Czech Technical University in Prague

Faculty of Electrical Engineering and Computer Science

Bachelor Thesis

Classification of Intrapartum Fetal Heart Rate Signals

Author: Mohamed Dariwsh

Supervisor: Ing. Jiří Spilka, Ph.D.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Safwat Mohamed** Personal ID number: **464324**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Electrical Engineering and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Classification of Intrapartum Fetal Heart Rate Signals

Bachelor's thesis title in Czech:

Klasifikace intrapartálních záznamů srdeční frekvence plodu

Guidelines:

Fetal Heart Rate (FHR) monitoring is used in every day clinical practice to help obstetricians assess fetal health status during delivery. However, the detection of fetal acidosis that allows relevant decisions for operative delivery remains a challenging task. This project aims to create a machine learning model for FHR classification that can predict whether fetuses suffer from acidosis.

1. Briefly study fetal heart rate characteristics and its changes related to fetal acidosis.
2. Study features and machine learning methods used for signal analysis and classification.
3. Select relevant features, methods (argument selection), implement them or use existing implementations.
4. Perform systematically experiments on CTU-UHB cardiotocography database.
5. Critically analyze results and interpret the final model (important features, why a prediction was made, typical errors the model has made).

Bibliography / sources:

[1] Georgieva A, Abry et al. Computer-based in-trapartum fetal monitoring and beyond: A review of the 2nd workshop on signal processing and monitoring in labor (October 2017, Oxford, UK). Acta Obstet Gynecol Scand. 2019;98:1207-1217.
[2] P. Abry, J. Spilka, R. Leonarduzzi, V. Chudáček, N. Pustelnik, M. Doret Sparse learning for Intrapartum fetal heart rate analysis In Biomedical Physics Engineering Express 4(3) 034002, 2018.
[3] A. Petrozziello, C. W. G. Redman, et al. "Multimodal Convolutional Neural Networks to Detect Fetal Compromise During Labor and Delivery," in IEEE Access, vol. 7, pp. 112026-112036, 2019.

Name and workplace of bachelor's thesis supervisor:

Ing. Jiří Spilka, Ph.D., Department of Biomedical Engineering and Assistive Technology, CIIRC

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: _____ Deadline for bachelor thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

Ing. Jiří Spilka, Ph.D.	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt _____ Student's signature _____

Declaration

I declare that I have written my Bachelor Thesis ‘Classification of Intrapartum Fetal Heart Rate Signal’’ on my own and I have used only cited literature and other professional sources.

Prague 13.8. 2020

Acknowledgment

I would like to express my gratitude for whomever supported my journey through my bachelors, including my professors, my friends and my family, also special thanks to Jiří Spilka for his continuous support and guidance through my project which I learned a lot from.

List of Abbreviations

AUC	Area Under receiver operating Characteristic
BpM	Beats per Minute
CTG	Cardiotocogram
CV	Cross-Validation
DECG	Fetal Electrocardiogram
FHR	Fetal Heart Rate
FIGO	International Federation of Gynaecology and Obstetrics
FN	False Negative
FP	False positive
kNN	k-Nearest Neighbour
LR	Logistic Regression
LTV	Long Term Variability
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
SE	Sensitivity
SMOTE	Synthetic Minority Over-sampling Technique
SP	Specificity
STV	Short Time Variability
SVM	Support Vector Machine
TP	True Positive
TN	True Negative

