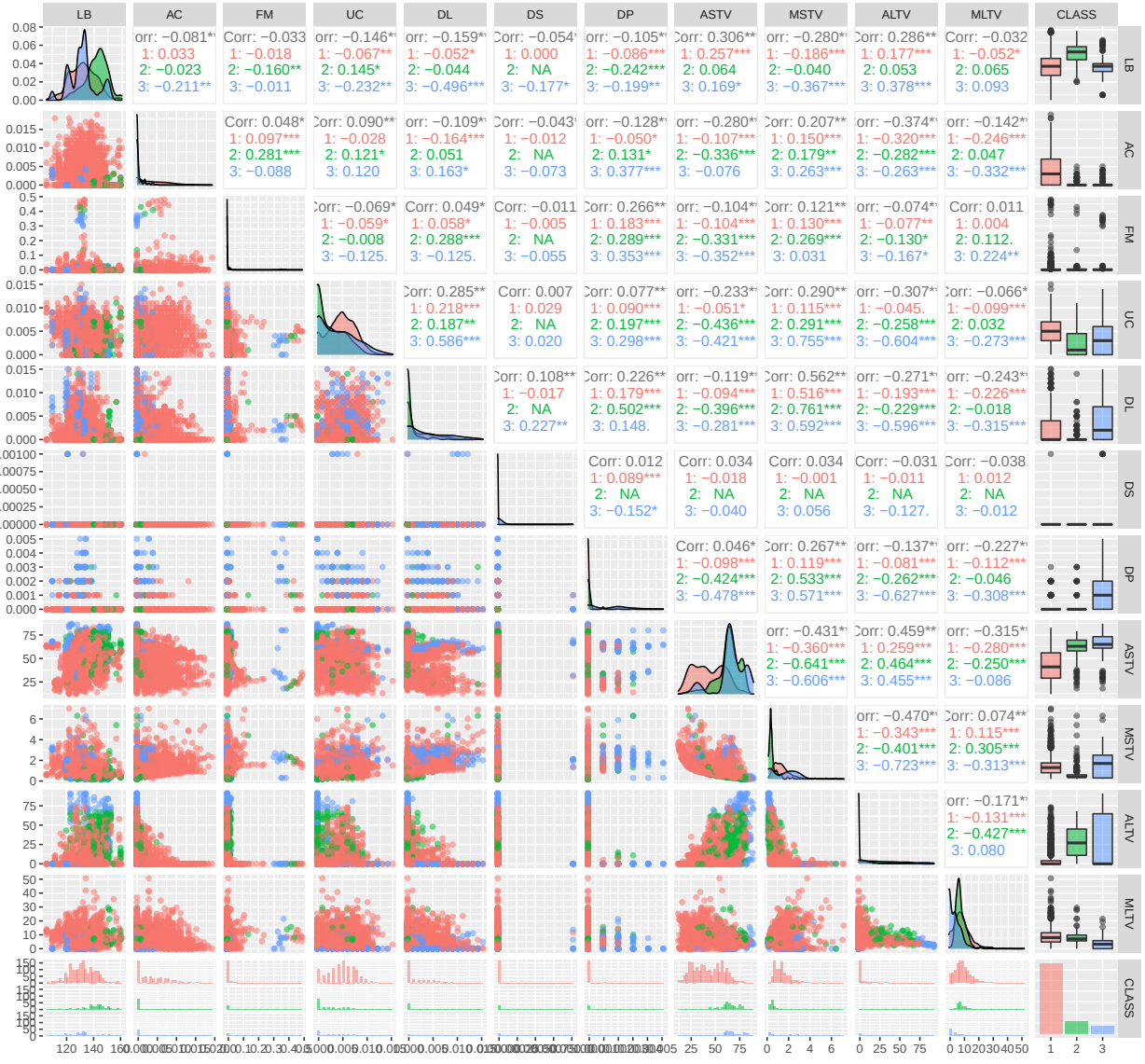
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The above correlation plot shows not only correlation in between the variables but correlation in each variable with the response variable and order by strong positive (top), not or medium (center), and negative (bottom) correlations. The results of the analysis are:

- In the data we found that there is not strong correlation in between the variables. The variables analysis are classified as quantitative inputs. There are not transformation of quantitative inputs or basis expansions.
- There are negative and positive correlations. For instance, the Number of deceleration per second with the mean value of short term variability. Why? The cardiogram shows results show that on average with a range of 3-5 BPM the fetal heart rate speeds up slightly and the slows down slightly.

- There is an important number of outliers for each of the explanatory variables presented in the data. So, performing linear regression might be not a good and optimal idea. Nevertheless, the concentrated data look normally distributed which might influence to the well fitting model.

Training and testing sets

Before starting the entire process we select the eighty percent of the actual data as the training set to perform the analysis, and the testing set to test the model. This is usually done to enhance the time processing model, and set the algorithm not only for this data but for general related data. I do not include validation data because I could not find another data related to FHR from a reliable source.

```
##       LB    AC    FM    UC    DL DS DP ASTV MSTV ALTV MLTV   CLASS
## 526  158 0.012 0.026 0.000 0.000  0  0   42  0.9    0  2.8  normal
## 195  150 0.000 0.000 0.006 0.000  0  0   56  0.5   19  7.9 suspect
## 1842 137 0.002 0.004 0.007 0.006  0  0   58  1.9    0  3.9  normal
## 1142 122 0.000 0.000 0.005 0.005  0  0   26  1.3    0  9.8  normal
## 1253 112 0.000 0.000 0.004 0.000  0  0   23  1.3   11 13.0  normal
```

```
##     LB    AC    FM    UC    DL DS    DP ASTV MSTV ALTV MLTV      CLASS
## 3  133 0.003 0.000 0.008 0.003  0 0.000   16  2.1    0 13.4     normal
## 5  132 0.007 0.000 0.008 0.000  0 0.000   16  2.4    0 19.9     normal
## 15 130 0.006 0.408 0.004 0.005  0 0.001   21  2.3    0  7.9     normal
## 21 129 0.000 0.340 0.004 0.002  0 0.003   30  2.1    0  8.5 pathologic
## 22 128 0.005 0.425 0.003 0.003  0 0.002   26  1.7    0  6.7     normal
```

Multinomial Log-Linear Model

Regression models, as it is described in the introduction, are not "black boxes" that do not described the variable relations and do not show much more description about the process but the result. I decided to use one powerful regression model belonging to Generalized Additive Models as being an extension for the traditional logistic regression model. The model will run for 3 possible classifications 2 independent binary logistic regression model. To perform the model, I used "nnet" package and "multinom" function.

