# Machine Learning exam report

Matteo Ferro 826815
Riccardo Mirmina 826173

28 Aprile 2022

## 1 Introduction

The dataset used for the analysis is **"Fetal Health"** from Kaggle (link here). The dataset contains information about 2126 records of features extracted from Cardiotocogram exams (CTGs), which were then classified by three expert obstetricians into 3 classes: Normal, Indeterminate, Abnormal.

## 2 Medical overview

Obstetricians classify fetal status in three categories based on a technique called **DR C BRAVADO**, that stands for:

DR = Determine Risk

C = Contractions

BRAVADO = **B**aseline **RA**te, **V**ariability, **A**ccelerations, **D**ecelerations, **O**verall assesment

**DR** is referred to maternal and fetal risk factors, in our case we got only one information, the one about number of fetal movements. **C** is included as number of contraction in one second. The remaining 20 dataset features are referred to the fetal heart rate tracing itself and to **BRAVADO** analysis.
In figure 1 we can see a general image of what a tracing look like, where above is shown the Fetal Heart Rate (FHR) and below is shown uterine contraction.

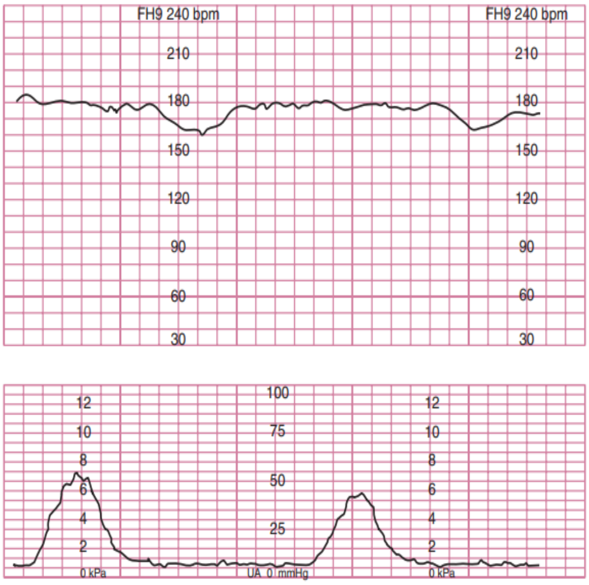


Figure 1: An exemple of FHR tracing

From the FHR tracing the clinician extract information about:

1. baseline rate:
   In quick words, is defined as the mean FHR, rounded to increments of 5, excluding periodic changes or big variations (see the appendix for more information).
   EX: In fig.1 it can be calculated as 180.

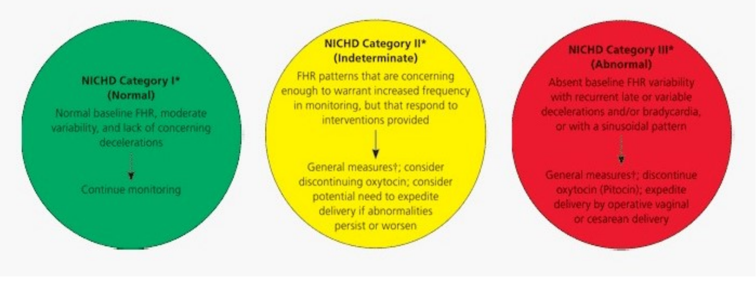
   The normal range for baseline FHR is defined by NICHD as 110 to 160 beats per minute. A baseline of less than 110 bpm is defined as bradycardia, where 100 to 110 bpm is Mild bradycardia. Rates of less than 100 bpm may be seen in fetuses with congenital heart disease or myocardial conduction defects. A baseline greater than 160 bpm is defined as tachycardia.

2. variability:
   The FHR normally exhibits variability from the baseline value and it is linked to the fetal central nervous system. Therefore, it is a vital clue in determining the overall fetal condition.
   The NICHD has categorized variability into the following classifications, depending on the amplitude of the FHR tracing: absent (range: undetectable), Minimal (range: detectable, but $\leq 5$ bpm), Moderate (range: 6 to 25 bpm) and Marked (range: $> 25$ bpm).
   EX: in fig.1 there is minimal variability

3. accelerations:
   Visually apparent, abrupt (onset to peak $< 30$ seconds) increase in FHR from the baseline. Abrupt increases in the FHR are associated with fetal movement or stimulation and are indicative of fetal well-being.
   EX: in fig.1 there are no accelerations

4. decelerations:
   Periodic changes in FHR, as they relate to uterine contractions.
   The decrease in FHR is calculated from the onset of the uterine contraction to the nadir of the deceleration.
   Early decelerations are transient, gradual decreases in FHR that are visually apparent and usually symmetric. They occur with and mirror the uterine contraction, in other words, the nadir of the deceleration occurs at the same time as the peak of the contraction. Early decelerations are nearly always benign and probably indicate head compression, which is a normal part of labor.
   Late decelerations are visually apparent, usually symmetric, and have the characteristic feature of onset of the deceleration after the onset of the uterine contraction. The timing of the deceleration is delayed, with the nadir of the deceleration occurring after the peak of the contraction.
   EX: in fig.1 we see two uterine contraction (below) and two late decelerations (above).
   Other patterns of decelerations are: variable decelerations and prolunged decelerations, see appendix for more information.