Abstract

Continuous monitoring of the fetal heart rate (FHR) signal has been widely used to allow obstetricians to obtain detailed physiological information about newborns. The objective of this thesis is to briefly study fetal heart characteristics and investigate machine learning and data mining techniques to gain vital information regarding the fetal health during labor and whether or not an intervention would be required. The thesis focuses on using machine learning techniques for FHR classification. A comprehensive set of 41 features is extracted from a large intrapartum CTG database, collected at Brno University Hospital (Czech Republic). The dataset contains 4462 subjects, from which 86 are positive cases with pH <= 7.05 and 4376 as negative cases with pH > 7.05. First, we examine the importance of individual features and measure their area under the ROC curve (AUC). The most relevant feature is energy_tot which ranked at the top with an AUC score of 0.61, followed by energy_VLF and mad, both with an AUC score of 0.60. Due to the clear misproportion between the positive and negative classes, AUC metric was taken into consideration while assessing the model performance, while also carefully monitoring the sensitivity and specificity of the model. Three models are used to classify the data, logistic regression, SVM and kNN. Double-loop cross-validation is performed to provide unbiased performance estimation; the highest AUC was obtained using logistic regression with AUC of 0.62. Further, we propose multiple hypotheses and discuss their contribution to the performance of the model (logistic regression) which include the use of feature extraction algorithm (PCA is used) to reduce the feature subspace, SMOTE analysis to oversample the minority class, as well as, measuring the performance after adding more features or more segments.

Keywords

Machine learning, Fetal heart rate, Correlation analysis, Statistical models, Evaluation metrics, Feature extraction, Over-sampling,

Machine learning

I. Definition

Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions. Most machine learning algorithms can be divided into two main categories: supervised learning and unsupervised learning techniques. In this thesis, we are only focusing on supervised learning.
In supervised learning, the model is given the data with its labels and the model has to come up with a learning function that best maps the input to its respective output,

$$Y = f(X)$$

The goal is to approximate the mapping function f so when it is subjected to a new input vector X, it predicts its output label Y. This is achieved by feeding the model training labeled data and then evaluating its approximation by testing it on a non labeled data. For instance, the problem we are discussing in this thesis falls under this category as we are given data of patients with an indication of whether it is positive or negative, and one of the main goals is to find the mapping function that can best predict the patients' diagnosis.
However, in unsupervised learning, the labels are unknown, so the model has to draw inferences from datasets consisting of input data without labeled responses, mainly used to find hidden patterns, the algorithm tries to separate the input data into clusters.

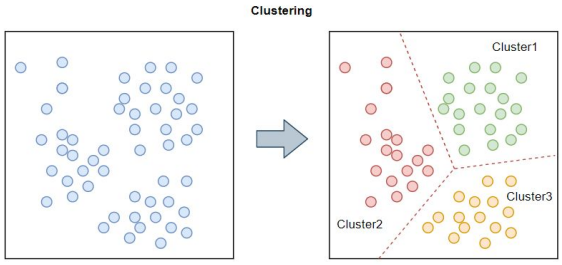


Figure1: Dividing the data into 3 clusters

A good example is collaborative filtering which was proposed by Dave Goldberg and later implemented by companies like Netflix, which is an algorithm based on supervised learning that helps divide the input data into clusters that behaves similarly, for instance, viewers with similar taste and likings are likely to fall under the same cluster, which facilitates the process of building a recommendation systems or ad targeting.

II. Contribution towards medicine

Machine learning is swiftly infiltrating many areas within the healthcare industry, from diagnosis and prognosis to drug development and epidemiology, with significant potential to transform the medical landscape. The field of medicine has so far relied heavily on heuristic approaches, whereby knowledge is acquired through experience and self-learning, which is imperative in the highly variable healthcare environment[1]. Moreover, machine learning algorithms can crunch massive amount of data which helps the algorithm yields to better results and far more stable, it is used in many diverse areas within the medical field, IBM Watson Genomics, for instance, is a prime example of the usage of machine learning in the role of identifying the disease and diagnosing it, where the model extract massive amounts of unstructured data and output meaningful insights out of it, and with the help of statistics and data mining techniques, a successful machine learning model can do much more than predicting the outcome, for example, it can explain how the features of the problem contribute to the outcome, in the medical field, scientists and doctors are not just interested in diagnosing the patient correctly but also what led to the diagnosis, so it is extremely vital to study the features and its relation with the outcome.