The output represents the relation in between the inputs and the log odds i.e. the log of the ratio probabilities which might be interpret as the probability of selecting suspect fetal rate health vs normal fetal health rate, and the probability of selecting pathologic fetal health rate vs normal fetal health rate.

Output

This model-running output includes some iteration history and include the final negative log-likelihood 497.01. The double of this value is the model's residual deviance equal to 994.02.

```
## # weights:  39 (24 variable)
## initial  value 1867.640891
## iter  10 value 841.227308
## iter  20 value 672.317467
## iter  30 value 616.454492
## iter  40 value 571.521562
## iter  50 value 537.196717
## iter  60 value 537.163389
## iter  70 value 529.360017
## iter  80 value 521.093901
## iter  90 value 505.078842
## iter 100 value 497.012822
## final  value 497.012822
## stopped after 100 iterations
```

```
## Call:
## multinom(formula = CLASS ~ ., data = train)
##
## Coefficients:
##            (Intercept)           LB        AC       FM        UC        DL
## pathologic   -11.06056 -0.004132869 -602.9275 9.694224 -178.4279  170.0267
## suspect      -13.74403  0.080526300 -598.2284 9.066279 -180.6823 -144.8778
##                    DS        DP       ASTV        MSTV       ALTV        MLTV
## pathologic 40.390626 1436.8554 0.12914127  0.90350242 0.05126934 0.002637609
## suspect    -1.900256  233.6529 0.04320507 -0.07508646 0.01750418 0.004514100
##
## Std. Errors:
##            (Intercept)         LB          AC       FM         UC          DL
## pathologic    2.354235 0.01568815 0.005608045 2.874280 0.04512010 0.039150693
## suspect       1.667158 0.01111987 0.009102137 2.053347 0.01756116 0.008381075
##                      DS          DP        ASTV      MSTV        ALTV
## pathologic 5.397658e-04 0.007437533 0.015229937 0.1754566 0.006807454
## suspect    5.641561e-05 0.005668285 0.008386592 0.1917992 0.005521643
##                  MLTV
## pathologic 0.03478628
## suspect    0.02323738
##
## Residual Deviance: 994.0256
## AIC: 1042.026
```

P - Values

Here, I decided to perform manually the p-values to measure the significance level for all the variables in each of the models.

	pathologic	suspect
(Intercept)	2.625e-06	2.220e-16
LB	7.922e-01	4.432e-13
AC	0.000e+00	0.000e+00
FM	7.442e-04	1.008e-05
UC	0.000e+00	0.000e+00
DL	0.000e+00	0.000e+00
DS	0.000e+00	0.000e+00
DP	0.000e+00	0.000e+00
ASTV	0.000e+00	2.581e-07
MSTV	2.612e-07	6.954e-01
ALTV	5.018e-14	1.523e-03
MLTV	9.395e-01	8.459e-01

The method is using the coefficients and standard errors from the multinomial regression, and we calculate one minus the "normalized" results. As it can be observed, the first log-linear equation (pathologic vs normal) has almost all variables significant respect to the log odds response variable but the FHR baseline and mean value of long term variability. On the other log-linear equation (suspect vs normal) has two not significant variables: mean value of short term variability and mean value of long term variability.

Best Multinomial Log-Linear Model

After calculating the p-values that evidence the significance values, I decided to perform a better version of this model removing the non significant variables from the model. The non significant values might be removed by two options:

1. Removing the non significant values manually. The result is:

with p-values given by,

```
##                pathologic      suspect
## (Intercept) 0.000000e+00 3.347253e-08
## AC          0.000000e+00 0.000000e+00
## FM          5.485084e-09 2.204681e-12
## UC          0.000000e+00 0.000000e+00
## DL          0.000000e+00 0.000000e+00
## DS          0.000000e+00 0.000000e+00
## DP          0.000000e+00 0.000000e+00
## ASTV        0.000000e+00 1.537657e-10
## MSTV        1.561063e-08 1.474682e-02
## ALTV        4.618528e-14 4.615659e-05
```

with an AIC and residual deviance of 1122.859 and 1082.85 respectively.

2. The "step" method might be useful in this case. Reading the manual this specific case belongs to the acceptable modeling regression that fit into the implementation using the "backward" direction and $k = log$.

```
## Call:
## multinom(formula = CLASS ~ LB + AC + FM + UC + DL + DP + ASTV +
##     MSTV + ALTV + MLTV, data = train)
##
## Coefficients:
##            (Intercept)          LB        AC       FM         UC         DL
## pathologic   -9.693925 -0.01615408 -690.8225 12.65505  -98.72835   58.19688
## suspect     -14.249399  0.08075672 -617.9600 12.07954 -128.93300 -315.97440
##                    DP       ASTV      MSTV       ALTV         MLTV
## pathologic 2070.3337 0.13387419 0.7892389 0.04961730 -0.022370724
## suspect     295.8782 0.04844476 0.1043056 0.01849734 -0.003428944
##
## Std. Errors:
##            (Intercept)         LB         AC       FM         UC         DL
## pathologic    2.449794 0.01668939 0.01113618 2.839991 0.04175835 0.02987663
## suspect       1.702221 0.01128789 0.01300760 1.990418 0.01509501 0.01152631
##                     DP        ASTV      MSTV        ALTV      MLTV
## pathologic 0.008143330 0.015611399 0.2006324 0.007239509 0.0436986
## suspect    0.004315559 0.008553432 0.1954211 0.005569273 0.0240173
##
## Residual Deviance: 980.1753
## AIC: 1024.175
```

Final Multinomial Log-Linear Model

After the best multinomial logistic model setting, I decided that the "step" model is the best choice. This model-running output includes some iteration history and include the final negative log-likelihood 490.087. The double of this value is the model's residual deviance equal to 980.1753.

$$\beta_{pathologic} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \\ \beta_{10} \end{bmatrix} = \begin{bmatrix} 2.449794 \\ 0.016689 \\ 0.01113 \\ 2.839 \\ 0.04175 \\ 0.02987 \\ 0.00814 \\ 0.01561 \\ 0.2006 \\ 0.00723 \\ 0.0436 \end{bmatrix} \qquad \beta_{suspect} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \\ \beta_{10} \end{bmatrix} = \begin{bmatrix} 1.7022 \\ 0.01128 \\ 0.01300 \\ 1.9904 \\ 0.01509 \\ 0.01152 \\ 0.00431 \\ 0.00855 \\ 0.1954 \\ 0.00556 \\ 0.02401 \end{bmatrix}$$

Estimates interpretation

- For one-unit increase in the variable fetal movement is associated with the increase in the log odds of being in suspect fetal rate health vs normal fetal health rate in the amount of 1.9904.
- For one-unit increase in the variable accelerations per second is associated with the decrease in the log odds of being in pathologic fetal rate health vs normal fetal health rate in the amount of 0.01113.
- For one-unit increase in the variable mean value of short term variability is associated with the increase in the log odds of being in suspect fetal rate health vs normal fetal health rate in the amount of 0.1954.
- For one-unit increase in the variable uterine contractions per second is associated with the increment in the log odds of being in pathologic fetal rate health vs normal fetal health rate in the amount of 0.0417.

