

Comparison of Classification Methods for the  
Fetal Health Cardiotocograms Data

STAT 6440 Data Mining Group 5 Project

Ashra Regmi, Mary Irungu, Pragya Khanal, Shakhnoza Takhirova

BGSU

## Table of Contents

Abstract .....	3
Introduction .....	4
Dataset .....	5
Methodology .....	7
Results .....	8
Multinomial logistic regression .....	8
Multinomial logistic regression with PCA .....	9
KNN .....	10
Decision Tree .....	10
Random Forest .....	11
Bagging .....	11
Neural Networks .....	11
Overall Results .....	12
Future Work .....	14
References .....	15
Tables .....	16
Figures .....	18

### **Abstract**

This paper's main goal is to build classification models to identify the status of pathological risk associated with the developing fetus using cardiotocograms data. To perform the analysis, data provided by UCI Machine Learning Repository was used. To solve multinomial classification problem, methods such as multinomial logistic regression, PCA applied multinomial logistic regression, KNN, Decision tree, Random Forest, Bagging and Neural networks were implemented to predict if the fetus with given cardiotocogram is developing abnormalities. Since the dataset was imbalanced, an oversampling method was used to get balanced data set for the models. Each model's performance was evaluated, and the best result was obtained from Random Forest, with respect to the kappa value.

*Keywords:* Cardiotocograms data, oversampling, multinomial classification, multinomial regression, PCA, KNN, decision tree, random forest, bagging, neural networks.

## Introduction

One of the key goals for the UN's Sustainable Development Goals is good health and well-being. As stated by (UNICEF, 2022, para.1), "Health and well-being are important at every stage of one's life starting from the beginning". To uphold this goal, we conducted an analysis of fetal health data during pregnancy. The aim is to detect early fetal development abnormality, to initiate medical intervention and reduce infant mortality and thus observe good health and well-being of the fetus and seemingly healthy newborns.

During pregnancy, obtaining direct information about the health of the fetus is not easy. There are many diagnostic tools to aid in obtaining that information. One of them is Cardiotocography (CTG), also called EFM, electronic fetal monitoring. Cardiotocography is a widely used accessible cost-effective diagnostics technique used to monitor the condition of a fetus in various stages of development and growth. Cardiogram (CTG) records the measures obtained from fetal heart rate and uterine contractions of the mother. Conventionally, obstetricians evaluate cardiotocograms by visual inspection. This evaluation produces inconsistent diagnostic conclusions among obstetricians. One of the implications of this are missing cerebral palsy cases due to false negatives and increase in cesarean sections due to false positives (Ogasawara et.al, 2021). An algorithm with high accuracy and efficiency would aid obstetricians to form more objective diagnostic conclusions and make better medical decisions.

This paper has been presented in four sections. In the first section, there is information about the dataset, the preprocessing methods that were used are discussed. In the second section, we have techniques of all seven algorithms that have been presented along with their results. The third section compares the model performance from all the obtained results and finally we have a brief conclusion from our findings.

## Dataset

Fetal Health Cardiotocograms data is obtained from the University of California Irvine Machine Learning Repository (Dua & Graph, 2019). It consists of diagnostic feature measures that were automatically extracted from 2126 fetal cardiotocograms (CTGs). This data set is originally in XLS format and allows to solve classification with respect to a morphologic pattern (A, B, C. ...) and to a fetal state (N, S, P), i.e., 10 class and 3 class classification problem. We used pre-processed version of this data set obtained from Kaggle (Larxel, 2020). This version omits morphologic pattern variables and is in CSV format.

As some variables had long names, and caused difficulties in visualization, they were renamed to shorter names as shown in table 1.

Understanding a given dataset is accessible only to experts who have relevant medical training. Nevertheless, extraction of a few important predictors of the data can be understood from images included in the figures section.

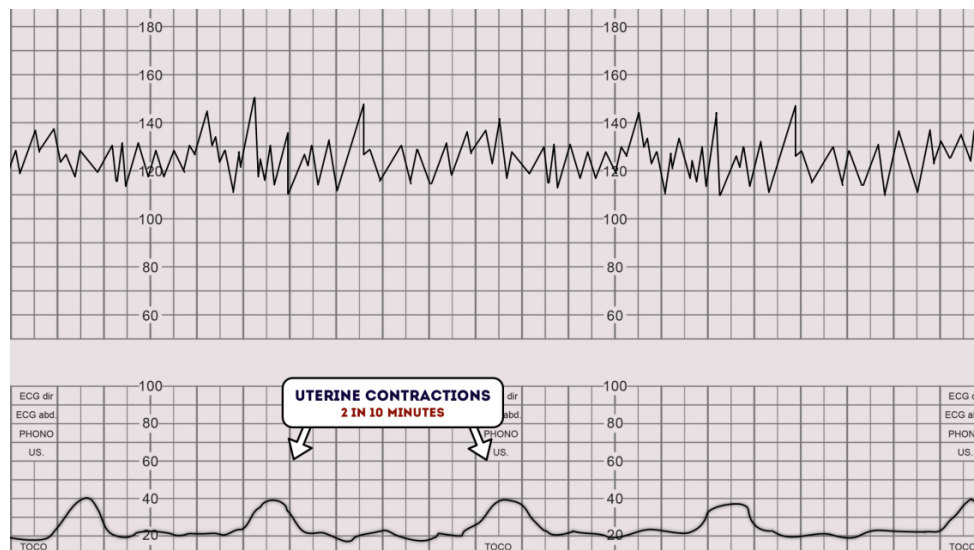


Figure 1: Uterine contraction [Image] (Potter.L, 2021)

All predictors in this dataset are numeric. The response variable *fetal\_health* is categorical with three categories: *Normal*, *Suspicious* and *Pathological*. The dataset is imbalanced.

Table 2: Imbalanced Dataset

Normal	Suspect	Pathological
1655	295	176

Applying classification models on imbalanced data might lead to misleading results. There are many ways to address imbalance in the data such as changing cut off values, undersampling, oversampling and SMOTE. The latter one is a popular method. In an attempt to balance the data, the experiments ran after applying SMOTE presented unsatisfactory results. This could be indicative of needing further investigation of the efficiency of SMOTE for multinomial data needs further researching. Undersampling reduced the data size significantly, and this also led to poor results. On the other hand, oversampling produced satisfactory results. The results presented here are obtained by applying oversampling to the dataset which increased dataset size significantly. *upSample()* function from *caret* package was used to carry out oversampling. Dataset after oversampling:

Table 3: Balanced Dataset

Normal	Suspect	Pathological
1655	1655	1655

Oversampled data was split into training and test datasets in 7:3 ratios.

The current dataset has a significant level of collinearity between variables as it can be seen in correlations plot (see Figure 8). Methods used in this paper can handle multicollinearity.

## Methodology

In this section, a brief overview of the methods used has been presented.