Introduction

I. Continuous fetal heart rate

Continuous fetal heart rate (FHR) monitoring remains the mainstay of intrapartum fetal surveillance. One of the goals is to prevent adverse labor outcomes such as hypoxic-ischemic encephalopathy by identifying incipient hypoxia/ischemia in a previously healthy fetus, at a time when intervention can prevent or mitigate permanent injury. Its goal is not considered to be the identification of infection during labor or trauma related to the delivery. However, preexisting injury, anomaly, or antenatal hypoxia may be suspected based on a deviant FHR pattern [2]. FHR is derived from the fetal heartbeat signals which are measured through the mother's abdomen. Classically based on the visual evaluation of FIGO criteria, FHR characterization remains a challenging task that continuously receives intensive research efforts, as though those efforts, we can gain vital information on fetal status during both antepartum and intrapartum periods.

II. Acidosis

Acidosis means a high hydrogen ion concentration in the tissues. Acidaemia refers to a high hydrogen ion concentration in the blood and is the most easily measured indication of tissue acidosis. The unit most commonly used is pH, which is the log to base 10 of the reciprocal of the hydrogen ion concentration[3]. The consequences of acidosis depend on its severity and duration and also the condition of the fetus before the insult, which if it continues for a long period, the baby can suffer from lifelong disabilities, brain damage, or even death. Acidosis occurs as a

result of tissue hypoxia and it is unclear whether the consequences of this process are due primarily to acidosis or hypoxia. What has become clear over the past decade is that the consequences of hypoxia/acidosis are very different, depending on whether this is acute or chronic. The normal human fetus is adapted to survive labor and has compensatory mechanisms that allow it to withstand even very severe hypoxia and acidosis for short periods, cf. several studies that examined neurological outcomes of neonates subject to severe asphyxia during delivery [3].

III. Goals

The advances in modern obstetric practice allowed many robust and reliable machine learning techniques to be utilized in classifying fetal heart rate signals. Not only machine learning provides us with multiple techniques that can help us identify hidden patterns, but also some other techniques are meant to investigate the features and explain the role it plays towards the target. In this thesis, we aim to analyse and further explore features contributing to fetal hypoxia and therefore acidosis, In addition to that, we fit multiple statistical models and study their behavior to point out their advantages and disadvantages.

Materials: Database and Features Exploration

I. Database

FHR data were collected from Brno university hospital in the Czech republic. Data were recorded using STAN S21 or S23 devices via internal fetal scalp electrodes, which combines standard CTG monitoring with a concurrent assessment of the fetal ECG, providing multiple information regarding fetal health, in which pH is one of them and the one we are interested in. The database consists of 4462 recordings, documented by obstetricians in charge of delivery. The FHR signals were obtained either directly using a Doppler ultrasound (US) probe placed on the mother's abdomen, or from direct electrocardiogram (DECG) measured internally by a scalp electrode attached to the fetal scalp [4]. In our dataset, we have a total of 2125 positive samples and 106506 negative samples. Among those positive records, 86 exists in the first stage while 65 are in the second stage. For our model, we are going to use samples coming from the first stage only, which is close to delivery. Previous contributions by the authors reported significant differences between the statistics of the temporal dynamics of the first (dilatation) and second stages of labor [5] so we chose only the first stage for our analysis which consisted of 4462 samples.

