**Name:** Nefeli Eleftheria Sextou

**Student ID:** 503

**E-mail:** pcs00503@uoi.gr

Shape

Description automatically generated

MY D8: Biomedical Data Processing and Analysis

**Brief Project Report**

Academic Year 2023-2024

**5 Feb 2024**

Dept. Of Computer Science and Engineering, University of Ioannina

**Project Goal:**

The goal is to clean, process and analyze fetal heart rate (FHR) signals included in the CTU-CHB Cardiotocography Database and proceed to utilize the produced data as input in three different classifiers and evaluate the outcomes.

**The Dataset:**

The dataset includes 552 CTGs (cardiotocography data) from the Czech Technical University (CTU, Prague) and the Hospital of Brno (UHB).

It can be found here: <https://physionet.org/content/?topic=ctu>

Information about the data included is found in the paper that accompanies the dataset : <https://bmcpregnancychildbirth.biomedcentral.com/counter/pdf/10.1186/1471-2393-14-16.pdf>

**Implementation Details:**

All code has been written in Python on Jupyter Notebooks that accompany the present report in both **.ipynb** and **html** form in folder **biomed\_project\_code**. The data is in folder **biomed\_project\_data**.

Data Cleaning and Pre-Processing:

**data\_loading\ data\_loading\_interp.ipynb (.html)**

Dataframe Preparation:

**dataframe\_creation\ age\_dataframes.ipynb (.html)**

**dataframe\_creation\ weight\_dataframes.ipynb (.html)**

**dataframe\_creation\ term\_dataframes.ipynb (.html)**

**dataframe\_creation\ ph\_dataframes.ipynb (.html)**

Classification using different feature selectors:

Linear SVC Feature Selector:

**svc\_feature\_slc\_classification\age\_lin\_svc.ipynb (.html)**

**svc\_feature\_slc\_classification\weight\_lin\_svc.ipynb (.html)**

**svc\_feature\_slc\_classification\term\_lin\_svc.ipynb (.html)**

**svc\_feature\_slc\_classification\ph\_lin\_svc.ipynb (.html)**

ExtraTreesClassifier Feature Selector:

**tree\_feature\_slc\_classification\age\_tree.ipynb (.html)**

**tree\_feature\_slc\_classification\weight\_tree.ipynb (.html)**

**tree\_feature\_slc\_classification\term\_tree.ipynb (.html)**

**tree\_feature\_slc\_classification\ph\_tree.ipynb (.html)**

***Data Cleaning and Processing:***

The first thing that had to be done was the loading of the data. The data is in .dat and .hea form which required the **Python waveform-database (WFDB) package** (<https://wfdb.readthedocs.io/en/latest/>)

After loading all available data onto a Pandas Dataframe it was necessary to:

1. Isolate the FHR Signal
2. Remove Caesarian Birth Entries
   1. This was achieved by deleting all entries with a value of 2.0 in the field ‘Deliv.type’
3. Extract Features From the ‘Comments’ feature since it contains

features that will be necessary.

**FHR Split into Stage I and Stage II:**

The FHR signal must be split into two parts since it should not be analyzed in its full form since the two parts it will be split into are representative of two stages of labor.

* **Stage I** has been limited by the creators of the database to a maximum of 60 minutes (60(mins)\*60(secs)\*4(fs) = 14000 samples) and a minimum of 30 minutes (30(mins)\*60(secs)\*4(fs) = 7200 samples )
* **Stage II** has likewise been constrained to a maximum of 30 minutes.

The split was achieved utilizing the provided starting point of stage II in the feature 'Pos.II.st.'.

**Removal of Zeros:**

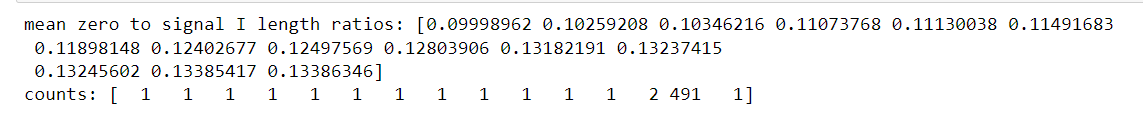
The signals have zeros which are the result of noise removal done by the dataset creators. Signals with way too many zeros must be discarded and the zeros must be handled in the remaining signals in the dataset.