5. Overall assesment:
   Using the "Stoplight algorithm" for intrapartum surveillance of fetal heat rate (FHR) the clinician divide fetuses in three main categories: normal, interminate, abnnormal.
   This is the response variable for which we want to classify.



NICHD Category I* (Normal)
Normal baseline FHR, moderate variability, and lack of concerning decelerations
Continue monitoring

NICHD Category II* (Indeterminate)
FHR patterns that are concerning enough to warrant increased frequency in monitoring, but that respond to interventions provided
General measures†; consider discontinuing oxytocin; consider potential need to expedite delivery if abnormalities persist or worsen

NICHD Category III* (Abnormal)
Absent baseline FHR variability with recurrent late or variable decelerations and/or bradycardia, or with a sinusoidal pattern
General measures†; discontinue oxytocin (Pitocin); expedite delivery by operative vaginal or cesarean delivery

Figure 2: stoplight rule

In addition to the features used for the BRAVADO analysis, in the dataset are included other information extracted from the FHR tracing such as mean FHR, median FHR, ecc...

## 3 Data exploration

The dataset used in the analysis does not contain any missing value, is composed by 2126 observations and 22 total variables: 21 numeric and one categorical, *fetal_health*, that takes on three modes: *Normal, Indeterminate, Abnormal*, with their respective frequencies 0.778, 0.139, 0.083.

We can see that the majority of observation are from the first category, for this reason we have to stratify for the response variable while creating the training and test set, otherwise we would obtain distorted performance results.

Nevertheless, we are still going to have problems during the model learning phase because of the low number of available data for category II and III.

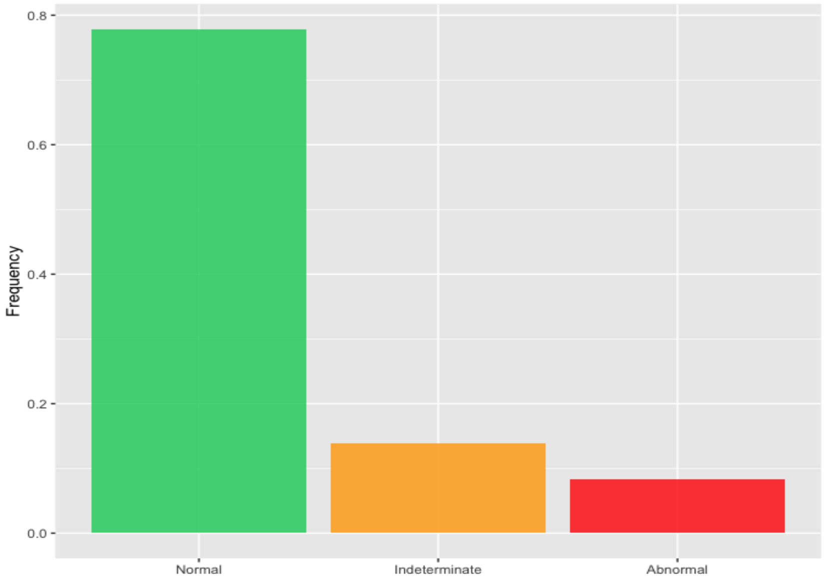


Figure 3: frequencies in the target variable *fetal_health*

The aim of the analysis is to classify each observation in one of these classes.

The following are the boxplots conditioned on the classes *Normal, Indeterminate and Abnormal*, for some of the most interesting explanatory variables from a clinical point of view.