### *Multinomial Logistic Regression*

Multinomial Logistic Regression is a regression model in which the dependent variable or the response variable can have more than two classification levels. It is a generalization of logistic regression model which uses one class as a reference class to calculate odds:

$$\log \left( \frac{P(X = X_i)}{P(X = X_1)} \right) = \sum_{j=1}^p \beta_{ij} x_j, \quad i = 2, 3, \dots, k$$

$k$  is the number of classes,  $X_i$  is  $i$ -th class,  $p$  is number of predictors,  $\beta_{ij}$ -is logistic regression coefficient for the  $i^{th}$  class,  $j^{th}$  variable.

### *Multinomial Logistic Regression with PCA*

Principal Component Analysis (PCA) is a dimension reduction technique that can be used to handle a large set of variables. It is designed to work with numerical variables. In PCA, new variables are created as weighted linear combinations of the original variables. The new variables, also known as principal components, thus created have zero correlation with each other (Sarkar, n.d).

### *K-nearest neighbor*

K-nearest neighbor (KNN) is a non-parametric supervised model that classifies new data points based on the nearest neighbor data points. The idea of KNN is to identify  $k$  records (neighbors) in the training dataset that are similar to a new record to be classified. It will assign the class of the majority of the neighbors (Sarkar, n.d).

### *Decision Tree*

The idea is to create a model that predicts the value of target variables by learning simple decision rules inferred from the data features. The decision is made by splitting the root node of a

feature into child nodes until reaching terminal nodes with all observations belonging to the same class. This predictive model uses recursive partitioning until a decision is made on the classification of a new observation. (Kumar, Gokul S., 2020)

### ***Random Forest***

Random Forest is an ensemble learning algorithm that consists of many decision trees on bootstrap samples from the training data. The idea is to reduce multicollinearity by creating de-correlated trees. It chooses from a random subset of variables to make a new split each time. The classification or selection of the final output follows a majority voting system.

### ***Bagging***

Bagging (also known as bootstrap aggregating) is another ensemble learning algorithm that fits the predictive model in B bootstrap samples. The final prediction in Bagging is an average of the predictions obtained from the B models fitted on bootstrap samples. For the classification of the new observation, the majority class label of the bootstrap samples is chosen.

### ***Neural Networks***

Neural Network is a data modeling tool that is used to model complex relationships between predictors and response variables and find patterns in large sets of data. It is based on a model of biological activity in the brain, where neurons are interconnected and learn from experience (Shmueli, et.al, 2018).

## **Results**

### **Multinomial logistic regression.**

For multinomial regression, *multinom()* function from *nnet* package was used as other functions to carry out logistic regression, such as *mlogit()*, require reshaping data. The multinomial model for the given data set consists of a system of the two following equations:



$$\log\left(\frac{P(X=Suspicious)}{P(X=Normal)}\right) = -15.561 + 0 * base.val -1103.941 * accel + 12.648 * fetal\_move -186.26 * \\
uter.cont -403.922 * light.decel -13.342 * sev.decel + 177.55 * \\
prolong.decel + 0.071 * abn.short.var -0.444 * mean.short.var + 0.013 * \\
perc.tlong.var -0.095 * mean.long.var + 0.009 * hist.width + 0.017 * \\
hist.min + 0.026 * hist.max + 0.199 * hist.npeaks -0.054 * hist.n0 -0.051 \\
* hist.mode + 0.189 * hist.mean -0.086 * hist.median + 0.058 * hist.var \\
+ 0.512 * hist.tend$$

$$\log\left(\frac{P(X=Pathological)}{P(X=Normal)}\right) = -16.38 + 0.305 * base.val -848.682 * accel + 18.213 * fetal\_move -220.67 * \\
uter.cont -241.809 * light.decel + 13.567 * sev.decel + 692.402 * \\
prolong.decel + 0.156 * abn.short.var -0.794 * mean.short.var + 0.063 \\
* perc.tlong.var -0.072 * mean.long.var + 0.018 * hist.width + 0.022 * \\
hist.min + 0.04 * hist.max -0.28 * hist.npeaks + 0.436 * hist.n0 + 0.001 \\
* hist.mode -0.03 * hist.mean -0.293 * hist.median + 0.08 * hist.var + \\
0.437 * hist.tend$$

The model converged in 360 iterations having negative log likelihood of 1117.951167. *Normal* category was used as baseline category, and the model calculates relative risks of being *Pathological* or *Suspicious* health status over being *Normal*.

Table 4: Multinomial model's overall test metrics

Accuracy	0.8710544
Kappa	0.8066109

### Multinomial logistic regression with PCA.

To address multicollinearity, PCA dimension reduction method was used to transform the training dataset. PCA was carried out using *preprocess=c("pca")* option in *caret* package and applied to multinomial logistic regression. Through 5-fold cross validation, 13 out of 22 principal components were selected which explains 95.6% of the variance. Through PCA significant dimension reduction was achieved. (Figure 9).

When multinomial logistic regression was applied to PCA, the accuracy and kappa significantly improved.

Table 5: PCA Multinomial model's overall test metrics

Accuracy	0.9106783
Kappa	0.8660118

## KNN

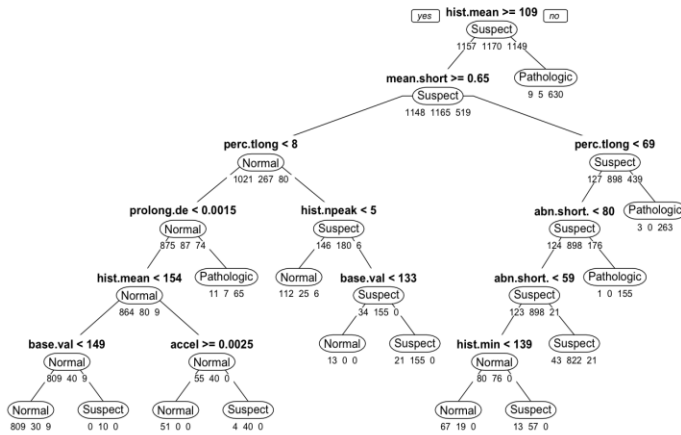
KNN model was trained using *train()* function of the *caret* package with *method="knn."* Centering and scaling of the test data was carried out. To select *k*, 10 parameters were tested using cross validation and *k=5* was chosen as the most optimal parameter.

Table 6: KNN model overall test metrics

Accuracy	0.9308261
Kappa	0.8962383

## Decision Tree

Figure 2: Decision Tree



Decision Tree model was trained using *train()* function of the *caret* package with *method="rpart"*. The complexity parameter was chosen using 5-fold cross validation. At *cp=0.00346*, decision tree attained highest kappa.

*abnormal\_short\_term\_variability* and *percentage of time with abnormal long term variability* were selected as the most important 2 predictors.

Table 7: Decision Tree model's overall test metrics

Accuracy	0.9207522
Kappa	0.8811304

### Random Forest.

Random Forest model was trained using *train()* function of the *caret* package with *method="rf"*. The number of predictors in a subset to be used at each split were selected using 5-fold cross validation, and *mtry=4* was chosen as the most optimal one to attain highest kappa.