Performance

The model performance has an acceptable precision due to the confusion matrix results and the specificity, sensitivity, and accuracy values for training and testing data. Nevertheless, there is an important miss-classification that might be avoid in each of the classes.

```
## Loading required package: lattice
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal        311          8      25
##   pathologic      4         26       1
##   suspect        10          8      33
##
## Overall Statistics
##
##                Accuracy : 0.8685
##                  95% CI : (0.8327, 0.8992)
##     No Information Rate : 0.7629
##     P-Value [Acc > NIR] : 3.499e-08
##
##                   Kappa : 0.635
##
##  Mcnemar's Test P-Value : 0.004211
##
## Statistics by Class:
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9569           0.61905        0.55932
```

```
## Specificity                 0.6733           0.98698        0.95095
## Pos Pred Value              0.9041           0.83871        0.64706
## Neg Pred Value              0.8293           0.95949        0.93067
## Prevalence                  0.7629           0.09859        0.13850
## Detection Rate              0.7300           0.06103        0.07746
## Detection Prevalence        0.8075           0.07277        0.11972
## Balanced Accuracy           0.8151           0.80301        0.75514
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal       1269         23      86
##   pathologic     11         84      13
##   suspect        50         27     137
##
## Overall Statistics
##
##                Accuracy : 0.8765
##                  95% CI : (0.8599, 0.8917)
##     No Information Rate : 0.7824
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6402
##
##  Mcnemar's Test P-Value : 0.0003207
##
## Statistics by Class:
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9541           0.62687        0.58051
## Specificity                 0.7054           0.98467        0.94740
## Pos Pred Value              0.9209           0.77778        0.64019
## Neg Pred Value              0.8106           0.96859        0.93338
## Prevalence                  0.7824           0.07882        0.13882
## Detection Rate              0.7465           0.04941        0.08059
## Detection Prevalence        0.8106           0.06353        0.12588
## Balanced Accuracy           0.8298           0.80577        0.76396
```

This above table evidence that each of those training and testing present an important percentage of miss-classification in each of the classes.

KNN

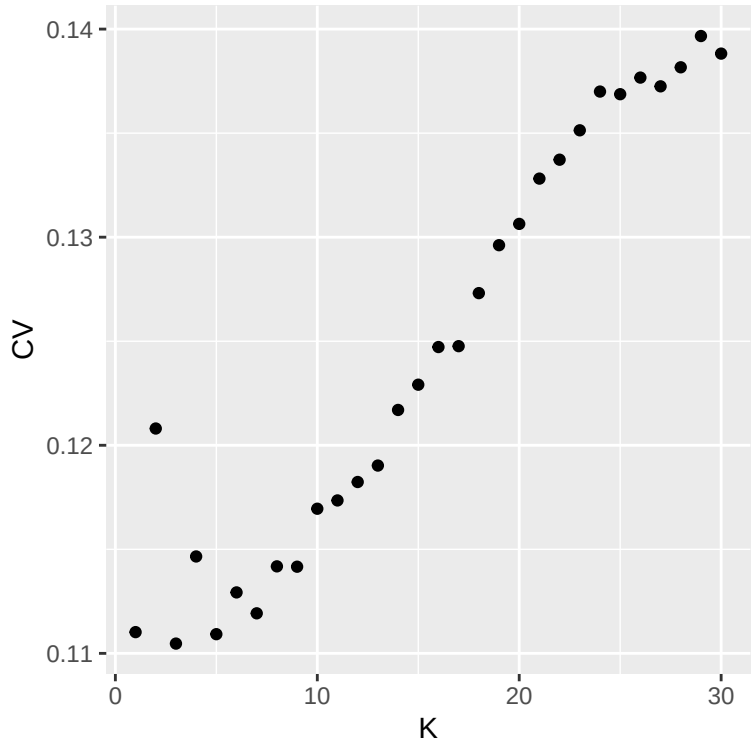
I decided to start the classification analysis using one of the most used and accurate classification methods: KNN. This supervised machine learning algorithm usually is used not only for classification problems but for regression analysis. Additionally, there is no need to build a model because it uses the input data calculating neighbors and classifying the data.

To perform this model, I used "purrr" and "class" libraries using the created function in Statistical Learning class at UTRGV[1] to obtain the best K for the classification. My decision was based on validating similar models to the one created in class selecting that one as the most accurate[2].

[1]For more information please email tamer.oraby@utrgv.edu

[2]Other methods use R libraries performing training functions.

After applying the best K (best possible K for this data is 3) function the attributes are ready to set the final model.



This model will use the inputs attributes to classify the training data. As I mention before, this model is simple and accurate when the best K is selected. Then, I applied that into the training and testing data.

Performance

The KNN performance is measured not by the estimates interpretation but the confusion matrix and the MSE in both: training and testing data.

