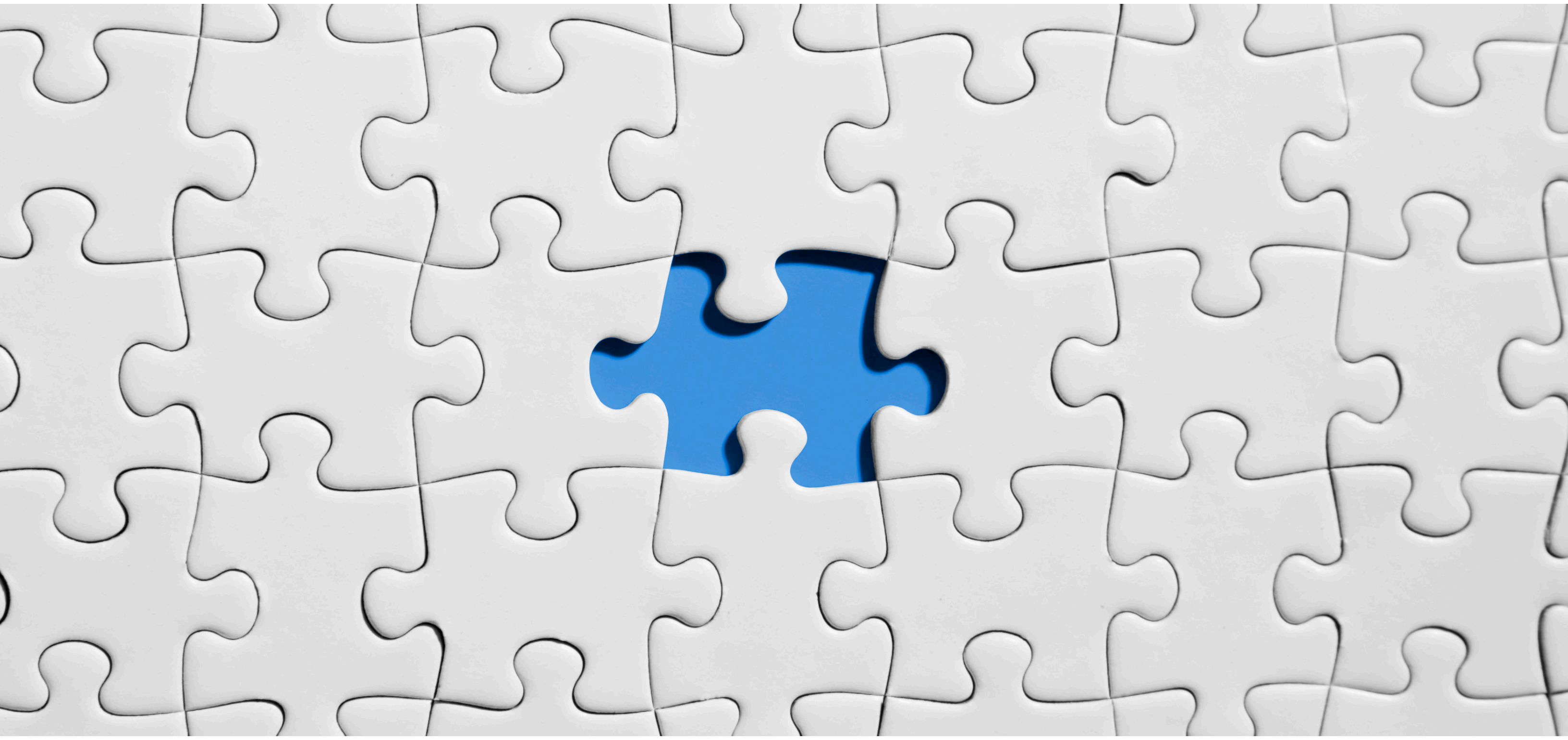
HUMAN FETUS HEALTH PREDICTION USING DECISION TREE



Rajasekhar Nannapaneni

Sr Principal Engineer, Solutions Architect

Dell EMC

Rajasekhar.nannapaneni@dell.com

Knowledge Sharing Article

Abstract

Human unborn baby health is monitored by electronic fetal monitor equipment which generates cardiotocographic data. This cardiotocographic data consists of fetal heart rate (FHR), uterus contractions rate, etc.

The inference achieved from analyzing the cardiotocographic data lets us know whether the fetus is normal, suspect with action needed or pathologic with immediate action needed. It is very important and critical to making this inference quickly as any delay could be a risk to both fetal and mother.

This article details the development of machine learning-based decision tree algorithm that will classify any given fetus health into normal, suspect and pathologic based on the given cardiotocographic data.

In this article, a dataset consisting of 2126 observations with 22 attributes or readings is considered and 90% of this data is used for training the decision tree model and the rest 10% is used to test the accuracy of the developed model.

Table of Contents

Disclaimer: The views, processes or methodologies published in this article are those of the author. They do not necessarily reflect Dell Technologies' views, processes or methodologies.

List of Tables

List of Figures

Abbreviations

Abbreviation	Full Form
FHR	Fetal heart rate
FM	Fetal movements/second
MSE	Mean Square Error

A1.1 Introduction

Healthcare for humans has been one of the top priorities and a lot of effort is spent improving diagnosis as well as treatments for various medical conditions. In the era of artificial intelligence, humans are equipped with new ways to improve healthcare and one such attempt was made in this article.