The statistics of the number of zeros, as well as the mean zeros to total signal length ratio were examined for each stage separately:

**Stage I Zero Statistics and Mean Zero-Signal Length Ratio:**

**A number of zeros and numbers

Description automatically generated**

****

**Stage II Zero Statistics and Mean Zero-Signal Length Ratio:**

**A close-up of numbers

Description automatically generatedA number and a number of zeros

Description automatically generated**

Looking at the statistics and ratios for Stage I signals, it is evident that up to 15% of the signal is made up of zeros, which is not that significant and can be countered with some form of interpolation.

With regard to the entries’ Stage II data, there exist signals with a significant amount of zeros with a ratio that would amount to 80% of the signal. Some cases that are concerning include: 110 signals with a ratio equivalent to roughly 80%, 2 signals with an approximate ratio of 78%, 159 signals with a ratio of about 40% and 2 signals with a ratio of about 39%. Another odd result apparent in the data is the 4 counts of 9.60976285e+02.

The entries that correspond to the ‘concerning’ cases described above are discarded (the entire entry/row, not just the Stage II feature).

It must be noted that the mean was chosen for the ratio given it is sensitive to outliers which *must* be included in this scenario.

The rest of the remaining entries are interpolated using Forward and Backward filling.

Zeros are transformed into nan values that the Pandas replace() function can locate and treat as “missing”, and then a Forward Fill pass is followed by a Backward Fill pass to ensure all values have been filled.

In the Forward Fill pass the “missing” values are replaced with the most recent “non-missing” values before them. In the Backward Fill pass, the “missing” values are replaced with the most recent “non-missing” values after them. The results are very similar to a simple linear interpolation applied on these parts.

Continuous areas of zeros were not removed, they were filled with a corresponding valid value as described above.

**A graph of a graph of a graph

Description automatically generated with medium confidenceExample of the Interpolation of a Stage I Signal:**

**A graph of a graph

Description automatically generated with medium confidenceExample of the Interpolation of a Stage II Signal:**

It is now possible to move on to feature extraction from the two parts of the FHR signals independently.

**FHR Feature Extraction:**

Specific features must be extracted from the FHR signal. The FHR signal contains heart rate variability (HRV) information. There are several types of methods that help extract features from such a signal and they include time domain methods, frequency domain methods, geometric methods, time-frequency methods and non-linear methods.

The features that will be extracted correspond to the frequency domain, time domain, time-frequency domain and some entropy based non-linear methods.

**Time Domain Features:** mean and median NN intervals (mean\_nni, median\_nni), SDNN, RMSSD, NNI20, PNNI20

**Frequency Domain Features:** LF/HF Ratio, Total Power, VLF

**Time-Frequency Features:** Haar Wavelet St. Deviation, Mean and Shannon Entropy

**Entropy Features:** Sample Entropy, Bubble Entropy and Shannon Entropy

An experimental set of features was added that quantifies the difference of some of the feature values of the categories above. The choice was arbitrary. If these features offer good information, they will be retained in the feature selection stage.

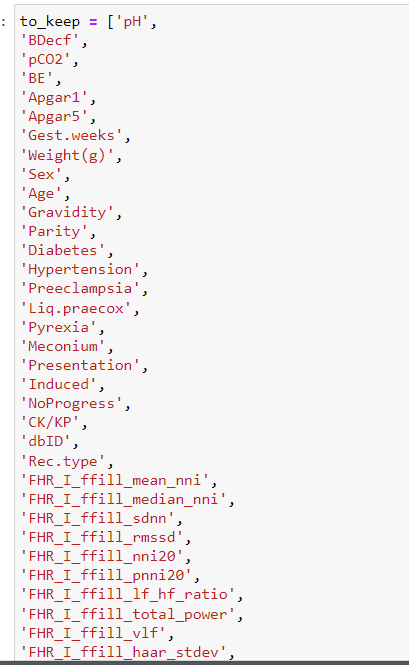
**Difference Features:** the absolute difference of nni20, LF/HF and of the Haar Wavelet St.Dev.

