

What Do You Know About Your Child and Maternal Health?



1. Introduction

During pregnancy, a woman's body goes through many changes. These changes are entirely normal, but may become very important in case there are complications or problems. Sadly, about 700 women die each year in the United States as a result of pregnancy or delivery complications. There were 295 000 deaths during and following pregnancy and childbirth in 2017. The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented.

Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress. The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under-5 mortality to at least as low as 25 per 1,000 live births.

In light of what was mentioned above, **Cardiotocograms (CTGs)** are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more.

2. Objectives:

The scope of this project is to build several machine learning algorithms, which can predict and classify the health of the fetus with the best accuracy possible. This can be broken down into the following milestones:

1. Data Cleaning, Exploration and Feature Engineering
2. Modeling
3. Selection of best model

The best model built could benefit the medical personnel in the task of automating the diagnosis of fetus and maternal health given the information gathered by the exam saving time, budget, also help in the search of the most impactful metrics or those most correlated to any pathology and finally in the aim to early detect diseases in both patients.

3. Data Understanding:

The dataset contains 2126 records of features extracted from Cardiotocogram exams, which were then classified by expert obstetrician into 3 classes:

1. Normal
2. Suspect
3. Pathological

Features:

- **'baseline value'** FHR baseline (beats per minute)
- **'accelerations'** Number of accelerations per second
- **'fetal_movement'** Number of fetal movements per second
- **'uterine_contractions'** Number of uterine contractions per second
- **'light_decelerations'** Number of light decelerations per second
- **'severe_decelerations'** Number of severe decelerations per second
- **'prolonged_decelerations'** Number of prolonged decelerations per second
- **'abnormal_short_term_variability'** Percentage of time with abnormal short term variability
- **'mean_value_of_short_term_variability'** Mean value of short term variability
- **'percentage_of_time_with_abnormal_long_term_variability'** Percentage of time with abnormal long term variability
- **'mean_value_of_long_term_variability'** Mean value of long term variability
- **'histogram_width'** Width of FHR histogram
- **'histogram_min'** Minimum (low frequency) of FHR histogram
- **'histogram_max'** Maximum (high frequency) of FHR histogram
- **'histogram_number_of_peaks'** Number of histogram peaks
- **'histogram_number_of_zeroes'** Number of histogram zeros
- **'histogram_mode'** Histogram mode
- **'histogram_mean'** Histogram mean
- **'histogram_median'** Histogram median
- **'histogram_variance'** Histogram variance
- **'histogram_tendency'** Histogram tendency

Target:

- **'fetal_health'** Tagged as 1 (Normal), 2 (Suspect) and 3 (Pathological)

4. Data Cleaning:

Data is very clean. Checking and confirming there are non-null values in each feature and their corresponding data type. As we see below, all features even the label are numerical, therefore processing of categorical variables is something that will not be done in this project, we will only explore the distribution and meaning of each numerical.

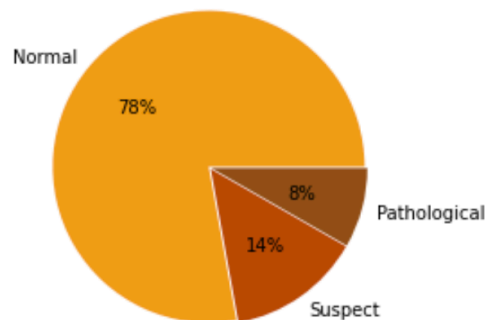
```
# Call info() on data
fetal_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   baseline value                           2126 non-null   float64
1   accelerations                           2126 non-null   float64
2   fetal_movement                          2126 non-null   float64
3   uterine_contractions                    2126 non-null   float64
4   light_decelerations                     2126 non-null   float64
5   severe_decelerations                    2126 non-null   float64
6   prolonged_decelerations                 2126 non-null   float64
7   abnormal_short_term_variability         2126 non-null   float64
8   mean_value_of_short_term_variability    2126 non-null   float64
9   percentage_of_time_with_abnormal_long_term_variability 2126 non-null   float64
10  mean_value_of_long_term_variability      2126 non-null   float64
11  histogram_width                          2126 non-null   float64
12  histogram_min                            2126 non-null   float64
13  histogram_max                            2126 non-null   float64
14  histogram_number_of_peaks                2126 non-null   float64
15  histogram_number_of_zeroes              2126 non-null   float64
16  histogram_mode                           2126 non-null   float64
17  histogram_mean                           2126 non-null   float64
18  histogram_median                         2126 non-null   float64
19  histogram_variance                       2126 non-null   float64
20  histogram_tendency                       2126 non-null   float64
21  fetal_health                             2126 non-null   float64
dtypes: float64(22)
memory usage: 365.5 KB
```