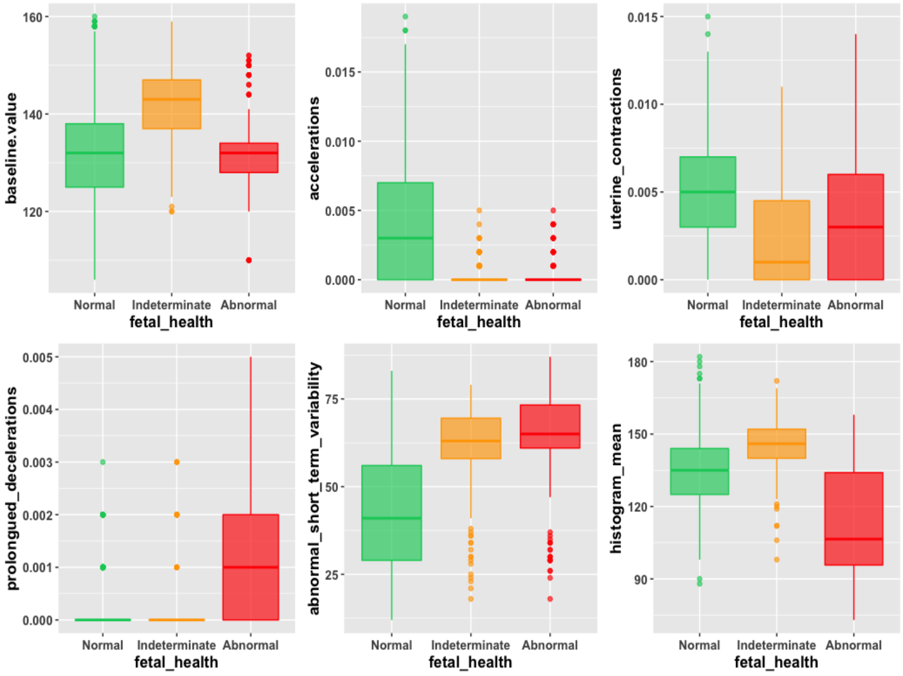


Figure 4: Conditioned boxplots to the classes *Normal, Indeterminate e Abnormal*

It can be observed that the different variables distribution within the three classes can make them particularly useful for the purposes of classification.
We see for example that in the indeterminate class the values of the *baseline value* variable are higher than in other classes. We can see that, in general, normal fetuses are characterized by an higher number of accelerations and abnormal fetuses have more prolonged decelerations. Last but not least we see that in some cases the indeterminate and abnormal classes take on similar values, like in *abnormal short term variability* and *uterie contraction*, this can create difficulties in the classification procedure.

## Split dataset into train and test sets

The data set it's been divided into a training dataset, consisting of 1594 observations, and a testing dataset consisting of 532 observations. Note that the target variable *fetal_health* is well represented in each of these sets

| | Normal | Indeterminate | Abnormal |
|---|---|---|---|
| dataset | 0.778 | 0.139 | 0.083 |
| training | 0.781 | 0.137 | 0.082 |
| testing | 0.771 | 0.143 | 0.086 |

Table 1: Relative frequencies of classes within sets considered

# 4 Tree based Classifier

Below, we will apply the random forest classifier that considers classification trees as their predictive tool. Well known the estimation of a tree depends strongly on the data set used, for this reason the random forest algorithm estimates many trees on different data sets and, before making a division, it considers only a subset of the explanatory variables, which varies at every division. For what has been said, the random forest is very useful in feature selection. It is in fact possible to know, among all the explanatory variables considered, which had been more influential in reducing the impurity in the nodes, according to the Gini index. Once this information has been acknowledged, we will try to estimate a simple classification tree based on a few of those most influential variable in order to obtain a good but simple classifier.

## 4.1 Random forest

We now want to determinate the number of explanatory variables to sample at every split ($mtry$), according to the Out-of-bag (OOB) error method. As said the random forest estimates more trees on different sets of data. These sets are obtained with the bootstrap technique, therefore some observations will be randomly selected (with replacement) while other are excluded from the sample and are going to form the so called *Out-of-bag Set.* We now evaluate the predictive capacity of the tree (estimated on the *Bootstrap Sample*) over the OOB set, thus obtaining the OOB error for the model. Repeating this on a large number of trees ($ntree$) and averaging the OOB errors obtained, we will get an estimate of the OOB error for the random forest. Note that it's stability will be increased as the number of trees considered will be higher. The graph below shows the OOB errors of the random forest obtained with $mtry = 1, 2, 4, 8, 16$ and $ntree = 1000$. It comes out that the optimal explanatory number that minimizes the OOB error is 8.

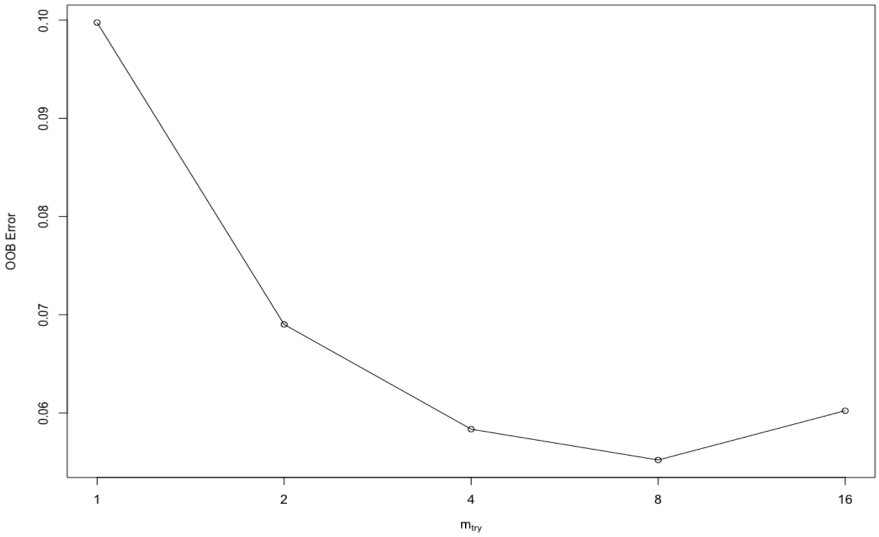


Figure 5: OOB error as the number of the explanatory variables change

At this point we estimate on the training dataset a random forest with $mtry = 8$ and $ntree = 1000$, below are the performances obtained on the training set and the test set.

| | Normal | Indeterminate | Abnormal | Class error |
|---|---|---|---|---|
| Normal | 1217 | 21 | 7 | 0.022 |
| Indeterminate | 48 | 168 | 3 | 0.233 |
| Abnormal | 7 | 7 | 116 | 0.108 |

Table 2: Confusion matrix (training set) obtained by classification with the out of bag method