*percentage of time\_with\_abnormal\_long\_term\_variability* and *abnormal\_short\_term\_variability* were selected as most important predictors.

Table 8: Random Forest model's overall test metrics

Accuracy	0.9858966
Kappa	0.9788450

### Bagging

Bagging model was trained using *train()* function of the *caret* package with *method="treebag"* option. Number of bootstrap samples to build trees on were set to *nbagg=50* and increasing number of bags did not improve the results.

*abnormal\_short\_term\_variability* and *percentage of time with abnormal long term variability* were selected as the most important predictors.

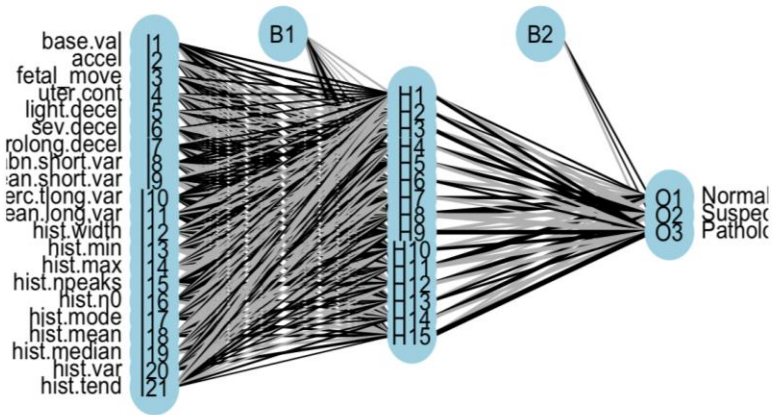
Table 9: Bagging model overall test metrics

Accuracy	0.9845534
Kappa	0.9768303

### Neural Networks

Figure 3: Neural Newtork

Neural Networks model was trained using *train()* function of the *caret* package with *method="nn"*. Number of hidden nodes and decay parameters were chosen via 5-fold cross validation. One



important fact to note is that the model was choosing the highest tuning parameter from the tuning grid for the number of hidden nodes, as the tuning grid was expanded. During tests, tuning grid was expanded until 30, and it

choose the maximum. Due to its high computational cost, max number of allowed hidden nodes were kept at 15.

Table 10: Neural Networks model's overall test metrics

Accuracy	0.9818670
Kappa	0.9728001

### Overall Results

It is interesting to note that *abnormal\_short\_term\_variability* and *Percentage of time with abnormal long-term variability* were selected as the most important two predictors by ensemble methods (decision tree, random forest, bagging) (see Figure 17, Figure 18, Figure 19).

Models	Accuracy	Kappa	Computation cost (sec)
Multinomial	0.8710544	0.8066109	0.832150
PCA Multinom	0.9106783	0.8660118	11.705304
KNN	0.9308261	0.8962383	3.497356
Tree	0.9207522	0.8811304	1.218004
Random Forest	0.9858966	0.9788450	34.142250
Bagging	0.9845534	0.9768303	36.037747
Neural Net	0.9818670	0.9728001	209.055836

Table 11: Overall model results

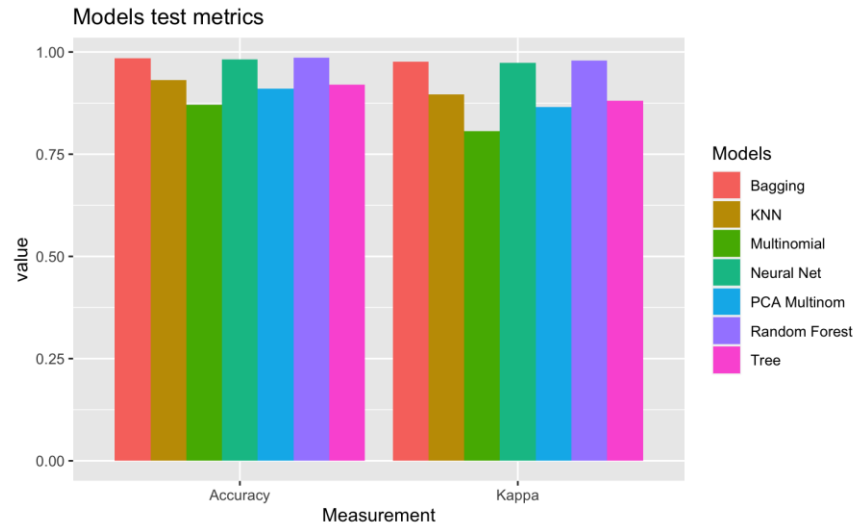


Figure 4: Models Test Metrics

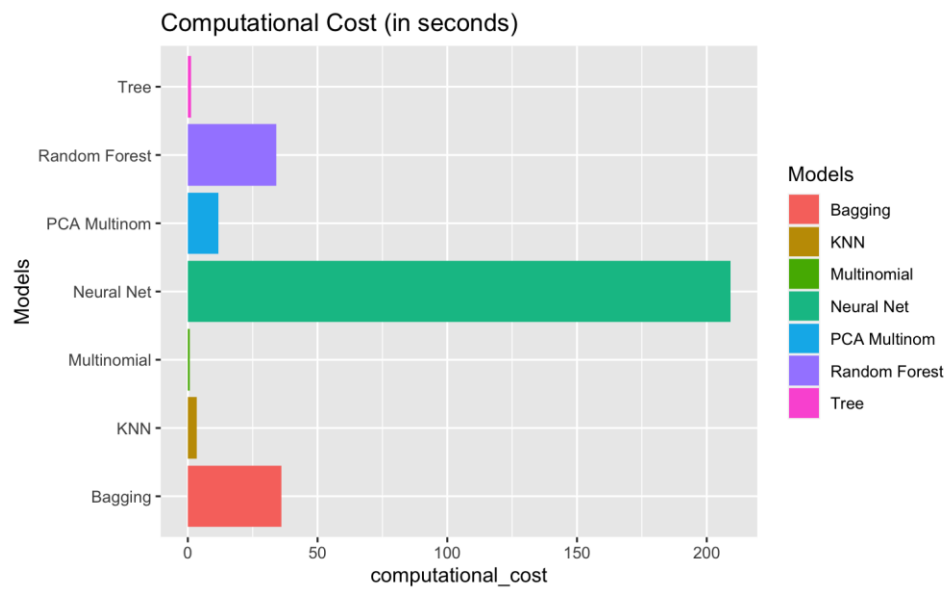


Figure 5: Computational Cost

From comparison table of all model test metrics, Random Forest, Bagging and Neural net are top performers and difference in test metrics is insignificant. If computational cost involved in training model taken into consideration, Random Forest is the most optimal best performer. Multinomial model had both lowest test metrics and lowest computational cost. Neural networks had the highest computational cost. As mentioned above, the neural networks model was choosing

the highest value tuning parameter, and when the tuning grid was expanded, the computational cost increased significantly with incremental improvement of the test metrics. Also, it can be noted that multinomial model test metrics improved significantly when dimension reduction technique PCA was applied.