*Note about the Haar Wavelet:* The Haar wavelet is applied with multiple scale settings. The goal is to choose the scale that provides a min median case for all signals in each stage and then extract more information from the detail coefficient vector of that scale. The **optimal scale** is found to be **1** for both stages independently.

The features utilizing the Shannon entropy had to be removed because they resulted in **-inf**  in all entries.

After this a “main” dataframe was created by retaining the features in the screenshot on the next page. The Shannon entropy features are commented out.

The 'BDecf', 'pCO2', 'BE' columns contained a few null values. The corresponding entries were removed.

A screenshot of a computer code

Description automatically generated

***Data Preparation and Class Creation:***

There are four target variables for which classification will be applied: the mother’s age (feature: ‘Age’), the neonate’s weight (feature: ‘Weight(g)’), the gestational term in weeks (feature: ‘Gest.weeks’) and the pH (feature: ‘pH’). Classes must be created for each of the above.

**‘Maternal Age’ Classes for the Mother’s Age:**

The classes for the mother's age are,initially, not chosen arbitrarily but based on methodology used on a well cited paper about the relation of maternal age to adverse pregnancy outcomes provided in the link below:

<https://bmcpregnancychildbirth.biomedcentral.com/articles/10.1186/s12884-019-2400-x>

Classes:   
A: <= 17 years old  
  
B: 18-28 years old  
  
C: 29-39 years old  
  
D: >=40 years old

However, this resulted in an imbalanced dataset so the problemed was turned into one of binary classification with classes:  
  
A: <= 29 years old  
  
B: > 29 years old

**‘Neonate Weight’ Classes for the Neonate’s Weight:**

The initial classes were chosen with reference to:

* The following resource that is partnered with some world renowned medical schools (Yale,UCSF,UCSD and more): <https://emedicine.medscape.com/article/938854-overview?form=fpf>
* The Mayo Clinic : <https://www.mayoclinic.org/diseases-conditions/fetal-macrosomia/symptoms-causes/syc-20372579>
* Penn Medicine Lancaster General Health: <https://www.lancastergeneralhealth.org/health-hub-home/motherhood/the-first-year/your-newborns-weight-gain>

Neonate Weight is classified as follows:

ELBW: Extremely low birth weight -> weight < 1000 g  
  
VLBW: Very low birth weight -> weight < 1500 g  
  
LBW: Low birth weight -> weight < 2500 g  
  
NORM: Normal birth weight -> weight >=2500 g and <=4000 g  
  
MACRO: Macrosomia -> weight > 4000 g

The ELBW and VLBW did not exist in my dataset, and the remaining three classes were imbalanced. The new classes that were created were the following:

ABNORM: Low birth weight -> weight < 2500 g

(includes abnormal ELBW, VLBW and LBW cases)  
  
NORM: Normal birth weight -> weight >=2500 g and <=4000 g  
  
ABNORM: Macrosomia -> weight > 4000 g

(includes MACRO abnormal cases)

**‘Gest Term’ Classes for the Gestational Term in Weeks:**

According to The American College of Obstetricians and Gynecologists (ACOG) : <https://www.acog.org/clinical/clinical-guidance/committee-opinion/articles/2013/11/definition-of-term-pregnancy>, pregnancy terms can be classified in weeks as follows:

ET: Early Term -> weeks>= 37 and <=38  
  
FT: Full Term -> weeks>=39 and <=40  
  
LT: Late Term -> weeks=41  
  
P: Postterm -> weeks>=42

While all four classes appeared, there was prominent class imbalance. To find a good midway point between interpretability and class balance, the 'FT' class can remain the same and the rest of the classes can be merged as 'OT' (other). So, the two categories will be full term pregnancies and 'other' pregnancies where some type of pathological phenomenon may exist.

OT: Other -> weeks>= 37 and <=38  
  
FT: Full Term -> weeks>=39 and <=40  
  
OT: Other -> weeks=41  
  
OT: Other -> weeks>=42

**‘pH risk Classes for the pH:**

In order to create two classes for the pH feature, it is necessary to find a threshold value. Many contesting scientific opinions on the value of this threshold exist and with regard to which ranges of values may indicate what pathological phenomenon or risk indicator. The ideal threshold value is the subject of ongoing research and there is no agreement on a threshold value that may be used to discern between normal and abnormal situations but those proposed are usually smaller than 7.2.