5. Exploratory Data Analysis

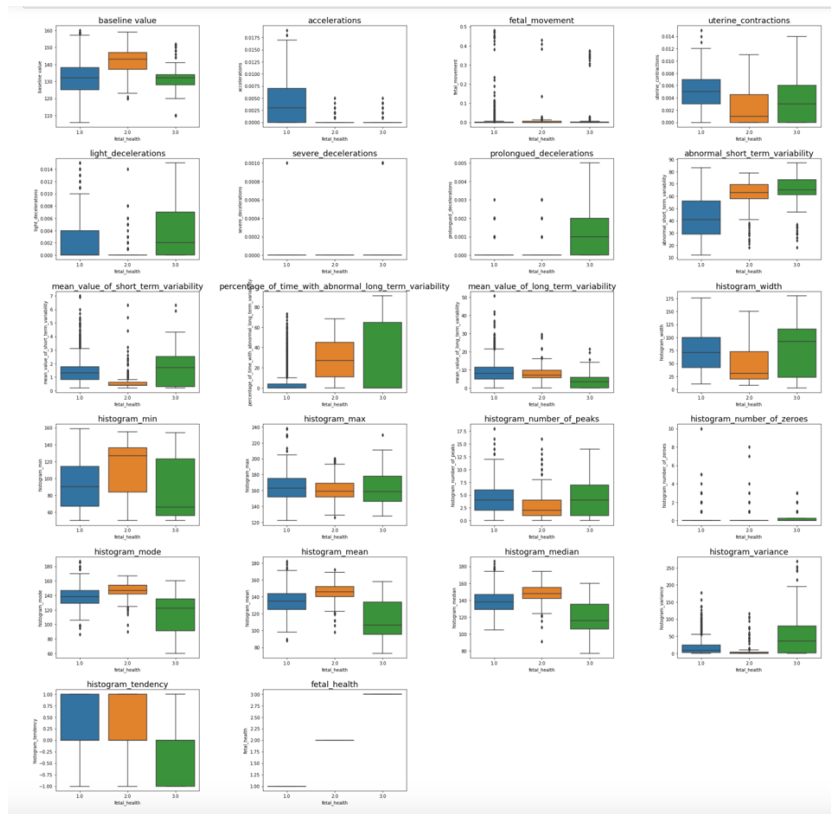
Data is imbalance between 3 classes with Normal: 78%, Suspect: 14%, Pathological: 8%

Pie chart of Fetal Heath

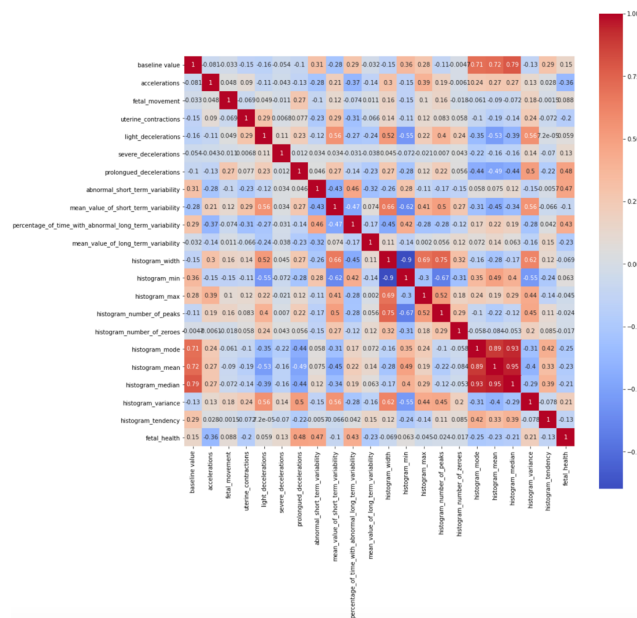


This is a case that tends to provide misleading classification accuracy. The performance measures that would provide better insight with:

- Confusion Matrix
- Precision
- Recall
- F1 Score

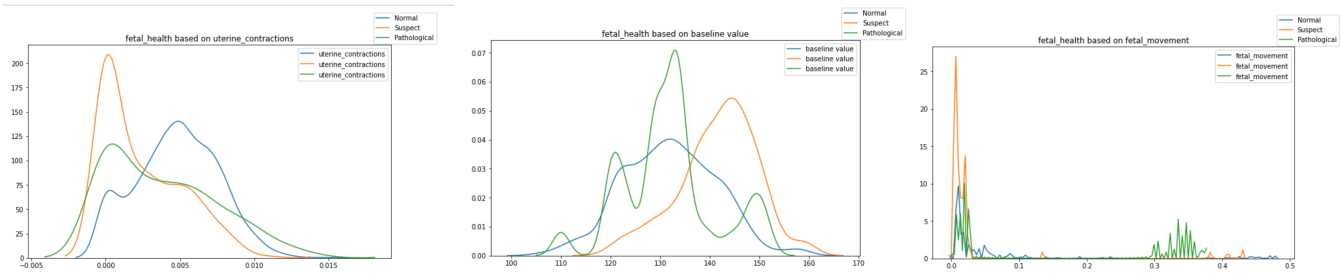


Most of the box plots show that Normal and Suspect are highly overlapping. We can see our label as a continuous variable because as the number increases in magnitude is more likely that the fetal would have a health problem, therefore we could correlate this with the features and interpret a positive Pearson correlation as a feature with direct proportion to a health problem. Now based on this assumption let's make a heat map showing the Pearson correlation of each feature to the label:



To provide a clearer correlation of each feature to fetus health, from matplotlib we import subplots to plot the correlations:

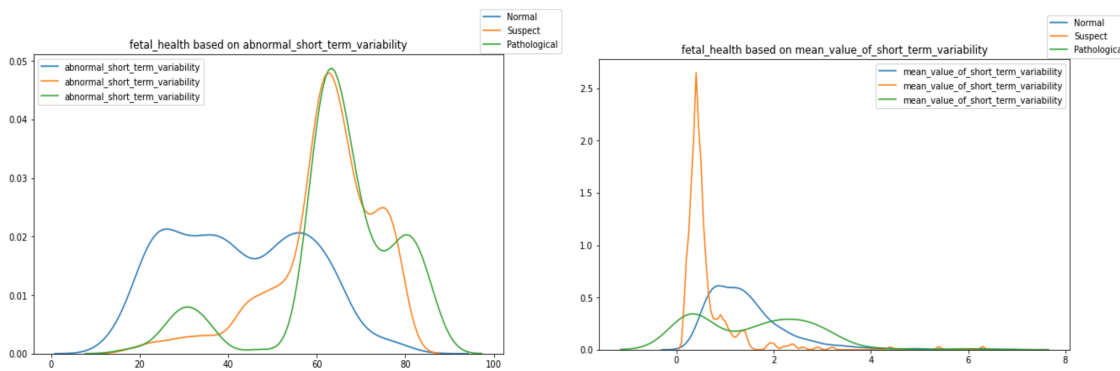
- 'uterine_contractions', 'baseline value', 'fetal_movement'**



As we see above:

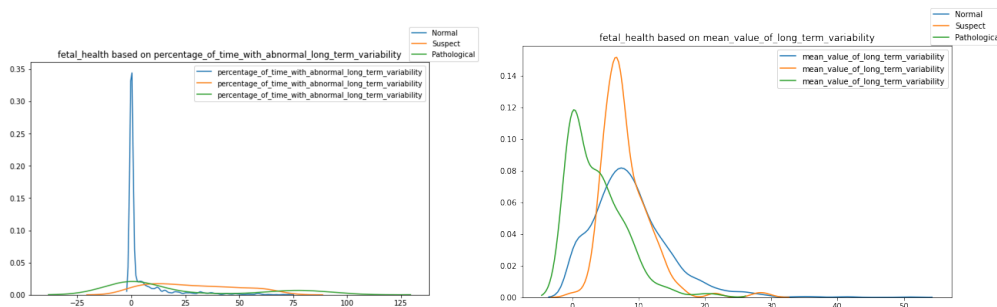
- Normal when *uterine_contractions* is > 0.0025 and Suspect when *uterine_contractions* is < 0.0025
- Pathological when *baseline value* is between 125-135 and Suspect when *baseline value* is 135+
- Suspect when *fetal_movement* is < 0.03 and Pathological when *fetal_movement* is between 0.3-0.38

- 'abnormal_short_term_variability', 'mean_value_of_short_term_variability'**



- Normal when *abnormal_short_term_variability* is < 50
- Suspect when *mean_value_of_short_term_variability* < 0.8 , Normal is 0.8-2, Pathological is > 2

- 'percentage_of_time_with_abnormal_long_term_variability', 'mean_value_of_long_term_variability'**