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal        314          7      11
##   pathologic      3         34       1
##   suspect         8          1      47
##
## Overall Statistics
##
##                Accuracy : 0.9272
##                  95% CI : (0.8983, 0.95)
##     No Information Rate : 0.7629
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8077
##
##  Mcnemar's Test P-Value : 0.5573
##
## Statistics by Class:
```

```
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9662           0.80952         0.7966
## Specificity                 0.8218           0.98958         0.9755
## Pos Pred Value              0.9458           0.89474         0.8393
## Neg Pred Value              0.8830           0.97938         0.9676
## Prevalence                  0.7629           0.09859         0.1385
## Detection Rate              0.7371           0.07981         0.1103
## Detection Prevalence        0.7793           0.08920         0.1315
## Balanced Accuracy           0.8940           0.89955         0.8860
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal       1297         13      45
##   pathologic      7        117       4
##   suspect        26          4     187
##
## Overall Statistics
##
##                Accuracy : 0.9418
##                  95% CI : (0.9296, 0.9524)
##     No Information Rate : 0.7824
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8349
##
##  Mcnemar's Test P-Value : 0.07567
##
## Statistics by Class:
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9752           0.87313         0.7924
## Specificity                 0.8432           0.99298         0.9795
## Pos Pred Value              0.9572           0.91406         0.8618
## Neg Pred Value              0.9043           0.98919         0.9670
## Prevalence                  0.7824           0.07882         0.1388
## Detection Rate              0.7629           0.06882         0.1100
## Detection Prevalence        0.7971           0.07529         0.1276
## Balanced Accuracy           0.9092           0.93306         0.8859
```

Confusion matrix interpretation

To check the KNN performance classifying the data we use the confusion matrix. The confusion matrix shows that based on the training data (testing data) the KNN classify the entire data with an accuracy of 94.35% (92.7%). The model predicted 1298 (314) to be classified as normal FHR out of 1700 training data (426 testing data), with a misclassification of 32 (101). The model predicted 118 (34) to be classified as suspect FHR out of 1700 training data (426 testing data), with a misclassification of 118 (35). Lastly, the model predicted 188 (47) to be classified as pathologic FHR out of 1700 training data (426 testing data), with a misclassification of 16 (5).

Regarding to the specificity and sensitivity we might notice that:

- Normal FHR.

  - This model has a precision of 84.3% when it makes the classification predictions, and it is correct 84.3% of the time. Additionally, it predict successfully 97.6% of the classifications.
- Suspect FHR.
  - This model has a precision of 97.9% when it makes the classification predictions, and it is correct 97.9% of the time. Additionally, it predict successfully 79.24% of the classifications.
- Pathologic FHR.
  - This model has a precision of 99.2% when it makes the classification predictions, and it is correct 99.2% of the time. Additionally, it predict successfully 87.3% of the classifications.

Remark *The caret package using the "confusionMatrix" library has an interesting feature called: Cohen's Kappa Statistic which is commonly used to provide measure of how good two evaluators can classify data. Here the Kappa value is 0.83 which according to the strength table is a good classification.*

Tree Classification

The tree statistical methods used for studying data are set to perform classification and regression. Classification trees perform a structural 1 and 0 (yes and no) decisions that guide the method to classify the data based on that decisions with two important parts: branches representing attributes in the data, and leaves representing decisions.

To perform this analysis I used "rpart" library (function at the same time) setting the R routine to treat our DATABASE as a categorical classification setting the method by "class".

```
## Call:
## rpart(formula = CLASS ~ ., data = train, method = "class")
##   n= 1700
##
##           CP nsplit rel error    xerror       xstd
## 1 0.20810811      0 1.0000000 1.0000000 0.04598334
## 2 0.14054054      1 0.7918919 0.7918919 0.04208765
## 3 0.08783784      2 0.6513514 0.6513514 0.03886959
## 4 0.04864865      4 0.4756757 0.5000000 0.03470292
## 5 0.04054054      5 0.4270270 0.4486486 0.03307810
## 6 0.01261261      6 0.3864865 0.4108108 0.03179663
## 7 0.01081081      9 0.3486486 0.4027027 0.03151177
## 8 0.01000000     10 0.3378378 0.4027027 0.03151177
##
## Variable importance
## MSTV ASTV ALTV   DP   LB MLTV   AC   FM
##   29   22   20   17    4    4    2    1
##
## Node number 1: 1700 observations,    complexity param=0.2081081
##   predicted class=normal      expected loss=0.2176471  P(node) =1
##     class counts:  1330   134   236
##    probabilities: 0.782 0.079 0.139
##   left son=2 (1397 obs) right son=3 (303 obs)
##   Primary splits:
##       MSTV < 0.55   to the right, improve=154.01780, (0 missing)
##       ASTV < 59.5   to the left,  improve=138.16440, (0 missing)
##       ALTV < 13.5   to the left,  improve=112.45400, (0 missing)
##       AC   < 0.0015 to the right, improve= 87.27798, (0 missing)
##       DP   < 0.0015 to the left,  improve= 80.13913, (0 missing)
##   Surrogate splits:
##       ALTV < 22.5   to the left,  agree=0.886, adj=0.360, (0 split)
##       ASTV < 67.5   to the left,  agree=0.878, adj=0.314, (0 split)
```

```
##       LB   < 158.5  to the left,  agree=0.827, adj=0.030, (0 split)
##
## Node number 2: 1397 observations,    complexity param=0.1405405
##   predicted class=normal      expected loss=0.1095204  P(node) =0.8217647
##     class counts:  1244    80    73
##    probabilities: 0.890 0.057 0.052
##   left son=4 (1317 obs) right son=5 (80 obs)
##   Primary splits:
##       DP   < 0.0015 to the left,  improve=93.63242, (0 missing)
##       MLTV < 0.05   to the right, improve=42.87232, (0 missing)
##       ASTV < 59.5   to the left,  improve=26.30107, (0 missing)
##       AC   < 0.0025 to the right, improve=20.46261, (0 missing)
##       MSTV < 2.15   to the left,  improve=10.26813, (0 missing)
##
## Node number 3: 303 observations,    complexity param=0.08783784
##   predicted class=suspect     expected loss=0.4620462  P(node) =0.1782353
##     class counts:    86    54   163
##    probabilities: 0.284 0.178 0.538
##   left son=6 (35 obs) right son=7 (268 obs)
##   Primary splits:
##       ALTV < 68.5   to the right, improve=34.44407, (0 missing)
##       ASTV < 59.5   to the left,  improve=27.78114, (0 missing)
##       UC   < 0.0025 to the right, improve=16.70990, (0 missing)
##       MLTV < 3.95   to the left,  improve=16.56872, (0 missing)
##       MSTV < 0.45   to the right, improve=13.56004, (0 missing)
##   Surrogate splits:
##       MLTV < 4.05   to the left,  agree=0.931, adj=0.4, (0 split)
##
## Node number 4: 1317 observations,    complexity param=0.01261261
##   predicted class=normal      expected loss=0.06302202  P(node) =0.7747059
##     class counts:  1234    18    65
##    probabilities: 0.937 0.014 0.049
##   left son=8 (720 obs) right son=9 (597 obs)
##   Primary splits:
##       AC   < 0.0025 to the right, improve=9.187151, (0 missing)
##       LB   < 142.5  to the left,  improve=8.963167, (0 missing)
##       ALTV < 6.5    to the left,  improve=7.311500, (0 missing)
##       ASTV < 42.5   to the left,  improve=5.080932, (0 missing)
##       DL   < 0.0085 to the left,  improve=4.908537, (0 missing)
##   Surrogate splits:
##       ALTV < 1.5    to the left,  agree=0.669, adj=0.270, (0 split)
##       DL   < 0.0065 to the left,  agree=0.608, adj=0.136, (0 split)
##       MSTV < 0.85   to the right, agree=0.598, adj=0.114, (0 split)
##       ASTV < 59.5   to the left,  agree=0.595, adj=0.107, (0 split)
##       MLTV < 8.55   to the left,  agree=0.591, adj=0.097, (0 split)
##
## Node number 5: 80 observations,    complexity param=0.01081081
##   predicted class=pathologic  expected loss=0.225  P(node) =0.04705882
##     class counts:    10    62     8
##    probabilities: 0.125 0.775 0.100
##   left son=10 (11 obs) right son=11 (69 obs)
##   Primary splits:
##       ASTV < 25     to the left,  improve=7.122661, (0 missing)
##       MLTV < 5.3    to the left,  improve=4.683699, (0 missing)
```