### **Future Work**

In this work, neural networks model has one hidden layer. It would be interesting to investigate the performance (accuracy and computational cost) of the multi-layer neural networks for the given dataset.

Also validating variable importance conclusions with medical professionals and clinical value of the findings would help reassess developed models.

### References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

Kumar, Gokul S. (2020). Decision Trees: A step-by-step approach to building DTs, 2020. Towards Data Science. Retrieved from [Decision Trees: A step-by-step approach to building DTs | by Gokul S Kumar | Towards Data Science](#)

Larxel (2020). Fetal Health Classification Dataset. Retrieved from <https://www.kaggle.com/andrewmvd/fetal-health-classification>

Ogasawara, J., Ikenoue, S., Yamamoto, H., Sato, M., Kasuga, Y., Mitsukura, Y., ... & Ochiai, D. (2021). Deep neural network-based classification of cardiotocograms outperformed conventional algorithms. Scientific reports, 11(1), 1-9

Potter, L. (2021). How to read a CTG? Retrieved from <https://geekymedics.com/how-to-read-a-ctg/>.

Sarkar, S. (n.d). Data Mining. *Course Materials*. Retrieved from [https://bgsu.instructure.com/courses/1388522/files/100009833/download?download\\_frd=1](https://bgsu.instructure.com/courses/1388522/files/100009833/download?download_frd=1)

Shmueli, Galit, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, and Kenneth C. Lichtendahl Jr.. Data Mining for Business Analytics: Concepts, Techniques, and Applications in R. John Wiley & Sons, 2018.

UNICEF.(2022, Apr. 17). UNICEF Data: Monitoring the Situation of Children and Womemn: Retrieved from <https://data.unicef.org/sdgs/goal-3-good-health-wellbeing/>

Tables

Original variables names	New names	Description
baseline.value	base.val	FHR baseline (beats per minute)
accelerations	accel	Number of accelerations per second
fetal_movement	fetal_move	Number of fetal movements per second
uterine_contractions	uter.cont	Number of uterine contractions per second
light_decelerations	light.decel	Number of light decelerations per second
severe_decelerations	sev.decel	Number of severe decelerations per second
prolongued_decelerations	prolong.decel	Number of prolonged decelerations per second
abnormal_short_term_variability	abn.short.var	Percentage of time with abnormal short term variability
mean_value_of_short_term_variability	mean.short.var	Mean value of short term variability
percentage_of_time_with_abnormal_long_term_variability	perct.long.var	Percentage of time with abnormal long term variability
mean_value_of_long_term_variability	mean.long.var	Mean value of long term variability
histogram_width	hist.width	Width of FHR histogram
histogram_min	hist.min	Minimum (low frequency) of FHR histogram
histogram_max	hist.max	Maximum (high frequency) of FHR histogram



histogram_number_of_peaks	hist.npeaks	Number of histogram peaks
histogram_number_of_zeroes	hist.n0	Number of histogram zeros
histogram_mode	hist.mode	Histogram mode
histogram_mean	hist.mean	Histogram mean
histogram_median	hist.median	Histogram median
histogram_variance	hist.var	Histogram variance
histogram_tendency	hist.tend	Histogram tendency
Class	fetal_health	Tagged as 1 (Normal), 2 (Suspect) and 3 (Pathological)

*Table 1: Renamed Variables.*

Figures

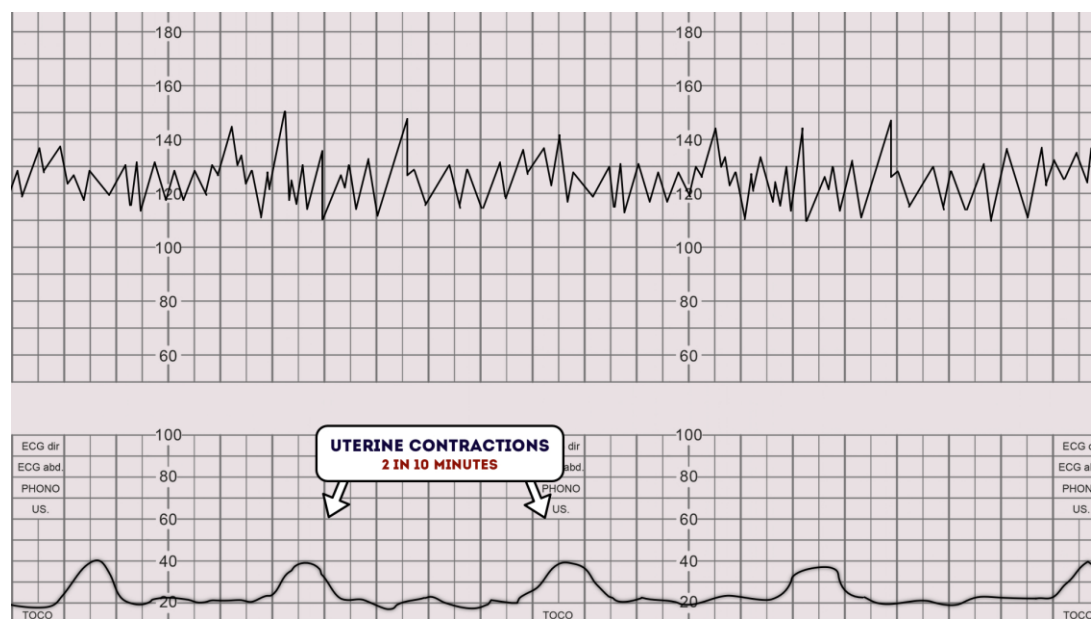


Figure 1. Uterine contraction [Image](Potter.L, 2021)

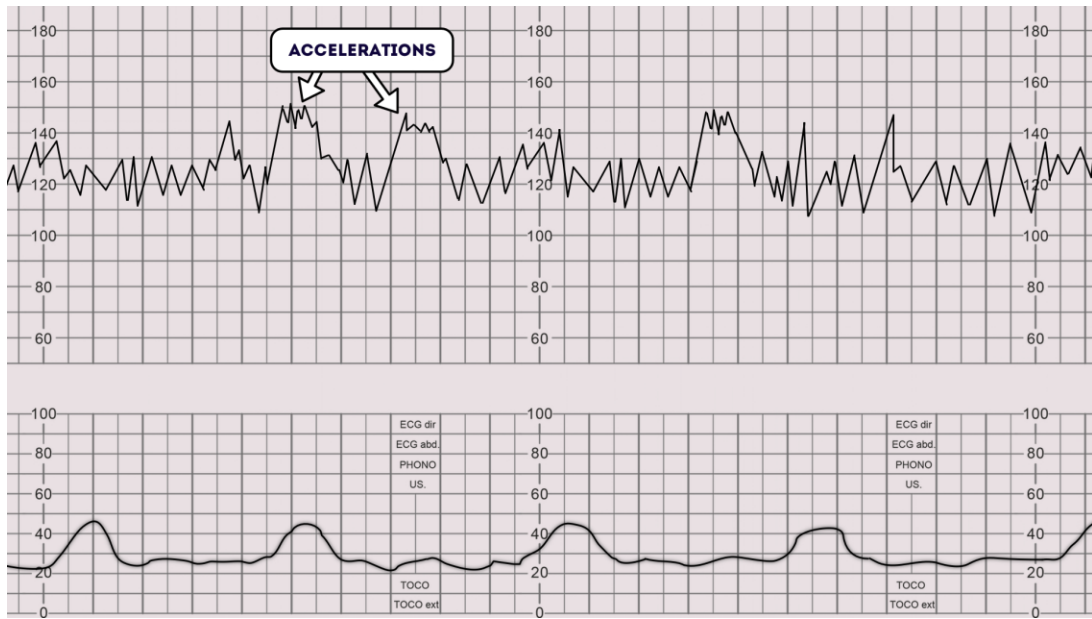


Figure 6. Accelerations. [Image](Potter.L, 2021)