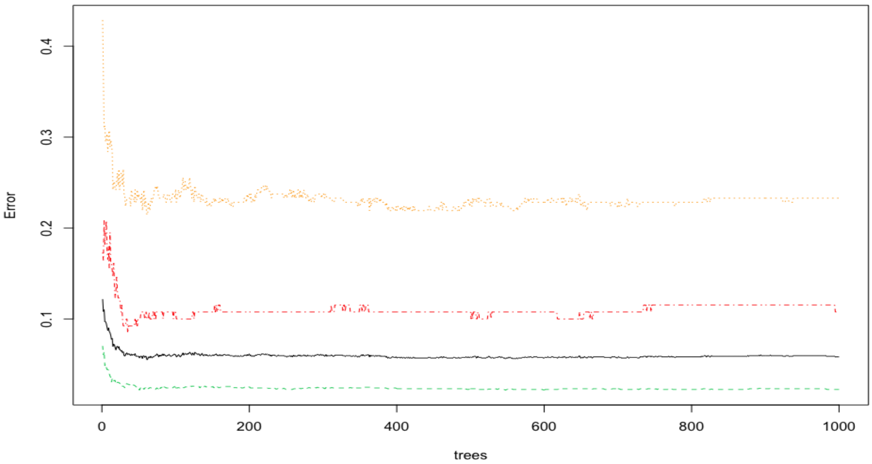


Figure 6: Classification errors when changing the number of trees estimated on the training set

Figure 6 shows the OOB error estimates for the response classes *Normal*, *Indeterminate*, *Abnormal* (respectively in green, orange and red) and the overall OOB error (in black) obtained varying the number of trees in the random forest on the training set. Accuracy in the training set is equal to $1 - \text{OOB error} = 1 - 0.0552 = 0.944$.

| | Normal | Indeterminate | Abnormal | Class error |
|---|---|---|---|---|
| Normal | 405 | 4 | 1 | 0.012 |
| Indeterminate | 23 | 53 | 0 | 0.303 |
| Abnormal | 2 | 2 | 42 | 0.087 |

Table 3: Confunsion matrix obtained on the test set

In Table 3 the class *Indeterminate* is characterized by a major classification error and the accuracy is 0.939.

We now want to investigate which, between the explanatory variables, have been most influential in reducing the impurity of the node according to the Gini index. The following graph sorts the explanatory variables by *Mean Decrease Gini.*

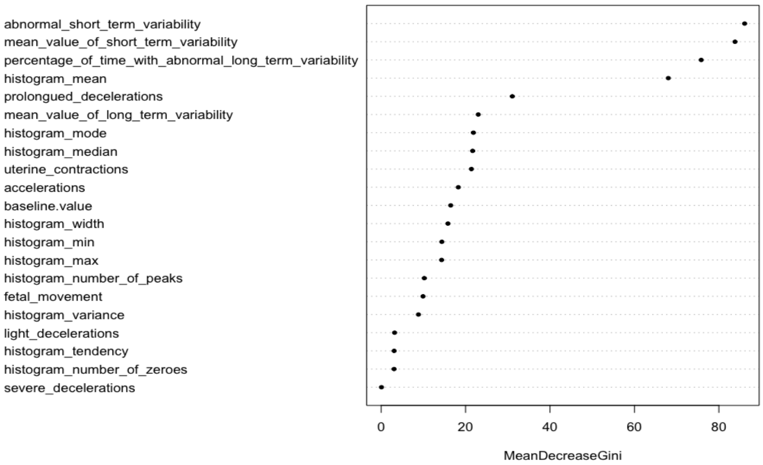


Figure 7: Variable importance as measured by a random forest

## 4.2 Classification trees

The random forest has allowed us to obtain excellent results at the expense of interpretability. In this section we want to get a good classifier but at the same time we want it as simple as possible, in order to do so we will take advantage of some of the results obtained in the section 4.1.

To obtain a classification tree with only few nodes, it is necessary to select a subset of the 21 explanatory variables available. We decided to select the first 4 explanatory variables that contributed the most to the reduction of the Gini index (measurement of impurity in nodes). The variables chosen are therefore:

- abnormal_short_term_variability
- mean_value_of_short_term_variability
- percentage_of_time_with_abnormal_long_term_variability
- histogram_mean

The growth of the tree is driven by the principle of purity of the nodes that are created from time to time. A node is considered pure when it contains only observations that belong to a single class. In this case the impurity function used is the Gini index:

$$\sum_{l \in \Omega} \hat{p}_{kl}(1 - \hat{p}_{kl}),$$

where

$$\hat{p}_{kl} = \frac{1}{card(R_k)} \sum_{t: x_t \in R_k} I(y_t = l)$$

where $R_k$ indicates a generic region and $\Omega$ the set of classes of the response variable. The tree has grown to 162 terminal nodes. The pruning phase, which follows the growth phase of the tree, is based on a function of cost complexity that relates the adaptation of the tree to the data with its size, the adaptation is penalized for the size tree. The function that has been used is the following one:

$$\sum_{k=1}^{|T^\alpha|} (1 - \max_{l \in \Omega} \hat{p}_{kl}) p_k + \alpha |T^\alpha|, \tag{1}$$

where $|T^{\alpha}|$ is the size of the tree and $(1 - \max_{l\in\Omega}\hat{p}_{kl})$ the classification error in the region $R_k$. K-fold cross validation determines the number $|T^{\alpha}|$ that minimizes the function (1), in this case the optimal size of the tree is 4.