```
##       LB   < 137    to the left,  improve=2.745401, (0 missing)
##       MSTV < 1.6    to the right, improve=2.410823, (0 missing)
##       DL   < 0.0015 to the right, improve=1.756025, (0 missing)
##   Surrogate splits:
##       FM   < 0.3475 to the right, agree=0.875, adj=0.091, (0 split)
##       MLTV < 13.1   to the right, agree=0.875, adj=0.091, (0 split)
##
## Node number 6: 35 observations
##   predicted class=pathologic  expected loss=0.08571429  P(node) =0.02058824
##     class counts:     3    32     0
##    probabilities: 0.086 0.914 0.000
##
## Node number 7: 268 observations,    complexity param=0.08783784
##   predicted class=suspect     expected loss=0.391791  P(node) =0.1576471
##     class counts:    83    22   163
##    probabilities: 0.310 0.082 0.608
##   left son=14 (61 obs) right son=15 (207 obs)
##   Primary splits:
##       ASTV < 59.5   to the left,  improve=28.62722, (0 missing)
##       ALTV < 7.5    to the left,  improve=18.35354, (0 missing)
##       UC   < 0.0065 to the right, improve=13.67833, (0 missing)
##       FM   < 0.0035 to the left,  improve=11.84845, (0 missing)
##       MSTV < 0.45   to the right, improve=11.10988, (0 missing)
##   Surrogate splits:
##       MSTV < 0.45   to the right, agree=0.813, adj=0.180, (0 split)
##       AC   < 0.0035 to the right, agree=0.791, adj=0.082, (0 split)
##       UC   < 0.0095 to the right, agree=0.780, adj=0.033, (0 split)
##
## Node number 8: 720 observations
##   predicted class=normal      expected loss=0.004166667  P(node) =0.4235294
##     class counts:   717     1     2
##    probabilities: 0.996 0.001 0.003
##
## Node number 9: 597 observations,    complexity param=0.01261261
##   predicted class=normal      expected loss=0.1340034  P(node) =0.3511765
##     class counts:   517    17    63
##    probabilities: 0.866 0.028 0.106
##   left son=18 (510 obs) right son=19 (87 obs)
##   Primary splits:
##       LB   < 142.5  to the left,  improve=17.732660, (0 missing)
##       ASTV < 41.5   to the left,  improve= 8.106155, (0 missing)
##       MLTV < 0.05   to the right, improve= 7.321643, (0 missing)
##       DL   < 0.0085 to the left,  improve= 5.494398, (0 missing)
##       ALTV < 6.5    to the left,  improve= 5.245311, (0 missing)
##
## Node number 10: 11 observations
##   predicted class=normal      expected loss=0.4545455  P(node) =0.006470588
##     class counts:     6     2     3
##    probabilities: 0.545 0.182 0.273
##
## Node number 11: 69 observations
##   predicted class=pathologic  expected loss=0.1304348  P(node) =0.04058824
##     class counts:     4    60     5
##    probabilities: 0.058 0.870 0.072
```