The part-A of this article gives a brief overview on cardiotocography and its associated monitoring. It gives an idea of how the monitoring is done and which critical observations are made.

Part-A of the article also discusses one of the popular machine learning algorithm called decision tree algorithm. A detailed overview of the mathematics involved in this algorithm is covered along with high level steps of the algorithm.

Part-B of this article emphasizes applying decision tree algorithm on a benchmark cardiotocographic data to identify the medical condition of the unborn fetus in the mother’s womb. The design, implementation and testing of the decision tree algorithm is performed on the cardiotocographic data and the test results are captured with appropriate analysis. The R code of the implemented algorithm is documented in the Appendix.

A1.2 What is cardiotocography?

During pregnancy, several factors contribute to the health of the fetus in the mother's womb. Problems can occur that affect the development of the baby due to conditions such as irregular blood pressure, infections, diabetes, etc. of the mother.

It is critical to monitor the health of the baby and this is done through electronic fetal monitoring system called cardiotocography. This device helps address and resolve complications that may impede fetal development.

This ultrasound-based device has 2 transducers; one helps to find the uterus pressure through observation of uterus contractions; the other monitors the heart rate of the baby. Though these 2 pivotal measurements, there are up to 22 observations or attributes by the device which could provide indirect information of the health of the baby.

Figure 1.1 shows the cardiotocography in action with the 2 transducers and their corresponding measurements.

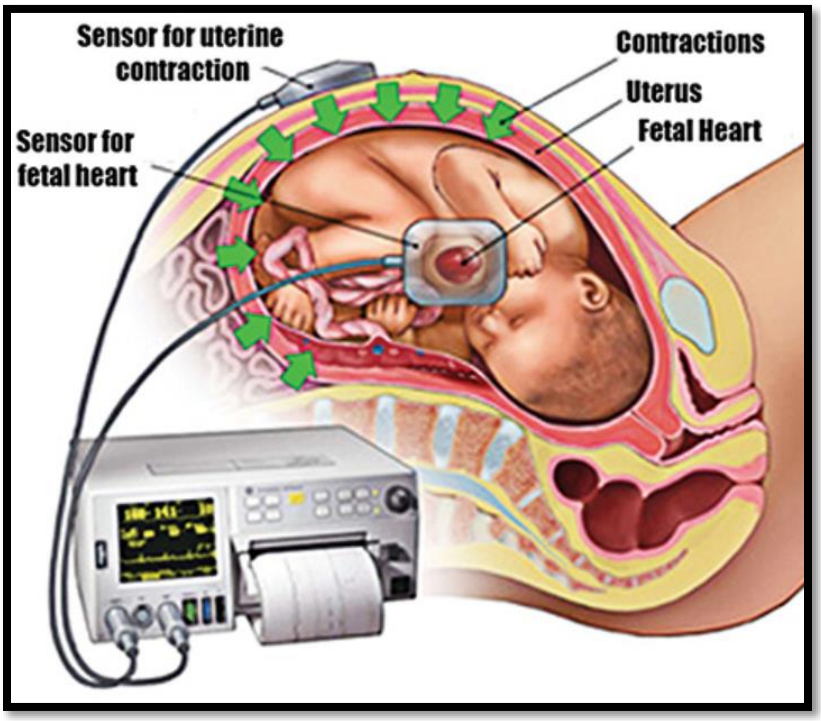


Figure 1.1: Cardiotocographic device

A1.3 What are decision trees?

Decision trees are machine learning algorithms that are mostly used for classification-based tasks. Decision trees are tree-like structured decision-making processes. In simpler terms, a decision tree follows the divide and conquer mechanism.

With classification as the objective, it’s imperative to divide the given dataset into classes and the division of the data into classes is dependent on the attributes of the data. In naïve terms one can consider them as recursive if-else conditions with an additional logical mechanism of deciding which attribute to consider making the division.

The decision process divides the dataset or attributes into subsets. Measures such as information gain and entropy from information theory plays a significant role in making the decisions of which attributes are primary at each step of the subdivision.

$$H(x) = -\Sigma\, p(x) \log p(x)$$

Where H(x) is entropy of x and p(x) is the probability of x. (1)

$$Gain = H(x) - [Weighted\ average * H(y)]$$

Where H(x) is entropy of x and H(y) is entropy of child. (2)

Equations (1) and (2) gives the formulas of entropy and information gain respectively. The attribute that has the most gain is considered the primary decision maker and the data is divided into sub sets based on this primary attribute. This process recursively divides the data with next best attribute based on the highest gain and so on.

Figure 1.2 shows a representation of inverse tree picture where the initial data can be considered at root node which is being sub divided into subsets of sub-tree data.