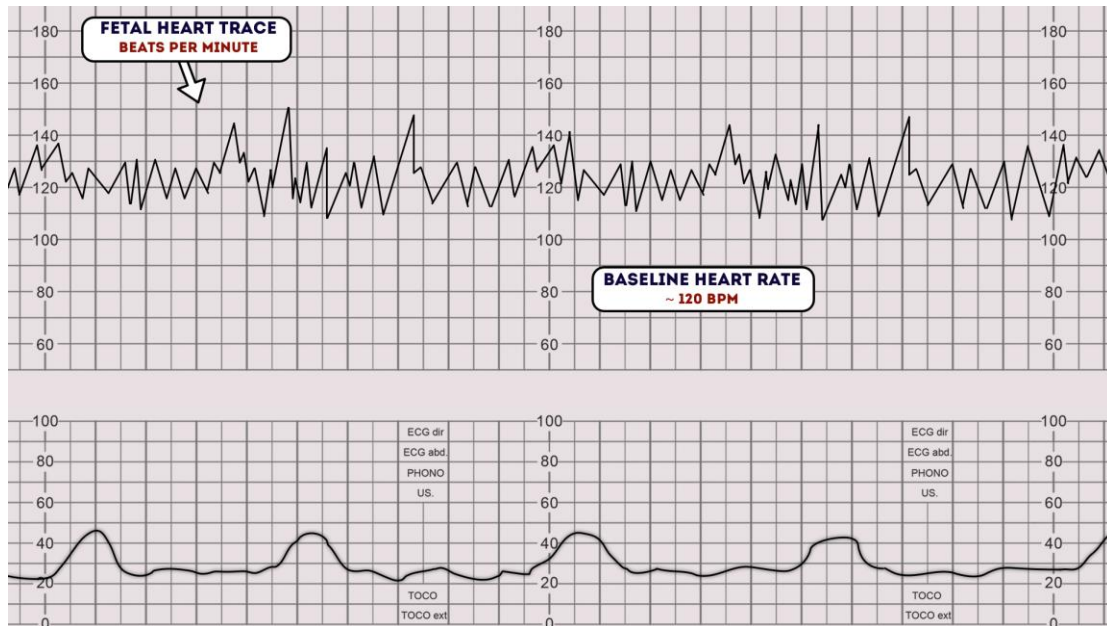


Figure 7. Baseline heart rate. [Image](Potter.L, 2021)