The approach followed here was the following:

The instances of unique pH values were counted:

A number grid with numbers

Description automatically generated

Then, they were grouped and total sums per group were calculated:

{6.87 6.92 6.93 6.98 6.99 7.} -> total count = 9  
  
{7.02 7.03 7.05 7.07 7.08 7.09 7.1} -> total count = 17  
  
{7.11 7.12 7.13 7.14 7.15 7.16 7.17 7.18 7.19 7.2}-> total count = 82  
  
{7.21 7.22 7.23 7.24 7.25 7.26 7.27 7.28 7.29 7.3} -> total count = 92  
  
{7.31 7.32 7.33 7.34 7.35} -> total count = 22

Looking at the existing pH values and their corresponding count values, there are two good split options that will lead to balanced classes. They can be considered equivalent. The first is using 7.2 as the splitting point for the two classes (class A : 108 elements, class B: 114 elements). The second is using 7.21 as the splitting point for the two classes (class A: 115 elements, class B:107 elements).

Instead of class A and B we can name the classes Risky 'RISK' and Non-Risky 'NORISK' given the fact there may be loose corellation to CTG data and pH values that indicate asphyxiation of the neonate or other pathological phenomena.

Given the dataset information and the information available in the paper that accompanies the initial dataset where a lot of clinical details are briefly explained, 7.2 is chosen as the split point.

***Classification:***

All implementations mentioned refer to existing python implementations that are well established and widely used.

The three classifiers that will be tried are the:

* K-Nearest Neighbors Classifier
* Random Forest Classifier
* XGBoost XGBClassifier

The features are selected using two different methods:

* Linear SVC based: An application of Linear Support Vector classification with L1 regularization as a penalty term that discourages the use of unnecessary features in the model. Features with non-zero coefficients are considered important in making predictions, while features with zero coefficients are ignored.
* ExtraTreesClassifier based: It builds a set of multiple decision trees through bootstrapping, where each tree is trained on a random subset of the training data. Features are randomly selected for splitting each node in a decision tree and this contributes to feature decorrelation. The results of the tree predictions are aggregated to achieve classification.

Data is split in 80% train and 20% test.

Metrics used to evaluate the results:

* The Confusion Matrix
* Accuracy = (TP + TN)/(TP+TN+FP+FN)
* Precision = (TP)/(TP+FP)
* Recall = (TP)/(TP+FN)
* F1-Score = 2\*((Precision\*Recall)/(Precision+Recall))
* ROC Curve :

True Positive Rate( Recall) – False Positive Rate(FP/(FP+TN)) and AUC

\*higher values closer to 1.0 are better for all

**For XBGClassifier, one extra evaluation method was utilized:**

The **SHAP (SHapley Additive exPlanations)** is a Python library designed for interpreting the output of machine learning models. It provides a framework for understanding the contributions of each feature in model predictions.

<https://shap.readthedocs.io/en/latest/index.html>

It is based on Shapely Values which are a concept from cooperative game theory that has found application in machine learning model interpretability. ( <https://en.wikipedia.org/wiki/Shapley_value> )

In brief:

In cooperative game theory, players form coalitions (subsets) to achieve a collective outcome. When used in machine learning, the players are the features, and the coalitions are subsets of features.

The Shapley values calculate each player’s/feature’s marginal contribution to every possible coalition it could join.

marginal contribution **=** (the model's prediction with a feature in the coalition) **–** (the model's prediction with a feature NOT in the coalition)

They then take the average of the marginal contributions over all possible coalitions. In this manner, each feature is attributed a fair share of the total contribution and all possible interactions with other features are considered.

Main lines of interpretation:

**Positive Shapley Value:** the presence of that feature contributes positively to the model's prediction.

**Negative Shapley Value:** the absence of that feature contributes positively to the model's prediction.

The magnitude reflects the strength of the contribution. The larger the magnitude, the more significant the feature’s presence is for the model's output.