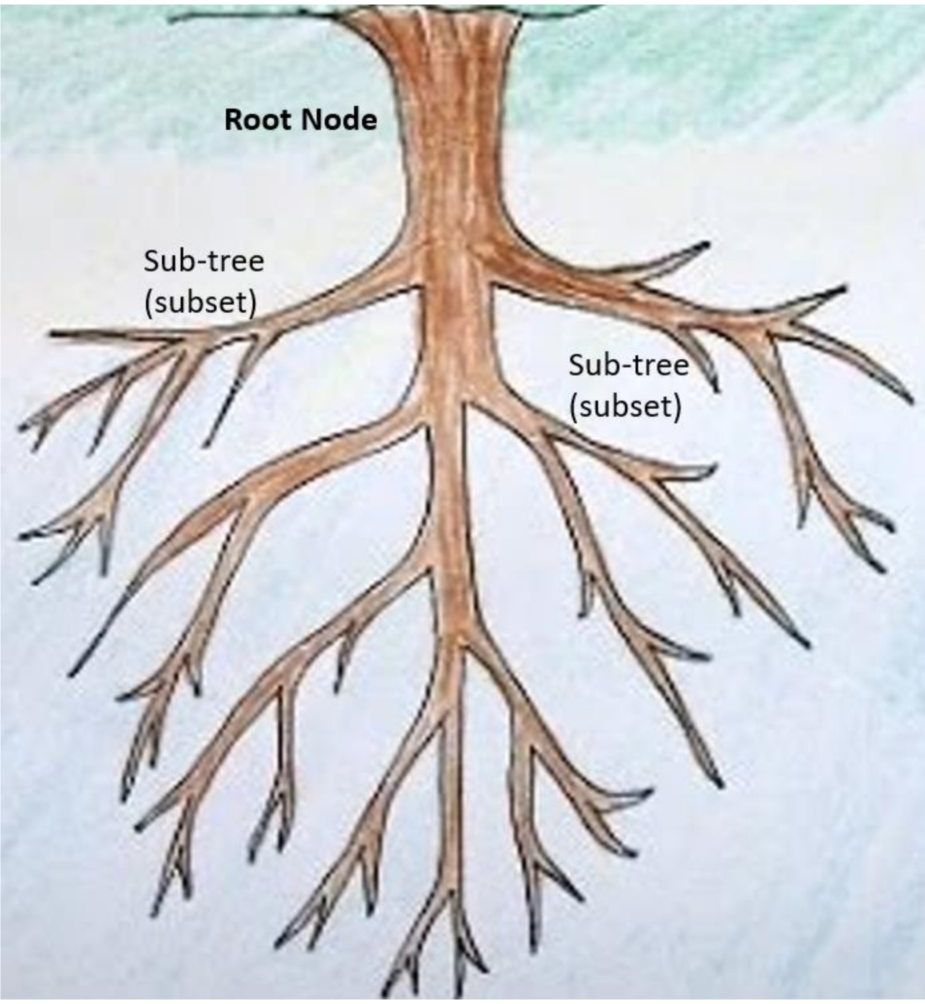


Figure 1.2: Decision tree representation

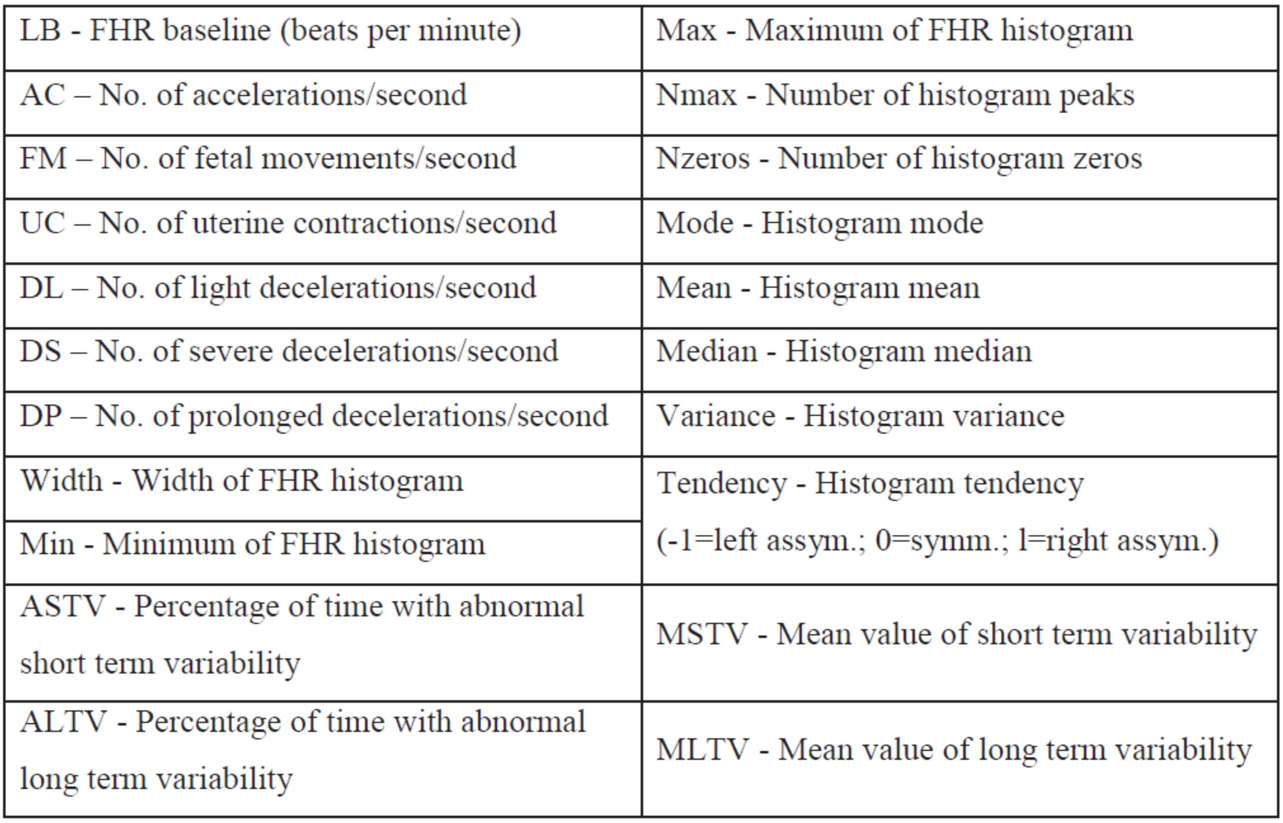
The decision tree implementation can be done in R language which has several different packages containing the decision tree function. In this paper, party package in R language is used to leverage the decision tree functionality.