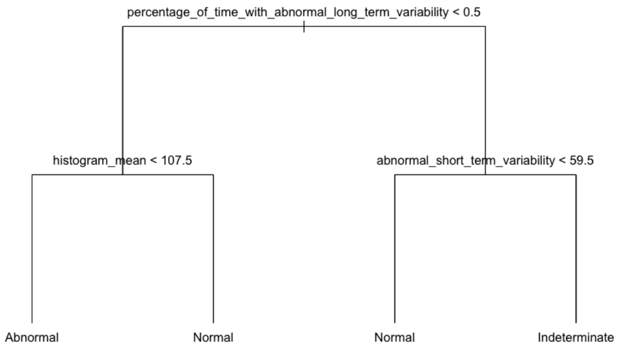

percentage_of_time_with_abnormal_long_term_variability < 0.5
histogram_mean < 107.5
abnormal_short_term_variability < 59.5
Abnormal
Normal
Normal
Indeterminate

Figure 8: Classification tree with 4 terminal nodes

The pruning of the tree, obtained with the method described above, has allowed us to obtain a tree with only 4 terminal nodes which excludes the variable *mean_value_of_short_term_variability*. Below is the performance of this classifier on the test set where it is possible to observe how the *Abnormal* class is characterized by a higher classification error and the obtained accuracy is equal to 0.872.

| | Normal | Indeterminate | Abnormal | Class error |
|---|---|---|---|---|
| Normal | 385 | 22 | 3 | 0.061 |
| Indeterminate | 23 | 53 | 0 | 0.303 |
| Abnormal | 5 | 15 | 26 | 0.434 |

Table 4: Confunsion matrix (test set) of classification tree with 4 terminal nodes

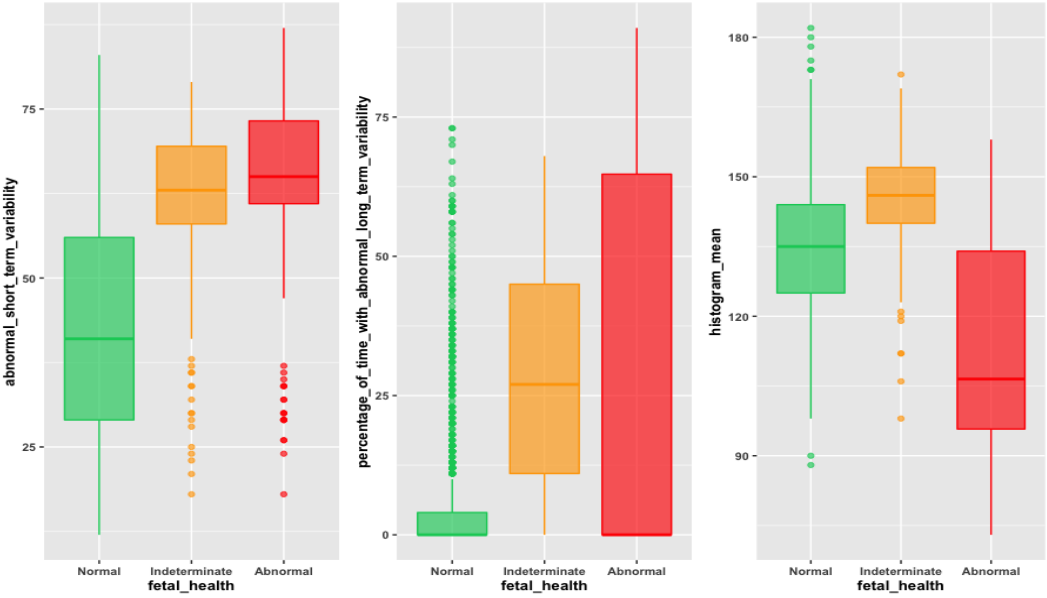


Figure 9: Conditioned boxplots to the classes *Normal, Indeterminate e Abnormal*

The boxplots (of the three variables used in the tree) conditioned to the response variable allow to understand the reason for such a high classification error for the classes *Abnormal* and *Indeterminate*. The first division, in the classification tree, is determined by the values of the variable *percentage_of_time_with_abnormal_long_term_variability*, it is observed that values close to zero are assumed generally by units in the classes *Normal* and *Abnormal*. The variable *histogram_mean* allows a better classification for observations in the class *Abnormal*, which assume lower values in that class, while the variable *abnormal_short_term_variability* allows a better classification for observations in the class *Normal*. The distributions of the variable *abnormal_short_term_variability* conditioned to the classes *Indeterminate* and *Abnormal* are very similar and generate uncertainty to the classifier. The above justifies the structure and the classification error of the tree estimated.