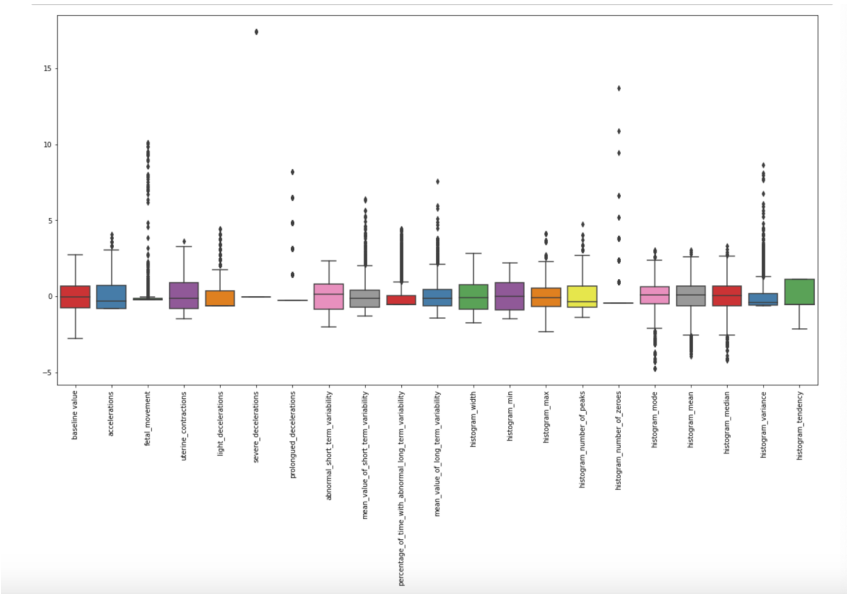


As we see above:

- Normal when *percentage_of_time_with_abnormal_long_term_variability* between (-8)-8, Suspect is 8-68 and Pathological is < -8 & > 68
- Pathological when *mean_value_of_long_term_variability* < 3 , Suspect is 3-10 and Normal is 10+

6. Model Pre-Processing:

We can see that the features had already been processed because some of them were created by binning or encoding categorical ordinal variables, such as: Light_decelerations, prolonged_decelerations, severe_decelerations, histogram_number_of_zeroes and histogram_tendency, which contain a specific number of possible values. Even the label was encoded too in this process this is why we had numbers instead of the appropriate name of the classes. About all others which were not mentioned above correspond to numerical continuous features, some of these are already standardized whereas others not yet. To fit in a model, we must scale everything to the same standardization.



7. Models:

I use Scikit-Learn for my prediction models. After scaling all features to the same range, data are partitioned into train and test sets with a size ratio of 7/3. The following models will be built and compared using their corresponding error measurement:

- Linear Regression** is an attractive model because the representation is so simple. The representation is a linear equation that combines a specific set of input values the solution to which is the predicted output for that set of input values. The model is trained with maximum number of iterations 1000. Below are the results of the model:

		precision	recall	f1-score	support
***** Linear Regression Results *****					
Accuracy	Normal	0.94	0.97	0.95	490
	Pathological	0.88	0.71	0.79	52
	Suspect	0.70	0.68	0.69	96
Recall	accuracy			0.90	638
Precision	macro avg	0.84	0.78	0.81	638
F1 Score	weighted avg	0.90	0.90	0.90	638

- **Random Forest**, like its name implies consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes become our model's prediction. As we know in tree-based models do not need to scale the features nor encoding, but these are already engineered and in order to compare the performance of models under certain context the random forest will be trained in the same way as the prior models.

It is common practice to use a grid search method for all adjustable parameters in any type of machine learning algorithm. Grid search is used to find the best hyperparameters for the model. Grid search is defined with 6 random of trees from 100 to 900. Max levels in each decision tree are randomly range from 1 to 16, and auto, square root and log2 are features to consider at every split. Gini and entropy criteria also included. The best model parameters are selected with accuracy score of 94. The model again is trained with selected parameters.

		precision	recall	f1-score	support
***** Random Forest with GridSearch Results *****	Normal	0.94	1.00	0.97	490
Accuracy : 0.932601880877743	Pathological	0.93	0.83	0.88	52
Recall : 0.932601880877743	Suspect	0.88	0.67	0.76	96
Precision : 0.930259022692456	accuracy			0.93	638
F1 Score : 0.9283964641780675	macro avg	0.92	0.83	0.87	638
	weighted avg	0.93	0.93	0.93	638

- **Gradient Boosting** is a stage-wise additive model that generates learners during the learning process. It usually has a problem that they are quick to learn and overfit training data. So, one effective way to slow down learning in the gradient boosting model is to tune the learning rate. Initially train the model with 500 trees and the number of trees can be pruned based on either the "Out-of-Bag" or "Cross-Validation" technique to avoid overfitting. We got the highest accuracy score in training and validation with learning rate at 0.25. Then we build the model again with the best learning rate to the model fit and predict on testing set; the model turns out better with higher score accuracy at 95%

		precision	recall	f1-score	support
***** Gradient Boosting Results *****	Normal	0.96	0.99	0.98	490
Accuracy : 0.9545454545454546	Pathological	0.92	0.92	0.92	52
Recall : 0.9545454545454546	Suspect	0.93	0.79	0.85	96
Precision : 0.9537670651001182	accuracy			0.95	638
F1 Score : 0.9532077463548505	macro avg	0.94	0.90	0.92	638
	weighted avg	0.95	0.95	0.95	638

8. Conclusion

We can see above that all models are above 90% of accuracy, but as we compare all the models together, Gradient Boosting model is the best model with highest accuracy at 95%.