B2.1 Decision tree classification for cardiotocographic data

Cardiotocographic data is usually analyzed by a doctor or a senior midwife. However, in this section, the decision tree is used as a classification tool for classifying the cardiotocographic data.

The dataset considered contains diagnosis results of cardiotocographic data which has the readings of fetal heart rate, fetal movements and uterine contractions that can allow doctors to decide whether given fetal readings are normal, suspect or pathologic.

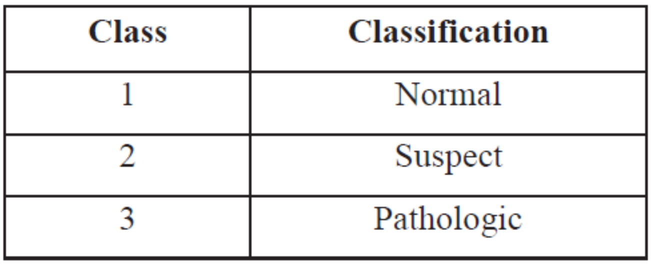
The dataset has 2126 observations with 22 attributes and 1 class attribute. The objective is to build a decision tree-based model which can be built by using training data and utilized as a classifier for test cardiotocographic data.



LB - FHR baseline (beats per minute)	Max - Maximum of FHR histogram
AC – No. of accelerations/second	Nmax - Number of histogram peaks
FM – No. of fetal movements/second	Nzeros - Number of histogram zeros
UC – No. of uterine contractions/second	Mode - Histogram mode
DL – No. of light decelerations/second	Mean - Histogram mean
DS – No. of severe decelerations/second	Median - Histogram median
DP – No. of prolonged decelerations/second	Variance - Histogram variance
Width - Width of FHR histogram	Tendency - Histogram tendency (-1=left assym.; 0=symm.; 1=right assym.)
Min - Minimum of FHR histogram	
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

Table 2.1: CTG Data Attributes

Table 2.1 lists all the attributes present in the cardiotocographic dataset and Table 2.2 lists the three classes in the data for classification of the fetal.

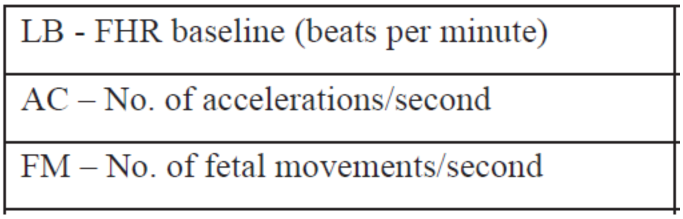


Class	Classification
1	Normal
2	Suspect
3	Pathologic

Table 2.2: CTG Data Classes

B2.2 Design for decision tree classification

The decision tree model for classification depends on the attributes in the data and the number of attributes in the given data is 23. As per the problem statement, only the first 3 attributes listed in Table 2.3 are taken for building the decision tree model.



LB - FHR baseline (beats per minute)
AC – No. of accelerations/second
FM – No. of fetal movements/second

Table 2.3: 3 CTG attributes used for training

The given dataset has 2126 observations and 90% of these observations can be used to train the decision tree model for classification. The remaining 10% can be used for testing the decision tree model that was built.

Figure 2.1 illustrates the decision tree classifier model that is being trained with 90% of the cardiotocographic dataset and tested with the remaining 10% of the cardiotocographic dataset. The decision tree varies per the attributes chosen to form the tree and, in this case, the first 3 attributes of the 23 attributes are used to build the tree.