```
##
## Node number 14: 61 observations
##   predicted class=normal      expected loss=0.2295082  P(node) =0.03588235
##     class counts:    47     0    14
##    probabilities: 0.770 0.000 0.230
##
## Node number 15: 207 observations,    complexity param=0.04864865
##   predicted class=suspect     expected loss=0.2801932  P(node) =0.1217647
##     class counts:    36    22   149
##    probabilities: 0.174 0.106 0.720
##   left son=30 (19 obs) right son=31 (188 obs)
##   Primary splits:
##       ASTV < 79.5   to the right, improve=25.946510, (0 missing)
##       ALTV < 7.5    to the left,  improve=21.202540, (0 missing)
##       MLTV < 9.3    to the right, improve=12.507330, (0 missing)
##       UC   < 0.0055 to the right, improve=11.474170, (0 missing)
##       FM   < 0.0035 to the left,  improve= 6.033816, (0 missing)
##   Surrogate splits:
##       FM < 0.1905 to the right, agree=0.918, adj=0.105, (0 split)
##       DP < 0.002  to the right, agree=0.918, adj=0.105, (0 split)
##
## Node number 18: 510 observations
##   predicted class=normal      expected loss=0.08627451  P(node) =0.3
##     class counts:   466    17    27
##    probabilities: 0.914 0.033 0.053
##
## Node number 19: 87 observations,    complexity param=0.01261261
##   predicted class=normal      expected loss=0.4137931  P(node) =0.05117647
##     class counts:    51     0    36
##    probabilities: 0.586 0.000 0.414
##   left son=38 (71 obs) right son=39 (16 obs)
##   Primary splits:
##       ASTV < 59     to the left,  improve=10.754430, (0 missing)
##       UC   < 0.0035 to the right, improve= 3.044734, (0 missing)
##       LB   < 148.5  to the left,  improve= 2.432703, (0 missing)
##       ALTV < 27.5   to the right, improve= 1.393464, (0 missing)
##       MLTV < 9.35   to the left,  improve= 1.167535, (0 missing)
##   Surrogate splits:
##       FM   < 0.0025 to the left,  agree=0.839, adj=0.125, (0 split)
##       MLTV < 3.7    to the right, agree=0.839, adj=0.125, (0 split)
##
## Node number 30: 19 observations
##   predicted class=pathologic  expected loss=0.05263158  P(node) =0.01117647
##     class counts:     1    18     0
##    probabilities: 0.053 0.947 0.000
##
## Node number 31: 188 observations,    complexity param=0.04054054
##   predicted class=suspect     expected loss=0.2074468  P(node) =0.1105882
##     class counts:    35     4   149
##    probabilities: 0.186 0.021 0.793
##   left son=62 (30 obs) right son=63 (158 obs)
##   Primary splits:
##       ALTV < 7.5    to the left,  improve=21.855350, (0 missing)
##       MLTV < 9.3    to the right, improve=14.221260, (0 missing)
```

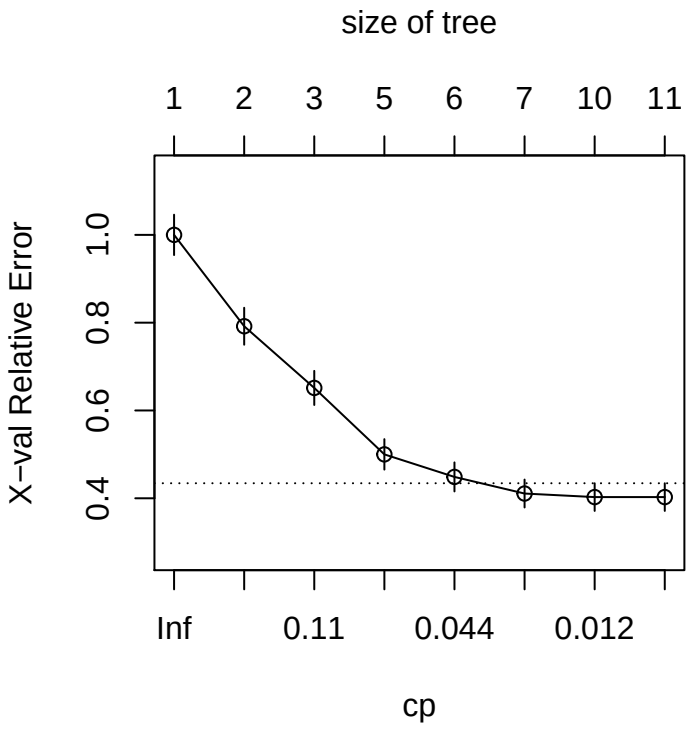
```
##       UC   < 0.0055 to the right, improve=11.630030, (0 missing)
##       LB   < 143.5  to the left,  improve= 5.458505, (0 missing)
##       FM   < 0.0025 to the left,  improve= 5.219000, (0 missing)
##   Surrogate splits:
##       MLTV < 9.3    to the right, agree=0.899, adj=0.367, (0 split)
##       LB   < 117    to the left,  agree=0.862, adj=0.133, (0 split)
##
## Node number 38: 71 observations
##   predicted class=normal      expected loss=0.2957746  P(node) =0.04176471
##     class counts:    50     0    21
##    probabilities: 0.704 0.000 0.296
##
## Node number 39: 16 observations
##   predicted class=suspect     expected loss=0.0625  P(node) =0.009411765
##     class counts:     1     0    15
##    probabilities: 0.062 0.000 0.938
##
## Node number 62: 30 observations
##   predicted class=normal      expected loss=0.2666667  P(node) =0.01764706
##     class counts:    22     1     7
##    probabilities: 0.733 0.033 0.233
##
## Node number 63: 158 observations
##   predicted class=suspect     expected loss=0.1012658  P(node) =0.09294118
##     class counts:    13     3   142
##    probabilities: 0.082 0.019 0.899
```

The summary expose each step showing the number of nodes, complexity parameter, class counts with their classification probabilities, and present the splits counts. For instance, analyzing the second node for 1397 observations the class count was: (class=Normal) 1244, (class=Suspect) 80, (class=Pathological) 73. Each of those with probabilities: 0.89, 0.057, and 0.052 respectively.

Performance

This method split the original root node dropping a relative error from 1.0 to 0.33784 where the root node is 0.33784. Next, analyzing the complexity parameter we might find the best complexity parameter value (CP) for this tree analysis. To select the optimal CP, I used "printcp" and "plotcp" function to check and visualize the CP with its relative error per node. Additionally, I check the RSQ loading "rsq.rpart" function. The CP plot evidence that the optimal number of nodes is 8 with an optimal CP equals to 0.01.

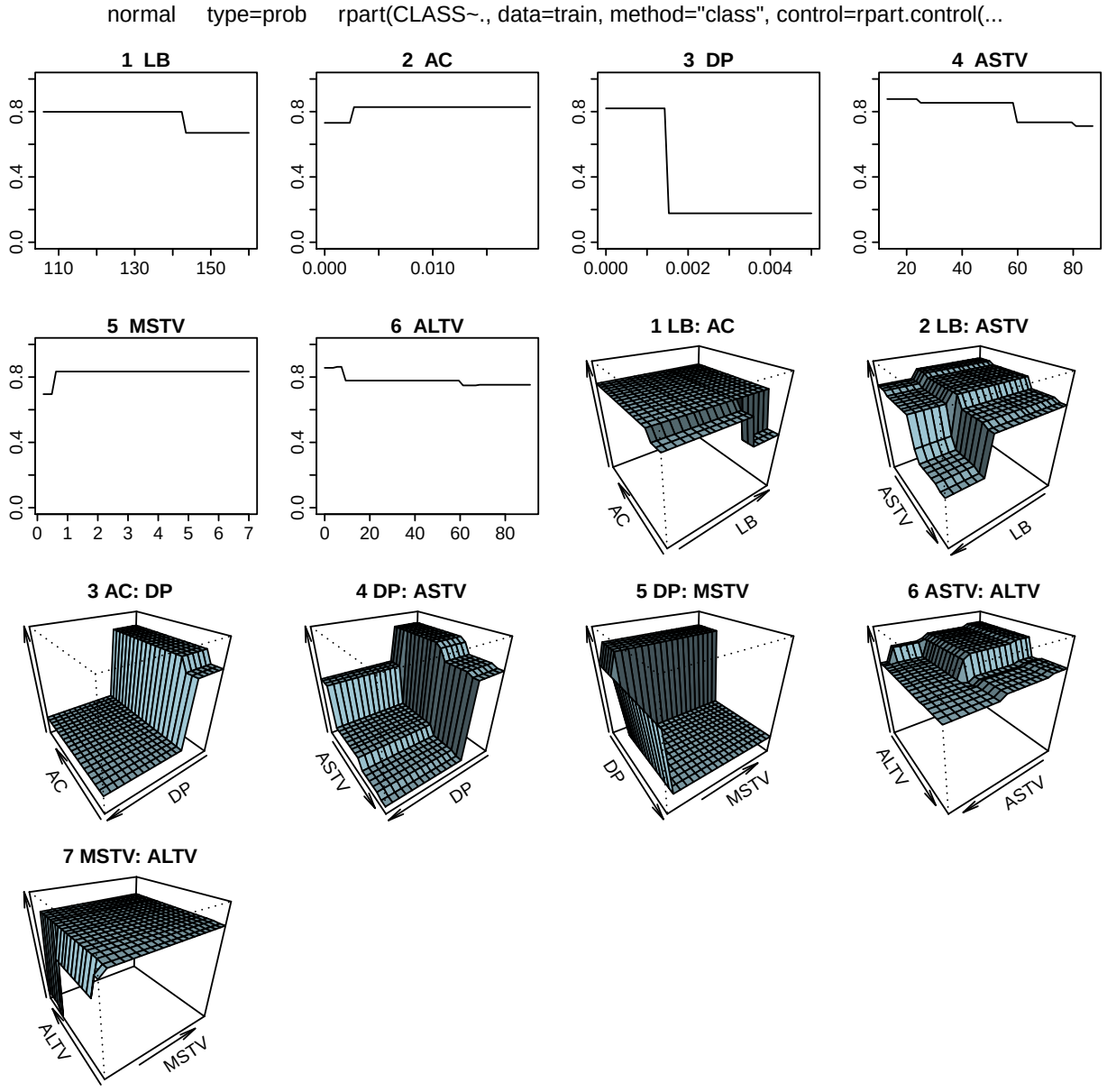
```
## Loading required package: Formula