**Linear SVC Feature Selection Results:**

**Maternal Age**

**A black and white text

Description automatically generated**

**Neonate Weight**

**A computer code with black and white text

Description automatically generated**

**A close-up of a computer screen

Description automatically generatedGest Term**

**A black and white text

Description automatically generatedpH risk**

ExtraTreesClassifier Feature Selection Results:

A white text with black text

Description automatically generatedMaternal Age

Neonate Weight

A white text with black lines

Description automatically generated

A text on a white background

Description automatically generatedGest Term

pH risk

A black text on a white background

Description automatically generated

***Maternal Age (Linear SVC) Results***

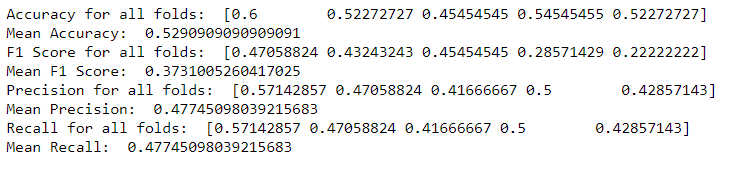
**KNN**

**A screenshot of a computer

Description automatically generated**

**A graph with a line

Description automatically generated**

****

**Random Forest**

**A screenshot of a computer

Description automatically generated**

**A graph of a positive rate

Description automatically generated with medium confidence**

**A number on a white background

Description automatically generated**

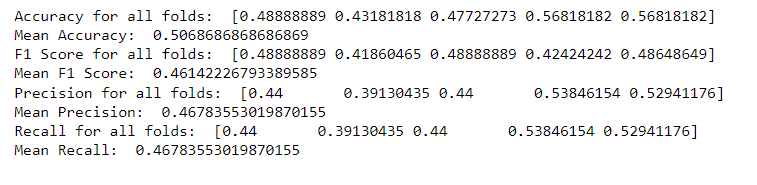
**XGBClassifier**

**A screenshot of a computer

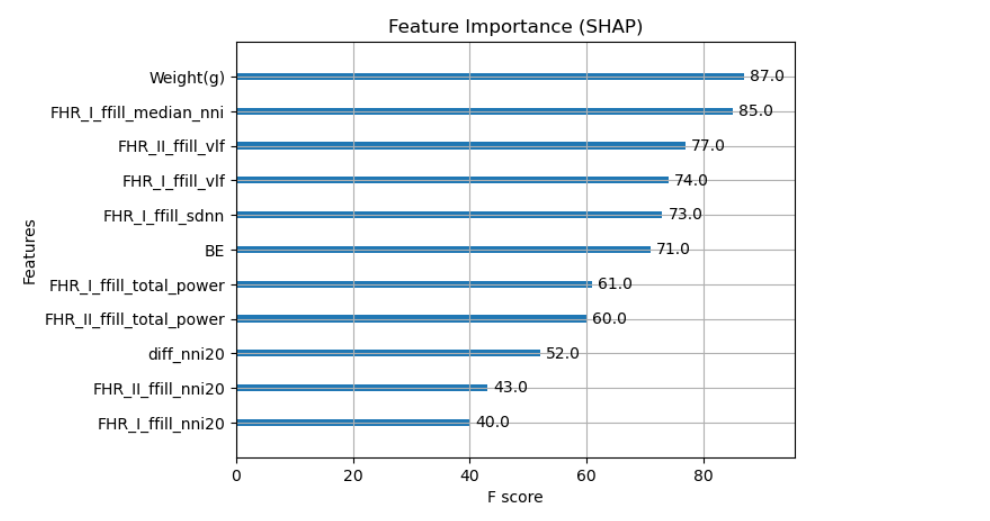
Description automatically generated**

A graph of a person

Description automatically generated



***A graph with red and blue squares

Description automatically generated***