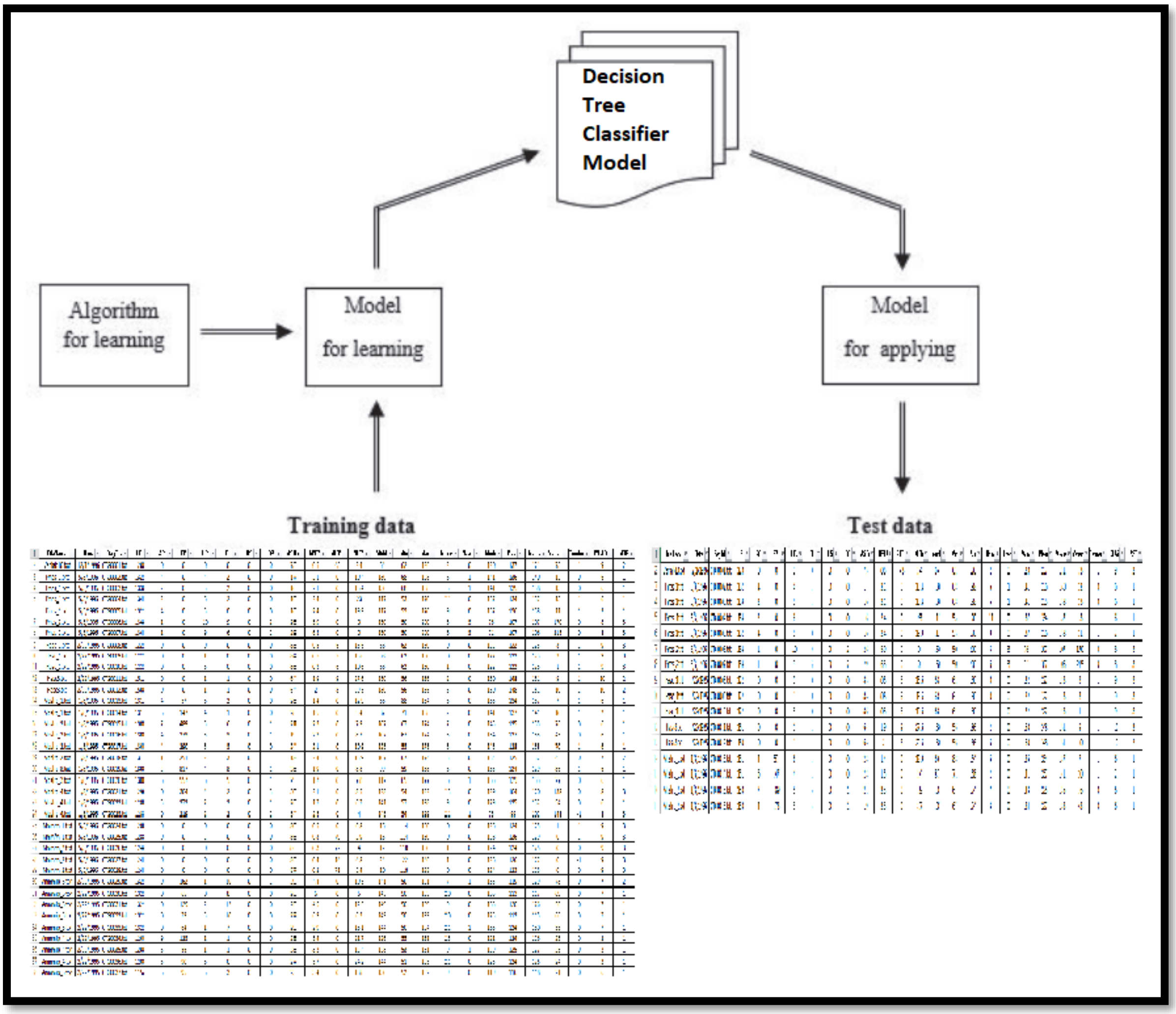


Figure 2.1: Decision Tree Classifier Model

B2.3 Implementation of decision tree classifier

The implementation of decision tree classifier for the cardiotocographic dataset is done using R language. Its code is mentioned in APPENDIX 1.

The cardiotocographic dataset available in “dataset_c.xlsx” Excel spreadsheet is read using “read_excel” command from “readxl” library in R language. Once the dataset is read, the observations are factorized into 3 classes; Normal, Suspect and Pathologic. Then the dataset is divided into 2 parts – training dataset and testing dataset.

A decision tree is formed using the “ctree” command from “party” package on the attributes LB, AC and FM which is shown in Figure 2.2.