## Loading required package: plotrix

## Loading required package: TeachingDemos
```



Additionally, I present a tree development plot representing all branches and leaves attributes by plotting not only linear classification process but also three dimensional plot (each node step as the third dimension) using the "plotmo'' library.

```
##
## predict.rpart[3,3]:
##         normal  pathologic     suspect
## 526  0.9958333 0.001388889 0.002777778
## 195  0.8245614 0.000000000 0.175438596
## 1842 0.9137255 0.033333333 0.052941176
##
## predict.rpart returned multiple columns (see above) but nresponse is not specified
##     Use the nresponse argument to specify a column.
##          Example: nresponse=2
##          Example: nresponse="pathologic"

## Warning: Defaulting to nresponse=1, see above messages

## calculating partdep for LB

## calculating partdep for AC

## calculating partdep for DP

## calculating partdep for ASTV

## calculating partdep for MSTV

## calculating partdep for ALTV

## calculating partdep for LB:AC 01234567890

## calculating partdep for LB:ASTV 01234567890

## calculating partdep for AC:DP 01234567890
```

```
## calculating partdep for DP:ASTV 01234567890
## calculating partdep for DP:MSTV 01234567890
## calculating partdep for ASTV:ALTV 01234567890
## calculating partdep for MSTV:ALTV 01234567890
```



This type of plot allow the reader to check the convergence in each node, and some important relations about the classified variables. Lastly, the confusion matrix using the optimal cp:

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal        314         10      20
```

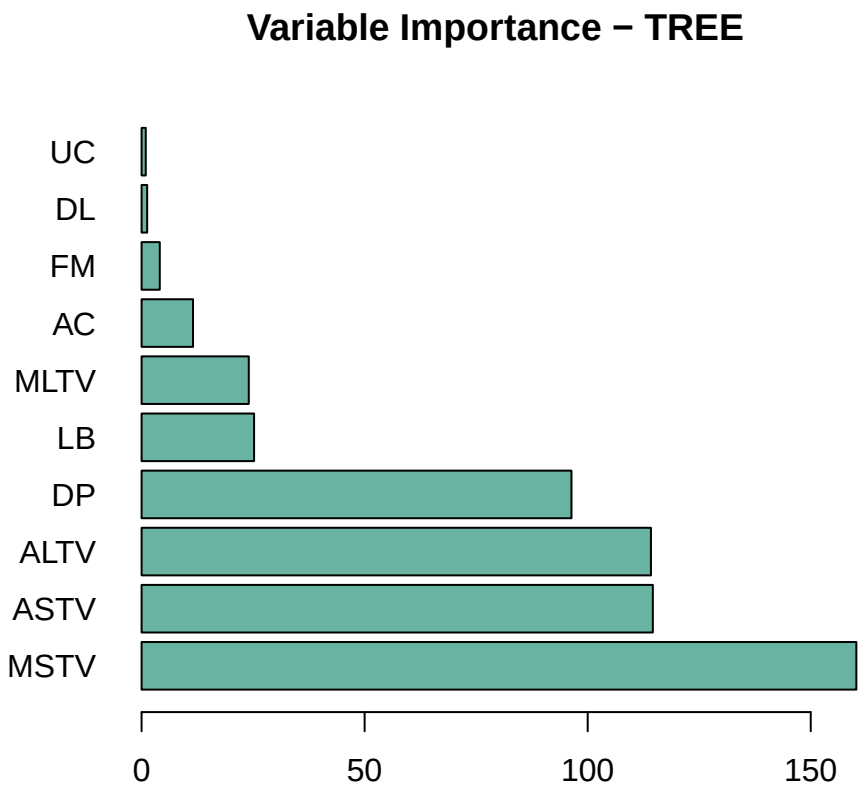
```
##   pathologic      8         32       2
##   suspect         3          0      37
##
## Overall Statistics
##
##                Accuracy : 0.8991
##                  95% CI : (0.8664, 0.926)
##     No Information Rate : 0.7629
##     P-Value [Acc > NIR] : 4.265e-13
##
##                   Kappa : 0.7206
##
##  Mcnemar's Test P-Value : 0.002008
##
## Statistics by Class:
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9662           0.76190        0.62712
## Specificity                 0.7030           0.97396        0.99183
## Pos Pred Value              0.9128           0.76190        0.92500
## Neg Pred Value              0.8659           0.97396        0.94301
## Prevalence                  0.7629           0.09859        0.13850
## Detection Rate              0.7371           0.07512        0.08685
## Detection Prevalence        0.8075           0.09859        0.09390
## Balanced Accuracy           0.8346           0.86793        0.80947
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   normal pathologic suspect
##   normal       1302         19      71
##   pathologic     14        112       8
##   suspect        14          3     157
##
## Overall Statistics
##
##                Accuracy : 0.9241
##                  95% CI : (0.9105, 0.9363)
##     No Information Rate : 0.7824
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7761
##
##  Mcnemar's Test P-Value : 5.777e-09
##
## Statistics by Class:
##
##                      Class: normal Class: pathologic Class: suspect
## Sensitivity                 0.9789           0.83582        0.66525
## Specificity                 0.7568           0.98595        0.98839
## Pos Pred Value              0.9353           0.83582        0.90230
## Neg Pred Value              0.9091           0.98595        0.94823
## Prevalence                  0.7824           0.07882        0.13882
## Detection Rate              0.7659           0.06588        0.09235
```

```
## Detection Prevalence         0.8188           0.07882         0.10235
## Balanced Accuracy            0.8679           0.91089         0.82682
```

Variable Importance and implementation plot

Classification trees perform the analysis using the data variables on at a time, decide and split the data. Then, the variable importance refers about the model variable use and accuracy over that variable. Usually, tree classification methods presents gini or information gain plots visualizing all the process locating the most important variable in the top.