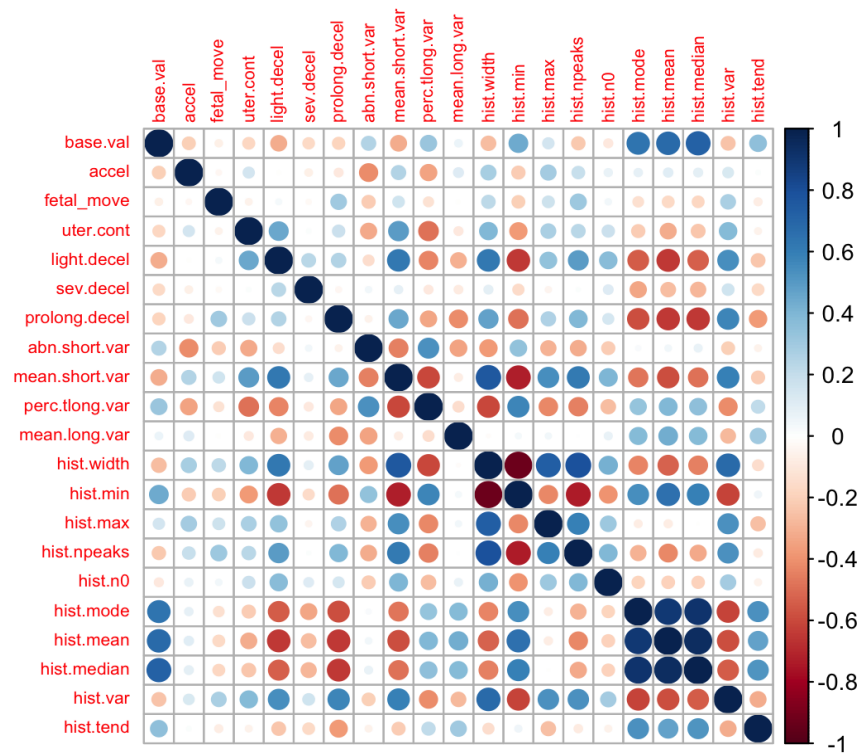


Figure 8. Correlation plots

Tuning Parameters

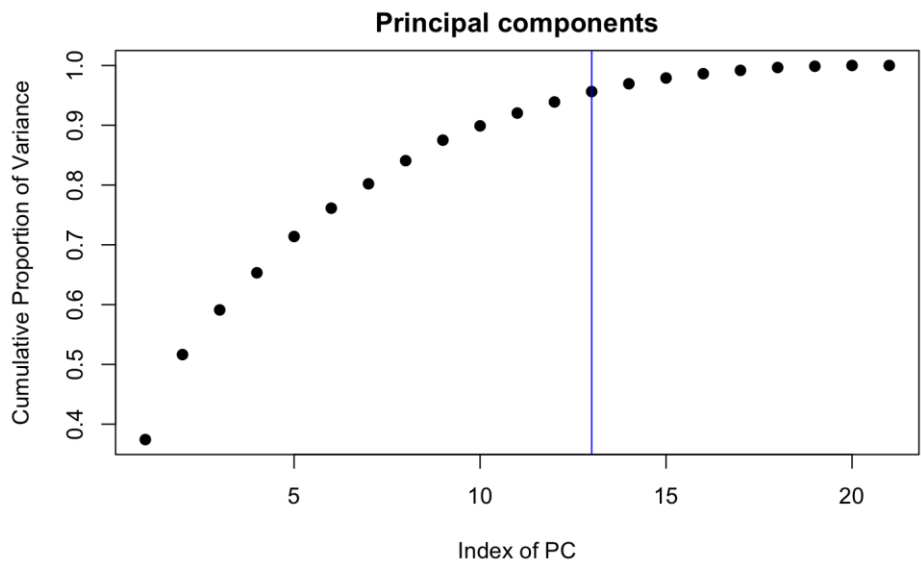


Figure 9: Principal Components

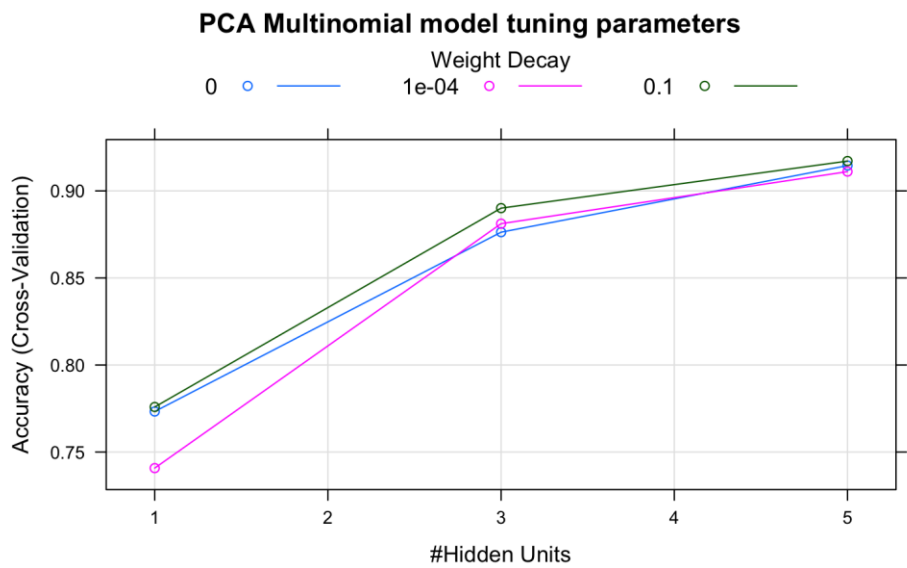


Figure 10: PCA Model Tuning Parameters

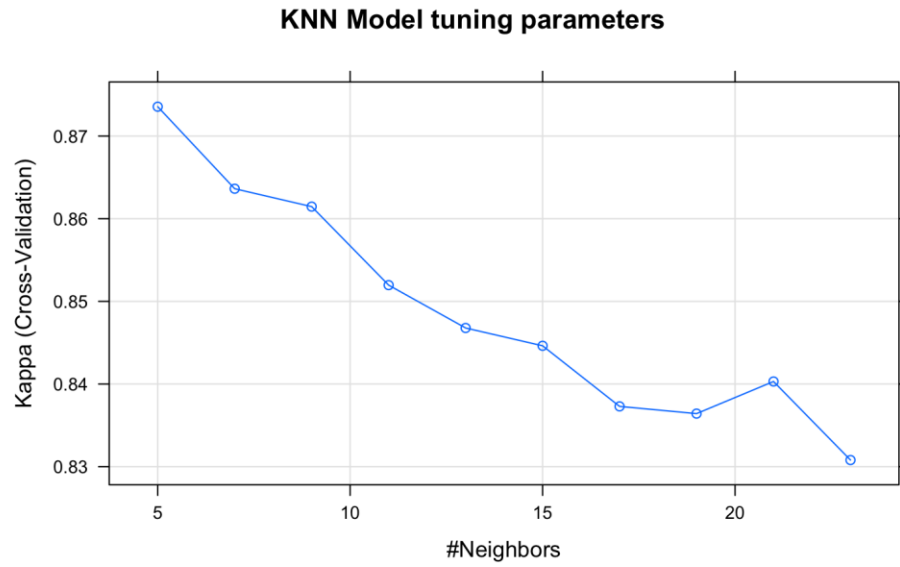


Figure 11: KNN Model Tuning Parameters

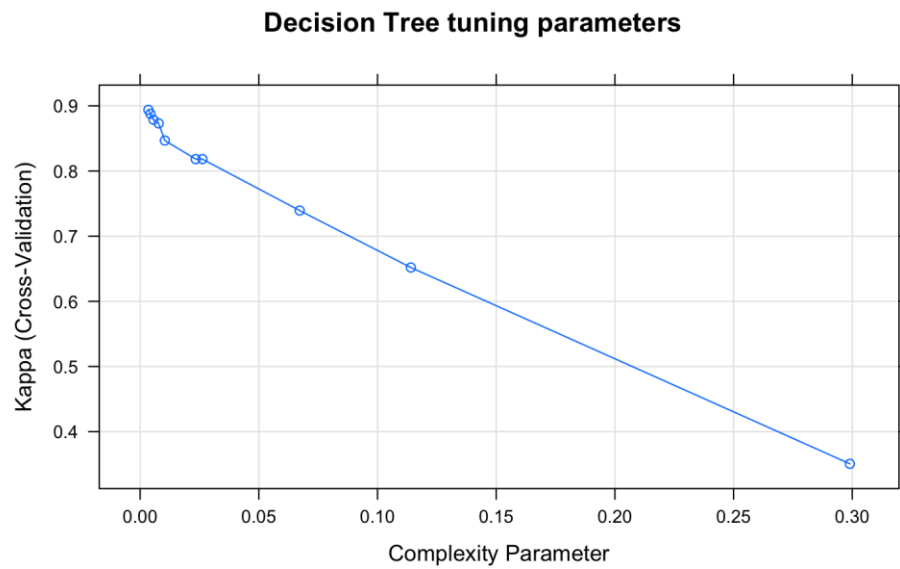


Figure 12: Decision Tree Model Tuning Parameters