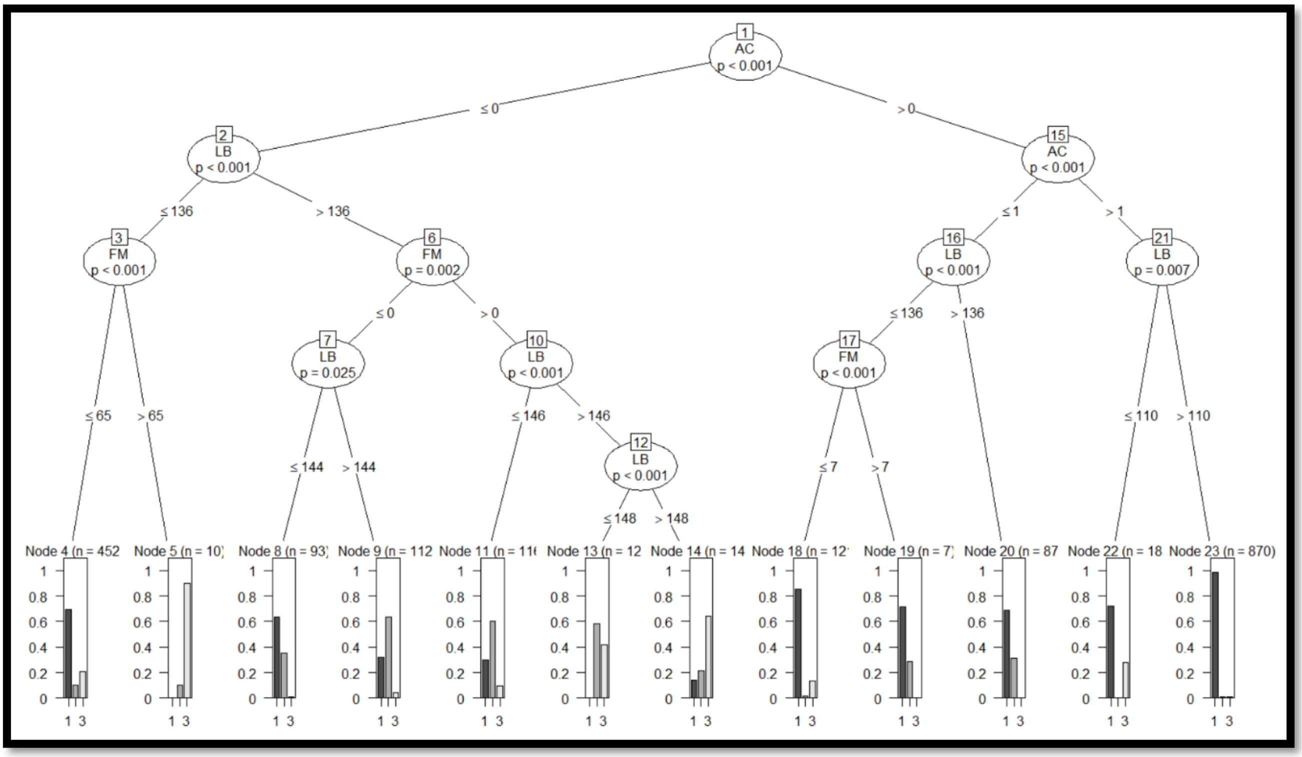


Figure 2.2: Decision tree formed using the attributes LB, AC and FM

As Figure 2.2 illustrates there is always one root node (AC in this case), several branched with internal nodes and end nodes called leaf. In this case, we have about 12 leaf nodes and each leaf node have a probability distribution of N (Normal), S (Suspect) and P (Pathologic) categorical information.

For example, the 1st leaf node has highest probability for N (Normal) category indicating that this leaf node is classified to be Normal. Similarly, each leaf node can be classified into one of the three categories – N (Normal), S (Suspect) and P (Pathologic) – depending on the probability assigned to each of the categories in the corresponding leaf.

APPENDIX 2 has R code for decision tree classifier after pruning, illustrated in Figure 2.3. The decision tree has been pruned using "minsplit" function under the condition of minimum criterion of 0.99 and tree must be split only if it has support from at least 500 observations.

The new decision tree after pruning in Figure 2.3 is simpler and has less leaf nodes of 5. Again, each leaf node has N (Normal), S (Suspect) and P (Pathologic) categorical information used to categorize or classify a given node into one of the 3 categories.