# 5 Neural Network Classifier

## Introduction

In order to solve the multiclass classification problem we are facing with we can try to use a powerful supervised learning technique called ”Artificial Neural Network” (ANN). An ANN is composed by various layers of neurons all connected together, creating what is commonly called a ”dense” network .
The following formula represent how the neuron in the (j+1)th layer work:

$$Y = a(\sum_{i=1}^{n_j} w_{ij} \cdot X_{ij} + b) \tag{2}$$

We first calculate a weighted sum of all the inputs $X_{ij}$ to the neuron, we then add a bias term $b$, and than pass the result through an activation function $a(.)$ which permits to add non linearity. We have finally obtained an output Y which is going to become an input for the neurons in the next layer.
In order to fit an ANN model we first have to choose an activation function and then we try to update all the parameters (weights and biases) based on the classification error obtained during training. The procedure used for estimating the set of weights that minimizes the error is called Gradient Descent. In our case we use a modified version of GD called SGD (Stochastic Gradient Descent) that replaces the actual gradient (calculated on entire data set) with an estimate (calculated from a randomly selected subset of the data).
Note that this method can be optimized using various algorithms as we are going to see.

## model

We start by creating a basic artificial neural network composed by 3 layers: one input layer with 21 neurons (one for each covariate we have access to), one hidden layer with 10 neurons and one output layer with 3 neurons (one for each possible value of the response).

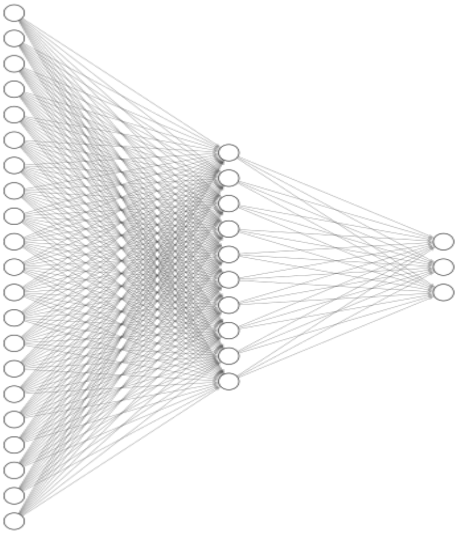


Figure 10: neural network architecture

To complete the definition of the model we need to specify the activation function and the algorithm used to perform gradient descent.
We are using the RELU (Rectified Linear Unit) activation function between neurons and the SOFTMAX function for the final output. The weights are estimated using the SGD algorithm

(Stochastic Gradient Descent).
The model is almost complete, the last component we need to specify is the metrics to be used in evaluating the model error/performance. We decide that the loss function we want to minimize during model training is the categorical cross-entropy and the metric we are going to use for comparing models performances is going to be the accuracy of the prediction.

### results

Due to the stochastic nature of the neural network estimation we try to fit the model 100 times and store all the information about the performance obtained.
In figure 11 is presented a plot of all the model accuracy (on test set) results.

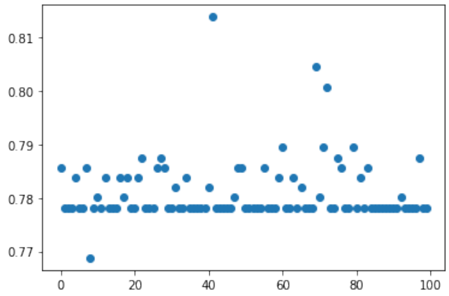
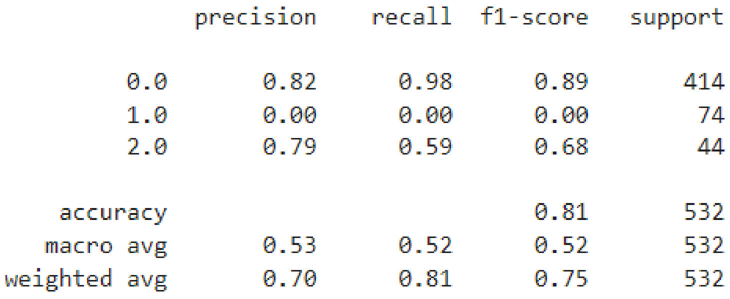


Figure 11: SGD model accuracy

### observations

We note that the majority of time we got the same model performance, for instance 0.7790 accuracy. The model having this performance is very interesting, in fact after a quick analysis we can see that it's classifying all observation as "category I".
Due to the high frequency of the first category of the response, the model achieves a good result even with a clear under-fitting strategy.
We now analyse the best model obtained: the model have an accuracy on test set equal to 81% and in figure 12 we can look at his classification performance.



```
              precision    recall  f1-score   support

         0.0       0.82      0.98      0.89       414
         1.0       0.00      0.00      0.00        74
         2.0       0.79      0.59      0.68        44

    accuracy                           0.81       532
   macro avg       0.53      0.52      0.52       532
weighted avg       0.70      0.81      0.75       532
```

Figure 12: best model classification report

As we can see, even this model cheats a little, it does not classify any observation to the "category II"! For this reason we now try to fit a model using alternative algorithms instead of the basic stochastic gradient descent.