The decision plot shows that the variable importance in classifying the data plotting the mean value of short term variability in first place. So, basically the tree classification method used this variable as the principal variable to grouped the data.

Recommendations

After reading the documentation about rpart plots, and related packages I could not find a nice rpart plot when the data have many inputs. So, I could find two solutions: quick solution and developing solution. The quick solution is gathering all the data on R, and using "graphviz' ' package on python which has more visual dependencies, and features. The developing option is creating better dependencies and libraries in R to visualize tree plots.

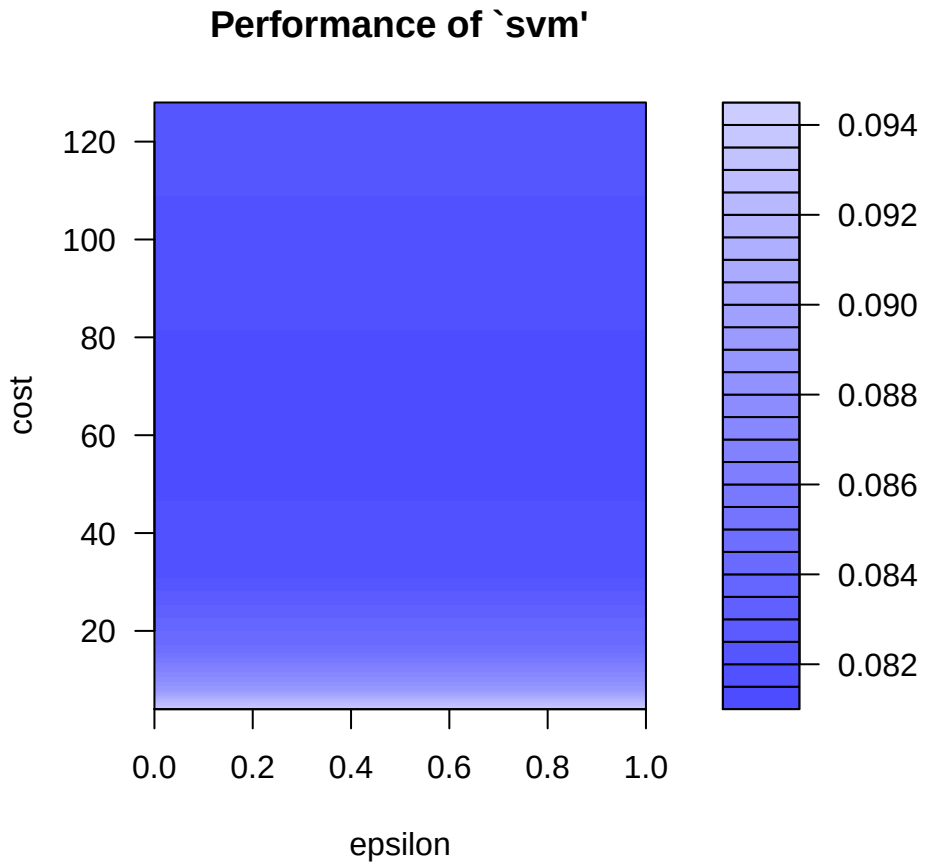
Supported Vector Machine

Here, I tried to use supported vector machine because of one important feature: overlapping data. The SVM method is optimal for generalizing the hyperplanes separation. How do we check for overlapping data? Well, the really descriptive allow us to check this type of overlapping behavior for two variable, and there would be importantly more overlapping characteristics for more than two variables.

To perform SVM analysis, I load the "e1071" R package with the "svm" function training a support vector machine and making the classification analysis. Before starting the process, as the classification variable was not numerical I made the conversion and used the "tune" implementation over a sequence of size 10, and cost exponentially big.

Performance

The resulting colorful analysis plot describes the best model using color coding where darker regions imply better performance (accuracy) in the implementation. This is a visual way to explore the tuning implementation for SVM. So, we can observe that the best epsilon runs from 0 to 1, while the optimal cost is in between 110 and 40.



According to the tune method, the best model is:

```
##
## Call:
## best.tune(method = svm, train.x = CLASS ~ ., data = train1, ranges = list(epsilon = seq(0,
##     1, 0.1), cost = 2^(2:7)))
```

```
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  64
##
## Number of Support Vectors:  374
##
##  ( 186 133 55 )
##
##
## Number of Classes:  3
##
## Levels:
##  normal pathologic suspect
```

Confusion matrix interpretation

To check the SVM performance classifying the data we use the confusion matrix. The confusion matrix shows that based on the training data (testing data) the SVM classify the entire data with an accuracy of 96.47% (91.08%). The model predicted 1317 (311) to be classified as normal FHR out of 1700 training data (426 testing data), with a misclassification of 13 (14). The model predicted 128 (38) to be classified as suspect FHR out of 1700 training data (426 testing data), with a misclassification of 108 (21). Lastly, the model predicted 195 (39) to be classified as pathologic FHR out of 1700 training data (426 testing data), with a misclassification of 61 (3).

Results and conclusions

I decided to stop the analysis, due to the fact the other implementations were not optimal. I tried the neural network, LDA, and k-means method for classification ant the MSE results were significantly high. I stop the classification analysis, and after implement the methods we find the following results:

Method	MSE Training	MSE Testing
Multinomial Log-Linear	0.1235	0.1314
KNN	0.058	0.0727
Tree Classification	0.924	0.899
SVM	0.0352	0.0892

Finally, I select two models: multinomial logistic regression and SVM. The SVM method above all the other methods choice was because of the distribution of the data. This type of analysis has a better performance for overlapping datasets.