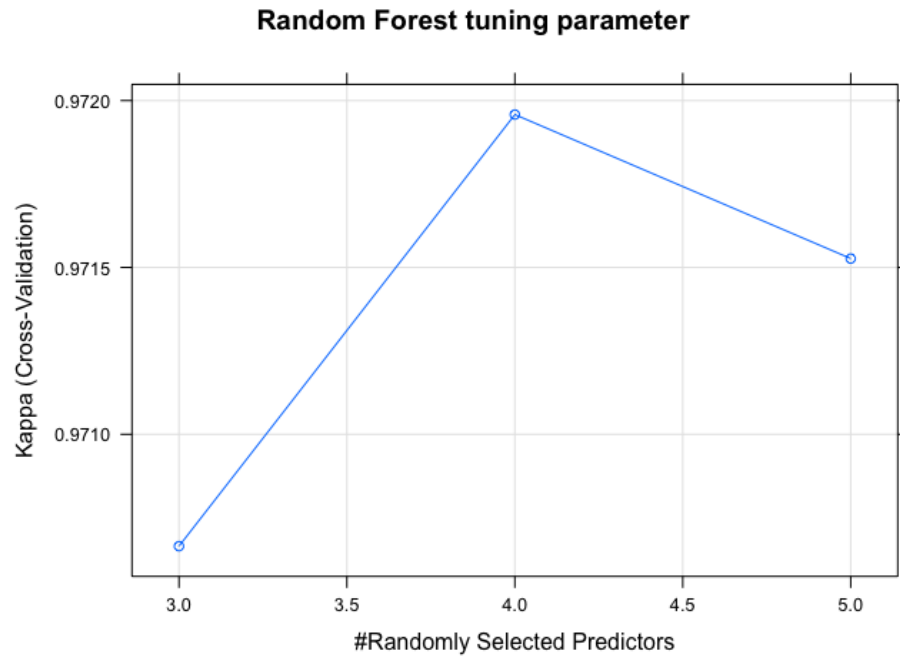


Figure 13: Random Forest Model Tuning Parameters

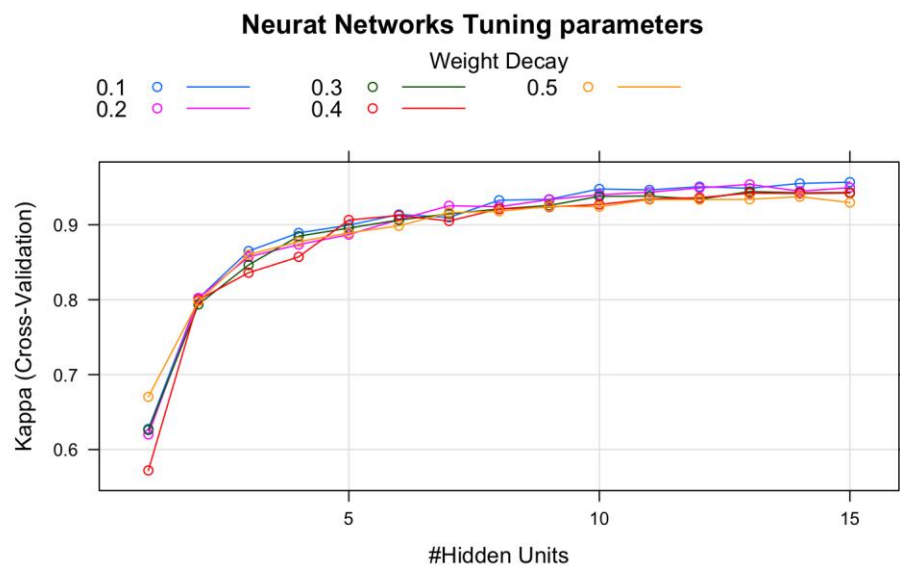


Figure 14: Neural Network Model Tuning Parameters

### Variable Importance

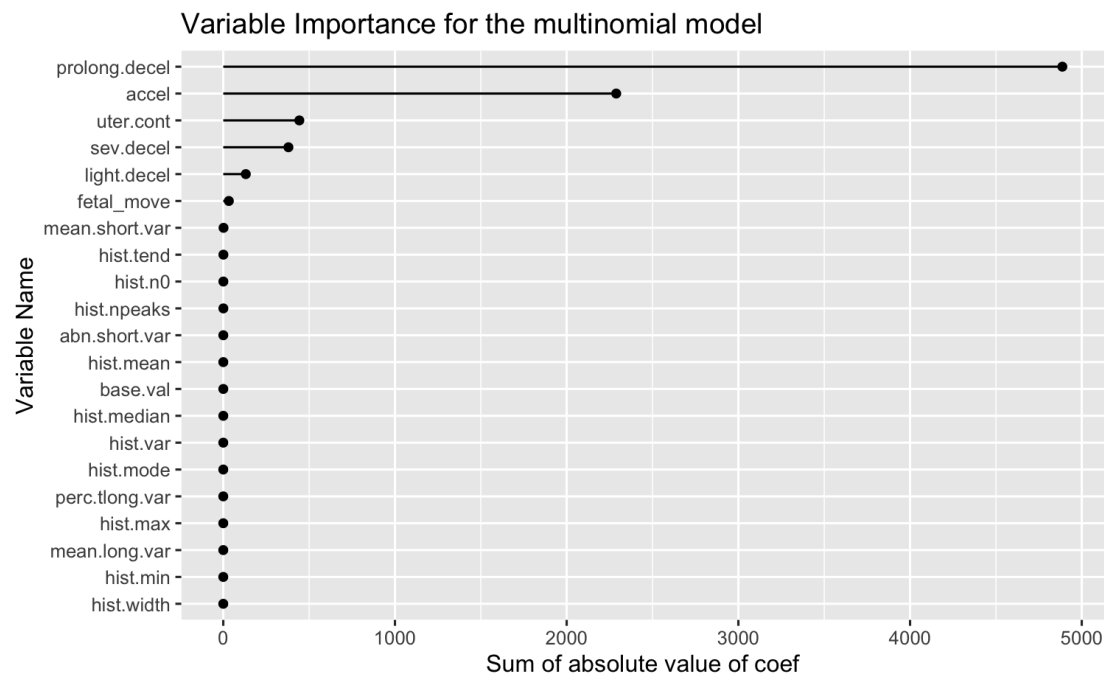


Figure 15: Variable Importance for The Multinomial Model

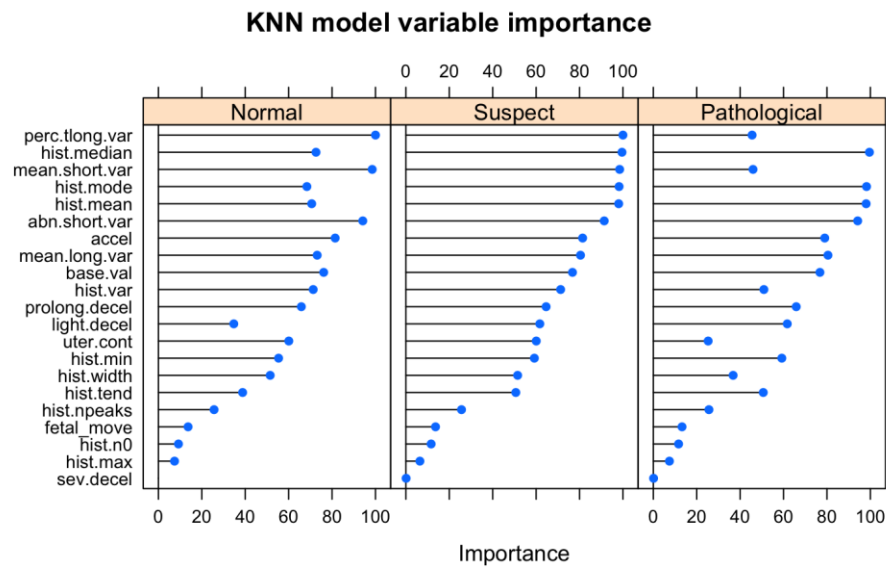


Figure 16: Variable Importance for KNN Model