## gradient descent optimization

In the stochastic gradient descent algorithm we have to define an hyper-parameter $\eta$ called "learning rate" (default in keras $\eta = 0.01$) that reflects how much we allow the parameters to follow the direction shown by the gradient.

$$w_t = w_{t-1} - \eta \cdot \nabla_{w_t} \tag{3}$$

Unfortunately, this hyper-parameter is very difficult to choose, because if we set it too small then the parameter update will be very slow. Otherwise, if we set it too large then the parameter will move all over the function too quickly. To make things worse, the high-dimensional non-convex nature of neural networks optimization could make a fixed learning rate even worse to use.
In order to solve this problem researchers had come up with the so called "adaptive algorithms" which adaptively scale the learning rate.

We are going to take a look and apply three different adaptive algorithms:
AdaGrad, RMSprop and Adam.

As we have done for the stochastic gradient descent algorithm we fit 100 model for each of the optimizers we have chosen. In table 5 is presented a summary of the accuracy on test set obtained for the best models.

| optimizer | best model accuracy |
|---|---|
| SGD | 0.814 |
| Adagrad | 0.808 |
| RMSprop | 0.823 |
| Adam | 0.818 |

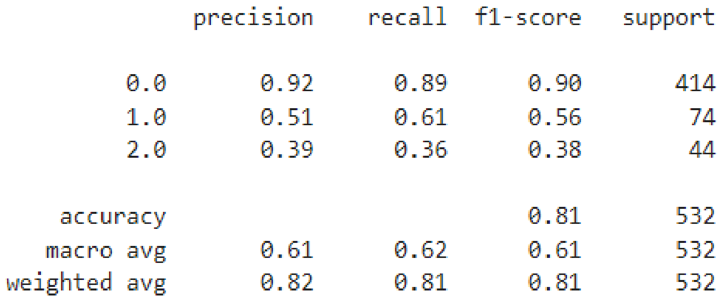
Table 5

### 1. Adaptive Gradient

$$\begin{aligned} w_t &= w_{t-1} - \frac{\eta}{\sqrt{v_t} + \epsilon} \cdot \nabla_{w_t} \\ v_t &= v_{t-1} + \nabla_{w_t}^2 \end{aligned} \tag{4}$$

where $w$ is the weight to be updated, $v$ is the correction term ,$\eta$ is the initial learning rate and $\epsilon$ is a small value to avoid division by zero

AdaGrad maintains a per-parameter learning rate based on the sum of squared gradients. This technique has the advantage of speeding along while learning for sparse datasets, but has the disadvantage that the squared gradients keep accumulating. Every additional term is positive, so the accumulated sum in the denominator continues to grow during training. This shrinks the learning rate until it's infinitesimally small, which eventually makes the algorithm unable to acquire additional knowledge.
AdaGrad considerably improves performance for problems with sparse gradients, such as natural language or computer vision problems.



```
              precision    recall  f1-score   support

         0.0       0.92      0.89      0.90       414
         1.0       0.51      0.61      0.56        74
         2.0       0.39      0.36      0.38        44

    accuracy                           0.81       532
   macro avg       0.61      0.62      0.61       532
weighted avg       0.82      0.81      0.81       532
```

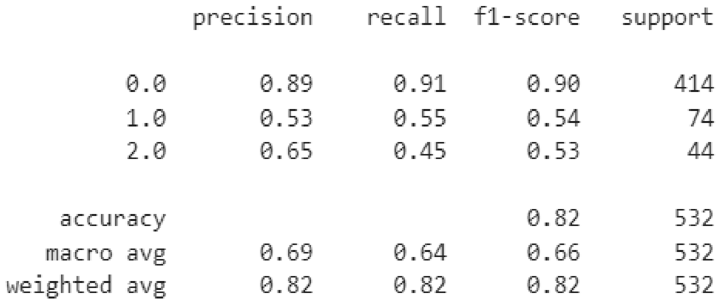
Figure 13: model obtained using AdaGrad optimizer

From figure 13 we see that the best model obtained using AdaGrad finally classify observation to all classes of the response!
This model have a higher performance in the classification of normal fetuses, compared to the one from the SGD model, but cannot discriminate well between the other two categories.

### 2. Root Mean Squared Propagation

$$\begin{aligned} w_t &= w_{t-1} - \frac{\eta}{\sqrt{v_t} + \epsilon} \cdot \nabla_{w_t} \\ v_t &= \beta v_{t-1} + (1-\beta)\nabla^2_{w_t} \end{aligned} \tag{5}$$

RMSprop, proposed by Geoffrey Hinton (the father of gradient descent), uses an average of squared gradients such as AdaGrad, except it try to solve the rapid decay of the learning rate using the parameter $\beta$. In fact, it balances the step size (momentum) creating an exponentially decaying average of past squared gradients.



```
              precision    recall  f1-score   support

         0.0       0.89      0.91      0.90       414
         1.0       0.53      0.55      0.54        74
         2.0       0.65      0.45      0.53        44

    accuracy                           0.82       532
   macro avg       0.69      0.64      0.66       532
weighted avg       0.82      0.82      0.82       532
```