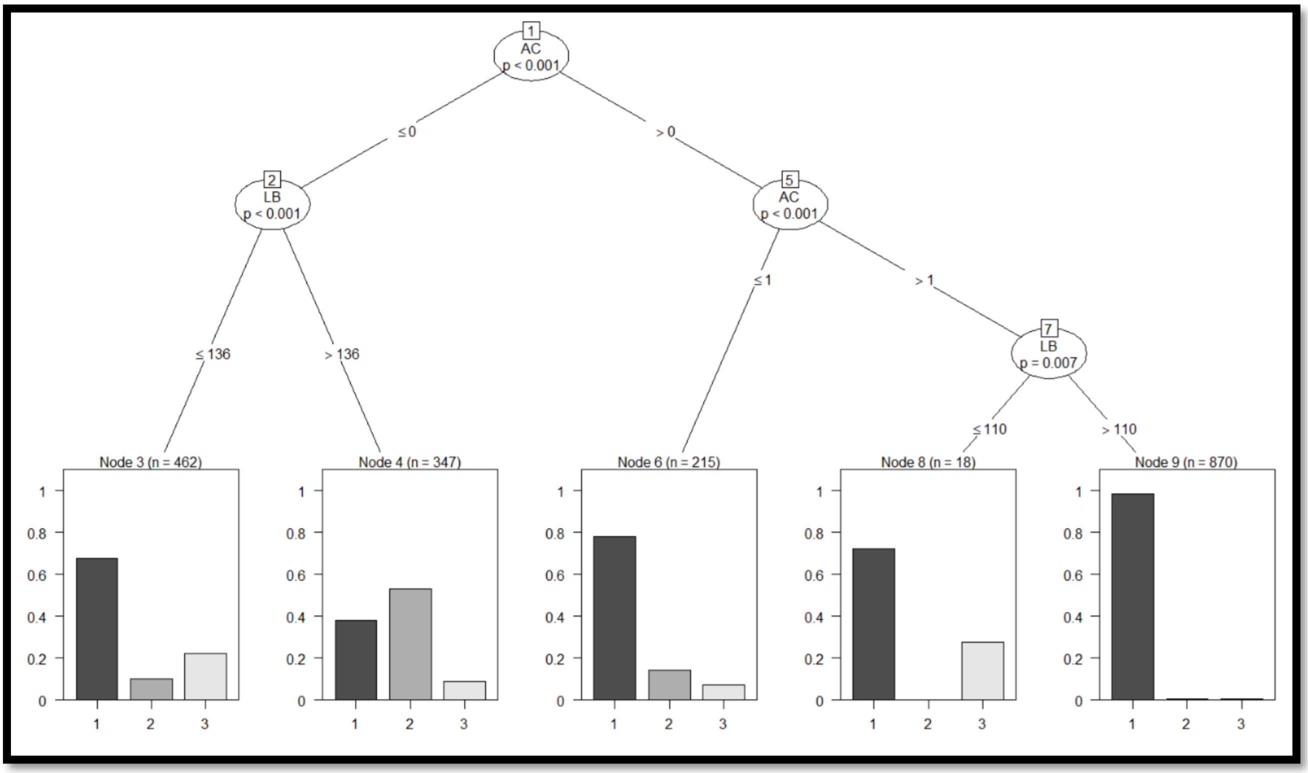


Figure 2.3: Decision tree formed using the attributes LB, AC and FM after pruning

B2.4 Testing the decision tree for cardiotocographic data

The R code in APPENDIX 2 has the variables whose definitions are mentioned in Figure 2.4.

df - Entire dataset with 2126 observations and 23 attributes

train - Training dataset which is 90% of the data from df/entire dataset

test - Testing dataset which is 10% of the data from df/entire dataset

tree_train - Trained decision tree model for classification

train_pred - Predicted classification information for training data

err_table1 - Training data classification results

train_err - Training data classification error

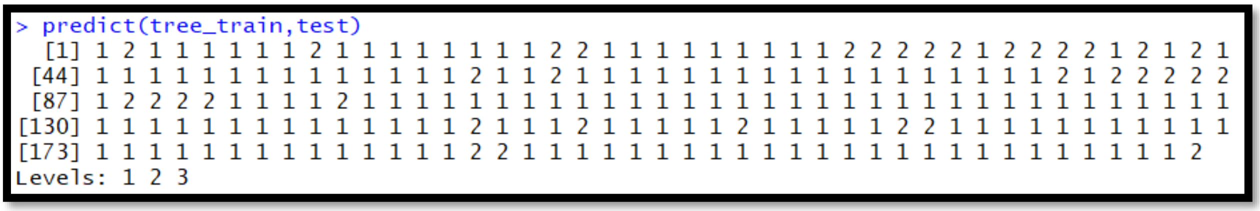
test_pred - Predicted classification information for testing data

err_table2 - Testing data classification results

test_err - Testing data classification error

Figure 2.4: Variables in R code in APPENDIX 2

In testing phase, the objective is to apply the 10% of the cardiotocographic test data (test variable per Figure 2.4) on the trained model (tree_train as per Figure 2.4). The decision tree model classifier output for test data is shown in Figure 2.5.



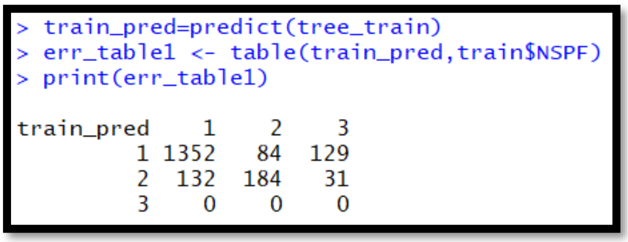
```
> predict(tree_train,test)
  [1] 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1
 [44] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2
 [87] 1 2 2 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[130] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1
[173] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
Levels: 1 2 3
```

Figure 2.5: Decision tree model classifier output for test data

Output in Figure 2.5 indicates classified class output for each of the 10% observations of the overall data.

B2.5 Result Tabulation of the decision tree classification

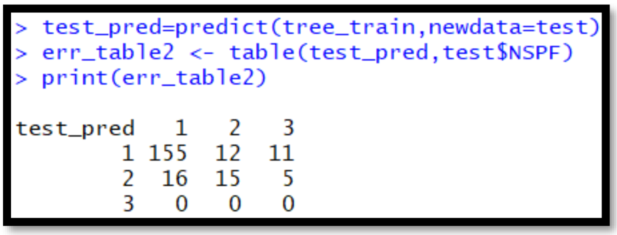
In this section, the results of training data class prediction are compared with actual classes of observations in training data. Also, the results of testing data class prediction are compared with actual classes of observations in testing data.



```
> train_pred=predict(tree_train)
> err_table1 <- table(train_pred,train$NSPF)
> print(err_table1)

train_pred    1    2    3
         1 1352   84  129
         2  132  184   31
         3    0    0    0
```

Figure 2.6: Tabular output of prediction results for training data

Figure 2.6 depicts the tabular output of the prediction of training data classes with respect to actual classes in training data.