***Neonate Weight (Linear SVC) Results***

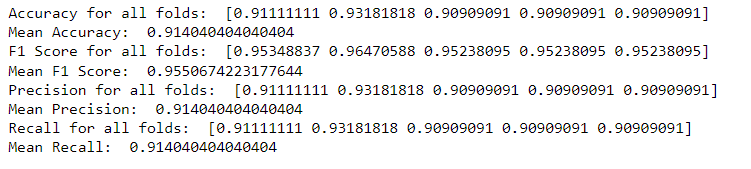
**KNN**

**A screenshot of a computer

Description automatically generated**

A graph with a line

Description automatically generated



**Random Forest**

A screenshot of a computer

Description automatically generated

A white background with black numbers

Description automatically generatedA graph with a line

Description automatically generated

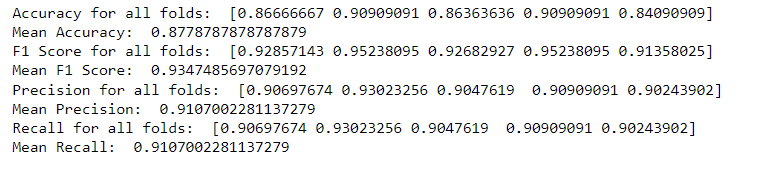
**XGBClassifier**

A screenshot of a computer

Description automatically generated

A graph with blue lines

Description automatically generated



A graph with blue bars

Description automatically generated

A graph with numbers and a number of numbers

Description automatically generated with medium confidence

***Gest Term (Linear SVC) Results***

**KNN**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated

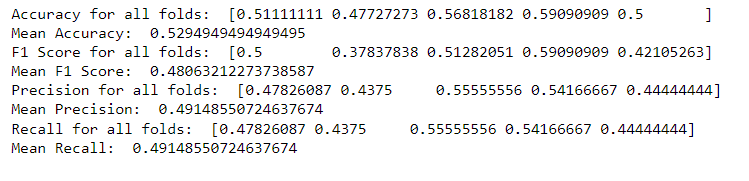
A number on a white background

Description automatically generated

**Random Forest**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated

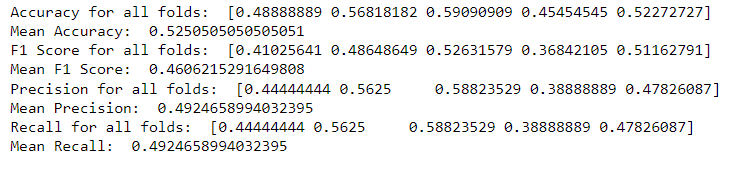
**XGBClassifier**

A screenshot of a computer

Description automatically generated

A graph of a staircase

Description automatically generated



A graph with numbers and letters

Description automatically generated

A graph with red and blue rectangles

Description automatically generated

***pH risk (Linear SVC) Results***

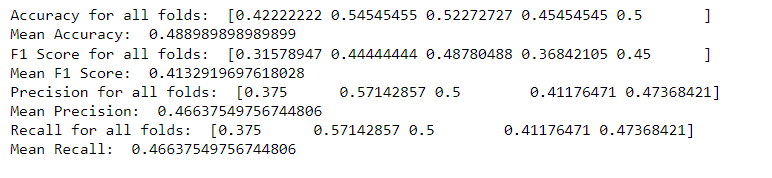
**KNN**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated



**Random Forest**

A screenshot of a computer

Description automatically generated

A graph with blue lines

Description automatically generated

A number on a white background

Description automatically generated

**XGBClassifier**

A screenshot of a computer

Description automatically generated

A graph of a positive result

Description automatically generated with medium confidence

A number on a white background

Description automatically generated

A graph with blue lines

Description automatically generated

A graph with numbers and symbols

Description automatically generated

***Maternal Age (ExtraTreesClassifier) Results***

**KNN**

**A screenshot of a computer

Description automatically generated**

A graph of a positive rate

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

**Random Forest**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated

A white background with numbers and symbols

Description automatically generated

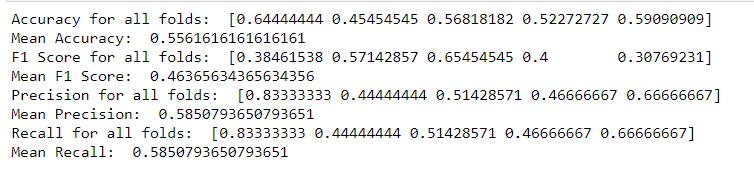
**XGBClassifier**

**A screenshot of a computer

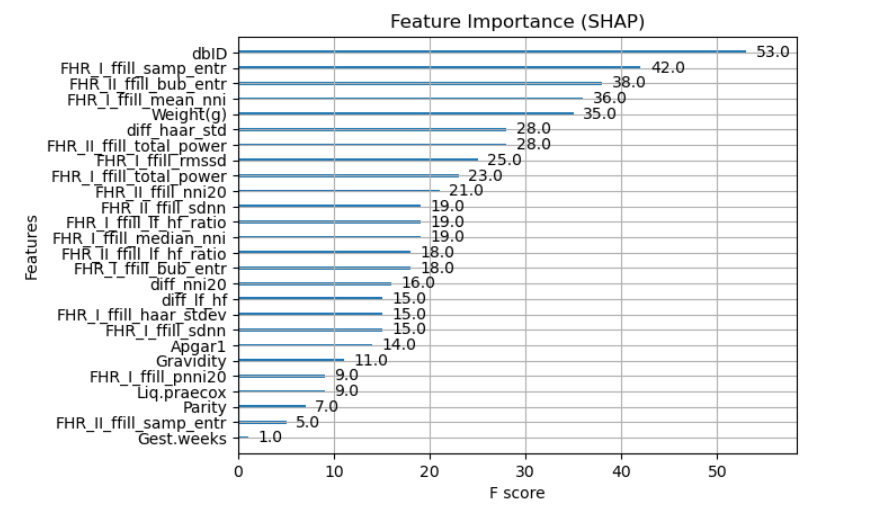
Description automatically generated**

A graph of a graph

Description automatically generated with medium confidence



A graph with numbers and symbols

Description automatically generated

***Neonate Weight (ExtraTreesClassifier) Results***

**KNN**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated

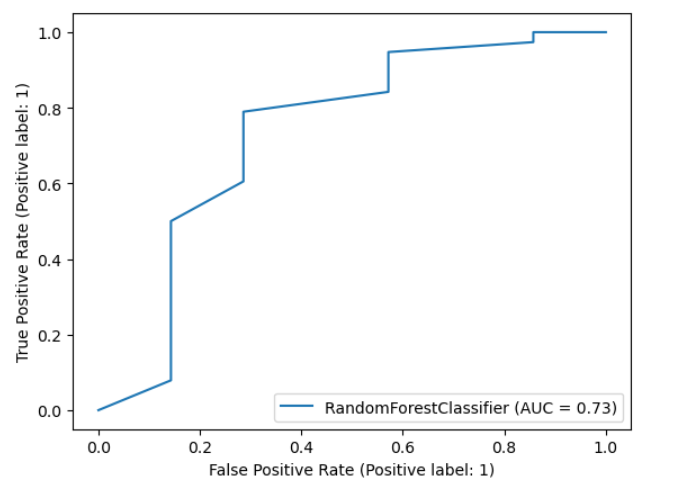
A white background with numbers

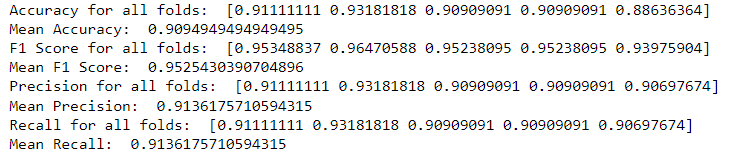
Description automatically generated

**Random Forest**

A screenshot of a computer

Description automatically generated





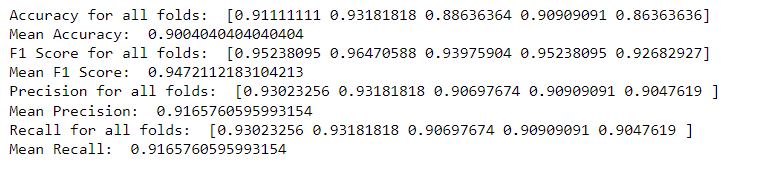
**XGBClassifier**

A screenshot of a computer

Description automatically generated

A graph of a line

Description automatically generated with medium confidence



A graph with red and blue rectangular boxes

Description automatically generatedA graph with numbers and letters

Description automatically generated

***Gest Term (ExtraTreesClassifier) Results***

**KNN**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated

A screenshot of a computer

Description automatically generated

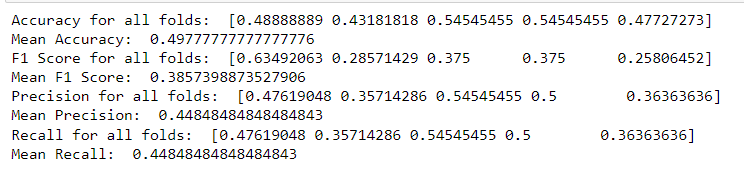
**Random Forest**

A screenshot of a computer

Description automatically generated

A graph with a line