Figure 14: model obtained using RMSprop optimizer

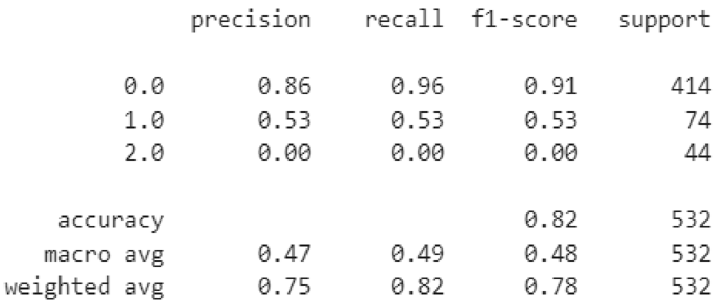
From figure 14 we see that the best model obtained using RMSprop again classify observation to all classes of the response.
We see that its performance is similar to AdaGrad, but this model can better discriminate observation in the abnormal category

**3. Adaptive Moment**

$$\begin{aligned} w_t &= w_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \\ v_t &= \beta_1 v_{t-1} + (1 - \beta_1)\nabla^2_{w_t} \\ m_t &= \beta_2 v_{t-1} + (1 - \beta_2)\nabla_{w_t} \end{aligned} \tag{6}$$

where $\hat{m}_t = \frac{m_t}{1-\beta'_2}$ and $\hat{v}_t = \frac{v_t}{1-\beta'_1}$

Adam extends on stochastic gradient descent to solve non-convex problems faster, while using fewer resources than many other optimization programs. It's most effective in extremely large datasets.
Adam combines the advantages of two other stochastic gradient techniques, Adaptive Gradients and Root Mean Square Propagation.
Instead of adapting the parameter learning rates based only on the average first moment ($m_t$), Adam also makes use of the average of the second moments of the gradients ($v_t$).



```
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91       414
         1.0       0.53      0.53      0.53        74
         2.0       0.00      0.00      0.00        44

    accuracy                           0.82       532
   macro avg       0.47      0.49      0.48       532
weighted avg       0.75      0.82      0.78       532
```

Figure 15: model obtained using AdaM optimizer

From figure 15 we see that the best model obtained using AdaM classify the observation only into the first to classes.

## summary

Given the four best models, we now have to choose one of them. We first start by looking at the overall performance: all the model are similar with a test accuracy around 80%, but the best is the one estimated using RMSprop, with a value of 0.823.
Second we take a look at the classification reports. The SGD model do not classify any observation to the second category, meaning that all the indeterminate fetuses are assigned to the category normal or abnormal by the machine itself and not by the clinician.
Even worse does the AdaM model which do not classify any observation to the abnormal category and is clearly not useful from a medical point of view.
The only two models that classify observation to all three catgories are RMSprop and AdaGrad. Of those, the model estimated using the RMSprop algorithm is better than the other as we already noted. In conlusion the RMSprop model is the best both for accuracy and classification results.

# 6 Conclusions

| classification method | accuracy on test set |
|---|---|
| random tree | 0.872 |
| random forest | 0.939 |
| ANN | 0.823 |

Table 6

We note that a random tree constructed with the four most influential variables is better than an ANN, showing that a simple but effective model can sometimes outperform complex algorithms.

Given the results obtained we see that the tree based classifiers performs better than the artificial neural network based classifier, this do not come as a surprise because the medical rule used to classify fetuses prior to the analysis is very similar to a decision tree for humans.

Artificial Neural Networks are strong and complex algorithms that have increased their popularity over the last years thanks to amazing results in artificial intelligence and deep learning, but they cannot be used for everything. We see that "simple" algorithms such as random tree/forest can still speak out for themself in classification task like the one we exposed in this report.

# 7 Bibliography

Hacker & Moore's, Essentials of Obstetrics and Gynecology 6th ed., chapther 9.

"Intrapartum Fetal Monitoring", R. Eugene Bailey, American Family Physician, 2009 Dec 15;80(12):1388-1396.

# 8 Appendix

## 8.1 NICHD

NICHD stands for ”National Istitute for Child Health and Human Developement” and was founded in 1962 to investigate human development throughout the entire life process, with a focus on understanding disabilities and important events that occur during pregnancy.

## 8.2 Baseline calculation

As defined in Hacker  Moore book, baseline rate is: ”The mean FHR rounded to increments of 5 excluding: Periodic or episodic changes, Periods of marked FHR variability, Segments of baseline that differ by more than 25 beats/min.
The baseline must be for a minimum of 2 min in any 10-min segment, or the baseline for that time period is indeterminate; in this case, one may refer to the prior 10-min window for determination of baseline”

## 8.3 Acceleration calculation

In general: Peak  15 bpm above baseline, duration  15 seconds, but ¡ 2 minutes from onset to return to baseline.
Before 32 weeks' gestation: peak  10 bpm above baseline, duration  10 seconds

## 8.4 Deceleration calculation

A gradual decrease is defined as at least 30 seconds from the onset of the deceleration to the FHR nadir, whereas an abrupt decrease is defined as less than 30 seconds from the onset of the deceleration to the beginning of the FHR nadir.If a deceleration lasts 10 min, it is a baseline change.

Variable deceleration are FHR decrease that commonly vary their onset, depth, and duration with successive uterine contractions. Prolonged deceleration is instead are represented by a decrease in the FHR from baseline that is $\geq$15 beats/min, lasting $\geq$2 min but $<$10 min in duration.