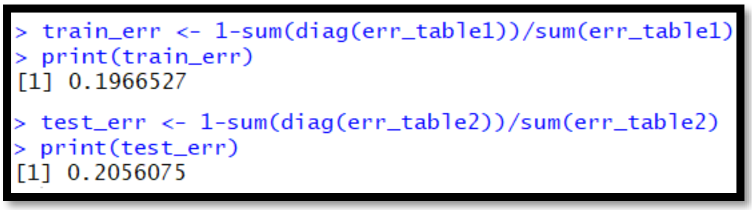
```
> test_pred=predict(tree_train,newdata=test)
> err_table2 <- table(test_pred,test$NSPF)
> print(err_table2)

test_pred   1   2   3
        1 155  12  11
        2  16  15   5
        3   0   0   0
```

Figure 2.7: Tabular output of prediction results for testing data

Figure 2.7 depicts the tabular output of the prediction of testing data classes with respect to actual classes in testing data.

B2.6 Performance evaluation of the decision tree classifier

The performance analysis for decision tree classifier model for both the training data and testing data is obtained by summing the diagonal elements of the tabular formats divided by the sum of elements of the entire table.



```
> train_err <- 1-sum(diag(err_table1))/sum(err_table1)
> print(train_err)
[1] 0.1966527
> test_err <- 1-sum(diag(err_table2))/sum(err_table2)
> print(test_err)
[1] 0.2056075
```

Figure 2.8: Performance analysis of decision tree classifier

Figure 2.8 shows the training data classification error is 19.66 % when the classes are predicted using decision tree classifier model and the testing data classification error is 20.56 % when the classes are predicted using decision tree classifier model.

Hence the accuracy of the decision tree classification during training is 80.34% and accuracy during testing is around 80%.

Algorithm performance could have been improved if the training data had more 3rd class (Pathologic) data so that the 3rd class could have been better classified. Hence, with the given dataset and training data, the algorithm achieved reasonable performance.

B2.7 Comments on results

The error on the predicted results for testing data is about 20.56% which appears to be optimum after modifying various parameters such as minsplit, ratio of training to testing data, etc. The 3rd class (Pathologic) in the data was not classified properly as it resulted in 0 in Figure 2.6 and Figure 2.7. This could be due to insufficient training data for 3rd class (Pathologic). Classification accuracy depends on the trained model and trained model accuracy depends on the trained data. As long as trained data has sufficient information to train a model, model accuracy can’t be improved. Thus the 3rd class (Pathologic) didn’t get classified due to this.

B2.8 Conclusion

Decision tree classifier model is a very flexible and elegant way to classify data into classes. It uses Naïve Bayes probabilistic ideology in the background and provides a structural and logical way to clearly represent the given data into classes. It can be concluded that the given cardiotocographic data is decently classified with ~20% testing error and provides reasonable insights into classification of fetal health status.

References

"[1] Chen, C. L. Philip and Chun-Yang Zhang. (2014) "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data." Inf. Sci. 275: 314-347."

"[2] Datameer. (2018). [ONLINE] Available at: https://www.datameer.com/pdf/Datameer-Customer-Analytics-ebook.pdf. [Accessed 3 July 2018]."

"[3] Fung, Han Ping. (2013). Using Big Data Analytics in Information Technology (IT) Service Delivery. Internet Technologies and Applications Research. 1. 6-10. 10.12966/itar.05.01.2013."

"[4] Kamath, Rajani & Kamat, Rajanish. (2018). Modeling fetal morphologic patterns through cardiotocography data: Decision tree-based approach. Journal of Pharmacy Research."

"[5] Karabulut, Esra & Ibrikci, Turgay. (2014). Analysis of Cardiotocogram Data for Fetal Distress Determination by Decision Tree Based Adaptive Boosting Approach. Journal of Computer and Communications. 02. 32-37. 10.4236/jcc.2014.29005."

APPENDIX

APPENDIX 1 – R code for decision tree classifier of cardiotocographic dataset

```
library("readxl")
library(“party”)

df<-read_excel("C:/Users/Desktop/DM/dataset_c.xlsx")
df$NSPF <-factor(df$NSP)
set.seed(4321)

dp <- sample(2,nrow(df),replace=TRUE,prob=c(0.9,0.1))

train <- df[dp==1,]
test <- df[dp==2,]
tree_train <- ctree(NSPF~LB+AC+FM, data=train)
plot(tree_train)
```

APPENDIX 2 – R code for decision tree classifier of cardiotocographic dataset with pruning

```
library("readxl")
library(party)

df<-read_excel("C:/Users/Desktop/DM/dataset_c.xlsx")
df$NSPF <-factor(df$NSP)
set.seed(4321)

dp <- sample(2,nrow(df),replace=TRUE,prob=c(0.9,0.1))
train <- df[dp==1,]
test <- df[dp==2,]

tree_train <- ctree(NSPF~LB+AC+FM, data=train, controls=ctree_control(mincriterion =0.99,minsplit = 500))
plot(tree_train)
predict(tree_train,test,type="prob")
predict(tree_train,test)

#training data classification error
train_pred=predict(tree_train)
err_table1 <- table(train_pred,train$NSPF)
print(err_table1)
train_err <- 1-sum(diag(err_table1))/sum(err_table1)
print(train_err)

#test data classification error
test_pred=predict(tree_train,newdata=test)
err_table2 <- table(test_pred,test$NSPF)
print(err_table2)
test_err <- 1-sum(diag(err_table2))/sum(err_table2)
print(test_err)
```