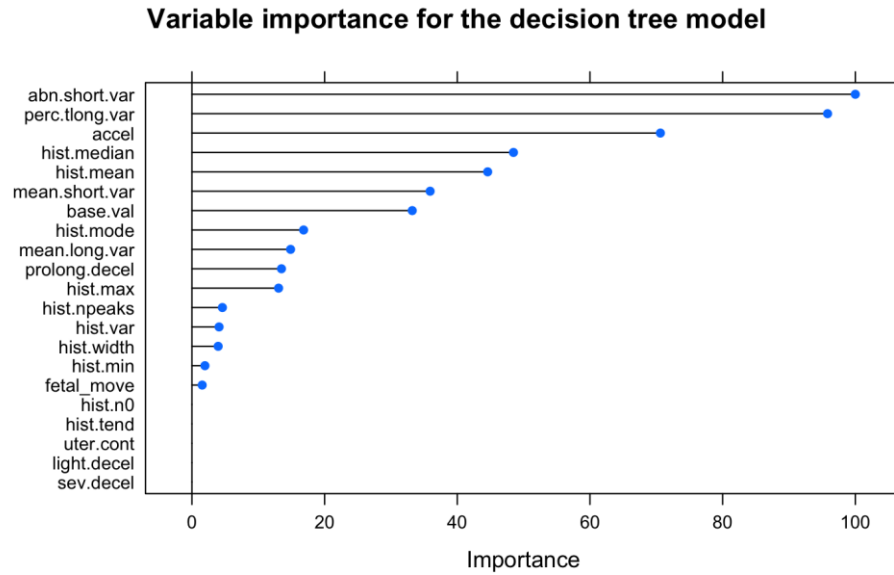


Figure 17: Variable Importance for the Decision Tree Model

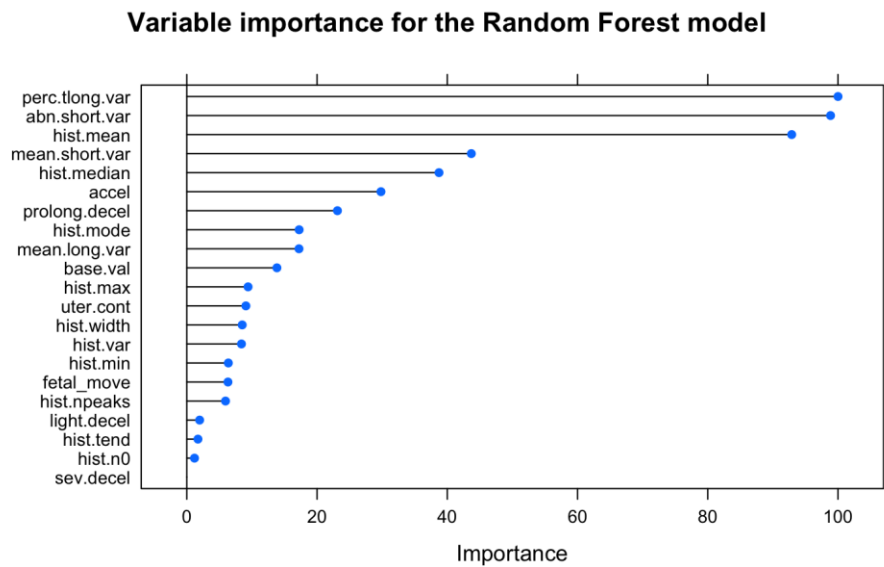


Figure 18: Variable Importance for the Random Forest Model



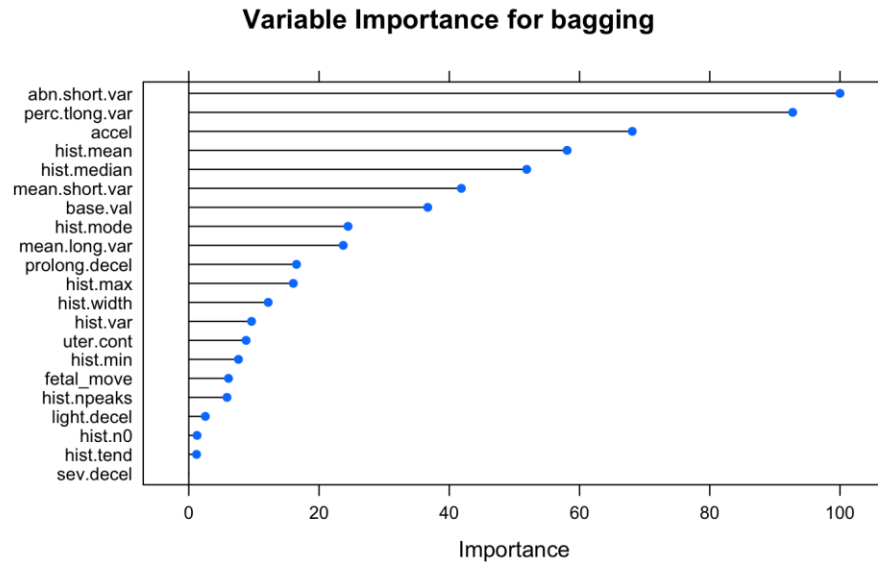


Figure 19: Variable Importance for the Bagging Model

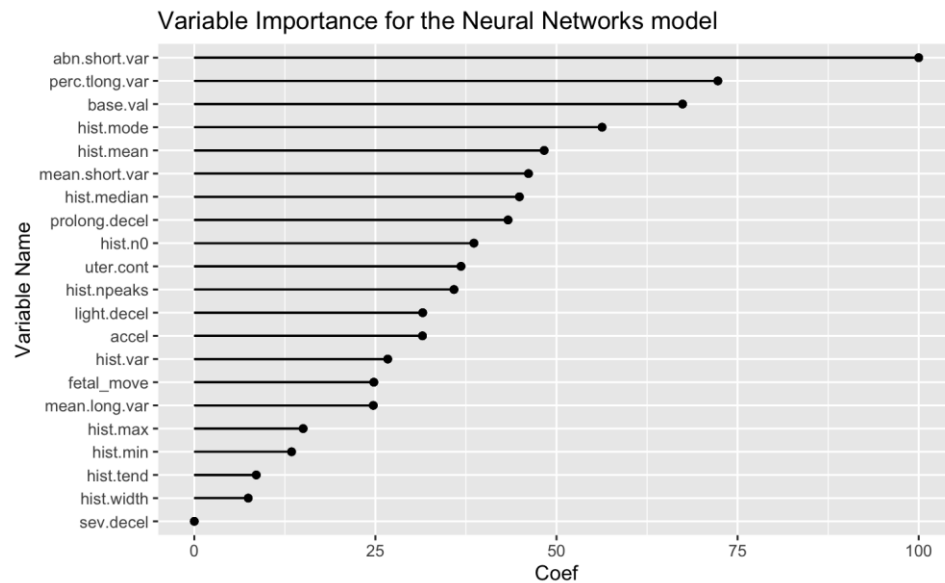


Figure 20: Variable Importance for the Neural Networks Model