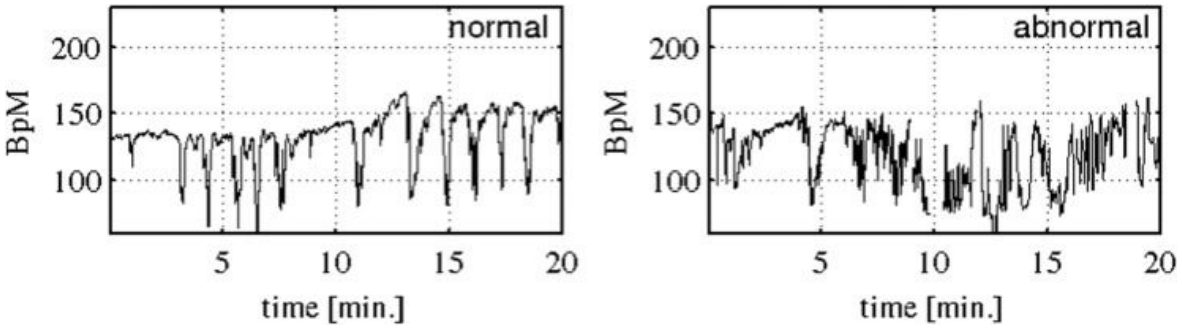


Figure 2:Typical FHR records for normal and abnormal cases [4]

In this work, 41 features were extracted from the FHR signals, coming from different domains. Different feature domains represent different points of view of the CTG, ranging from FIGO-based features that try to emulate the information extractable by eye, to time-domain features that are very understandable to clinicians yet almost impossible to be estimated bya naked eye, to more complex feature domains, which quantify the signal using frequency and nonlinear analysis tools [4]. The comprehensive description and detailed computation of all features is described in [5, 7].

- FIGO-based features: baseline, number of acceleration/deceleration, and long term variability. In this work for the extraction of the FIGO-based features, which describe the macroscopic properties of the FHR [4].
- Time-domain features: quantifying Short Term Variability (STV) and Long Term Irregularity (LTI) [4].
- Frequency domain: energy in different frequency bands [4].
- Scale invariance features: Hurst exponent estimated on different scales, and multifractal coefficient c_1 to c_4, cf. [4] for computation and details. All these features try to quantify the scale invariance of the signal under investigation [4].

II. Features exploration

We start by performing exploratory data analysis to find insights and summarize the main characteristics, it is a vital step that allows us to get the data prepared for further processing using machine learning models. The data consist of 4462 subjects and 49 features(41 extracted from FHR signals) in which the pH level is among those features which serve as the target we try to predict. Patients having their pH level below 7.05 are potentially at the risk of having their baby diagnosed with acidosis, while the rest of the 49 features are also numerical continuous entities except for the categorical feature “segStage”.

Mean	Std	Median	Mad	Skewness	Kurtosis	stressRatio	accNumb	accNumb
areaDecelTriangle	LTV_FIGO_bpm	STV_Sonicaid	bslnBeta0	bslnBeta1	decDtrdMAD	energy_VLF	energy_LF	energy_HF
energy_LF_HF	energy_tot	spectrum_slope	H29	MF_c1_29	MF_c2_29	MF_c3_29	MF_c4_29	

Table1: List of features

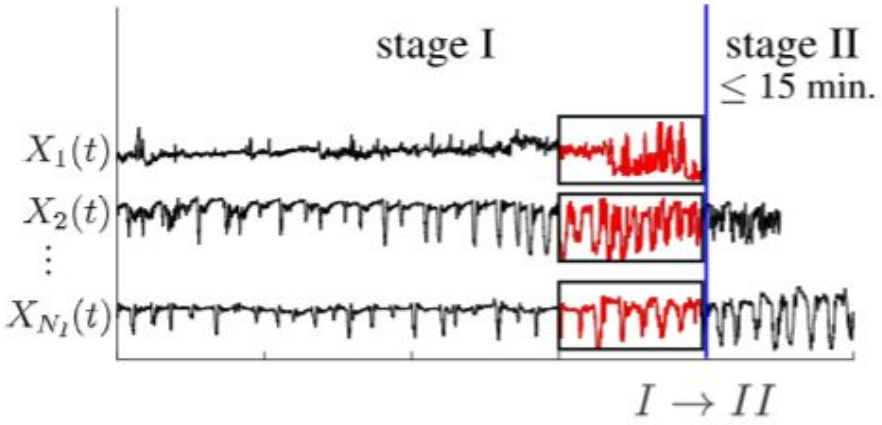


Figure3: Stage splitting Analyzed FHR data marked by the rectangular box [6]