Description automatically generated



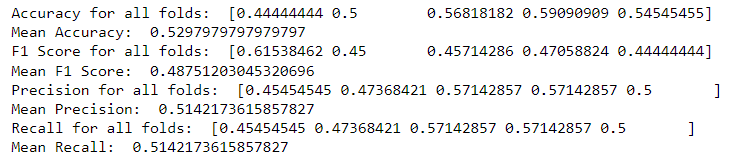
**XGBClassifier**

A screenshot of a computer screen

Description automatically generated

A graph of a line

Description automatically generated with medium confidence



A graph with numbers and letters

Description automatically generated

A graph with red and blue squares

Description automatically generated

***pH risk (ExtraTreesClassifier) Results***

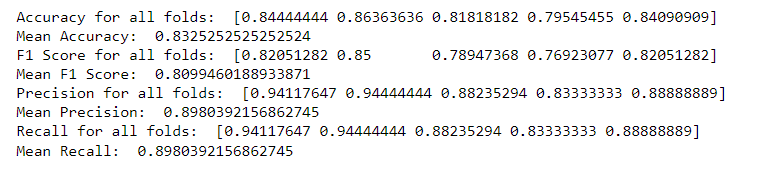
**KNN**

A screenshot of a computer

Description automatically generated

A graph with a blue line

Description automatically generated



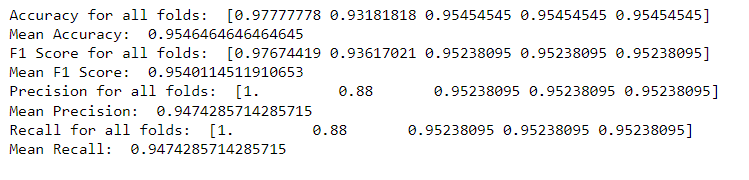
**Random Forest**

A screenshot of a computer

Description automatically generated

A graph with a positive rate

Description automatically generated with medium confidence

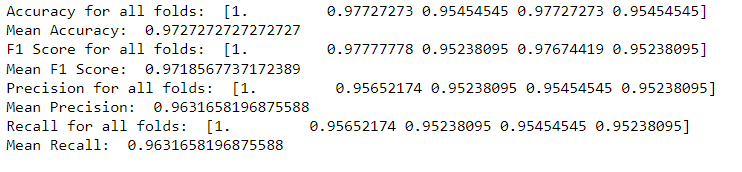


**XGBClassifier**

**A graph of a positive rate

Description automatically generatedA screenshot of a computer

Description automatically generated**



A graph with numbers and a bar graph

Description automatically generatedA graph with blue bars

Description automatically generated

***Results Summary:***

* Excellent performance for target **‘ph risk’**
* Good Performance for target **‘Neonate Weight’ (but biased by the NORM class)**
* Mediocre to poor performance for targets **‘Mother’s Age’** and **‘Gest Term’**
* The ExtraTreesClassifier based feature selection method is a better choice since it leads to better outcomes.

***Some Possible Problems:***

* Signals with useful information in Stage I were discarded due to too many zeros in Stage II
* Mistakes/Misunderstandings in pre-processing or the need for alternative methods to be applied at any part of that process.
* Individual and more refined processing being required that is assisted in part by relevant clinical information
* Lack of meaningful relationships amongst available data, requirement of further data extraction efforts.

***Further work:***

* Utilize the XGBClassifier data to explore the outcomes of removal or choice of features based on the SHAP information
* Manual feature extraction based on the overall information extracting via SHAP
* If that fails, further exploration of other machine learning models, feature extraction methods and data processing techniques.