We firstly discuss the "segStage" feature, as it is the only categorical feature (aside from 'year'), which has three values divided into two stages during pregnancy, the first stage describes the mother during dilatation, the second stage is during the labor, and thirdly the stage which is marked as "12" and occurs in between the stage 1 and stage 2. The data collected from the first stage which was chosen for the analysis comes from continuous monitoring of the FHR signals, ending less than 20 minutes before delivery [7] as shown in Figure 3 above.

III. Target analysis and correlation

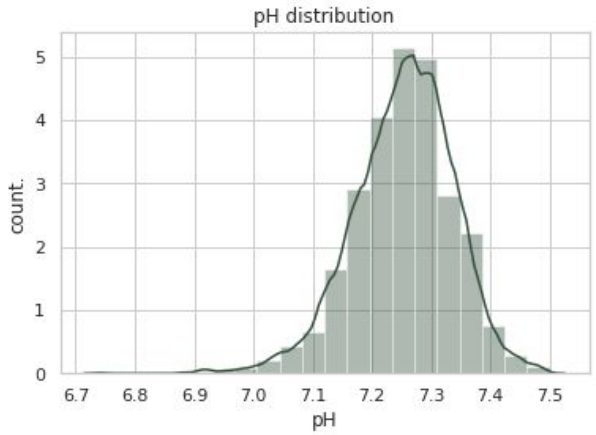


Figure4: pH distribution for positive cases

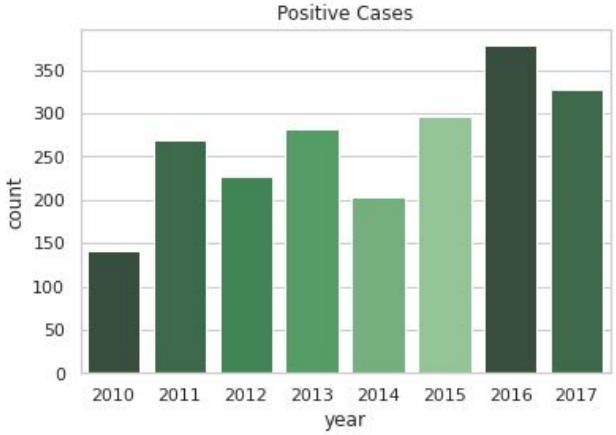


Figure5: Positive cases (2010-2017)

The recordings from the first stage are processed and pH is measured which leads to 4462 samples which were divided into normal at which pH measurement is > 7.05 and pathological samples having pH <= 7.05. A good practice is to study the correlation between the features and the target, also among the features. Features that are correlated with the pH are going to be important for our model if a significant correlation occurs. Correlation analysis is used to investigate the dependence between multiple variables at the same time. Even though the correlation assesses linear dependence it is a useful method to discover redundancy in the feature set [5]. The Correlation coefficient is a value that lies between -1 and 1 and indicates how strong the relationship is between variables. The interpretations of the values are;

- -1: Perfect negative correlation, a relationship does exist between the variables and it is negative relation (i.e., variables tend to move in the opposite direction)
- 0: No correlation, which indicates that there is no linear relation between the variables
- 1: Perfect positive correlation, a relationship does exist between the variables and it is positive relation (i.e., variables tend to move in the same direction)

The correlation coefficient could be calculated using multiple methods, we used the sklearn library in python to help us compute the coefficient, which measures the Pearson correlation coefficient using the following formula:

$$\rho = \frac{cov(x_1, x_2)}{\sqrt{var(x_1)var(x_2)}}$$

Where x_1 and x_2 are the two variables in which we want to calculate their correlation coefficient, $\overline{x_1}$ and $\overline{x_2}$ are their means respectively and ρ is the Pearson correlation coefficient we are interested in.

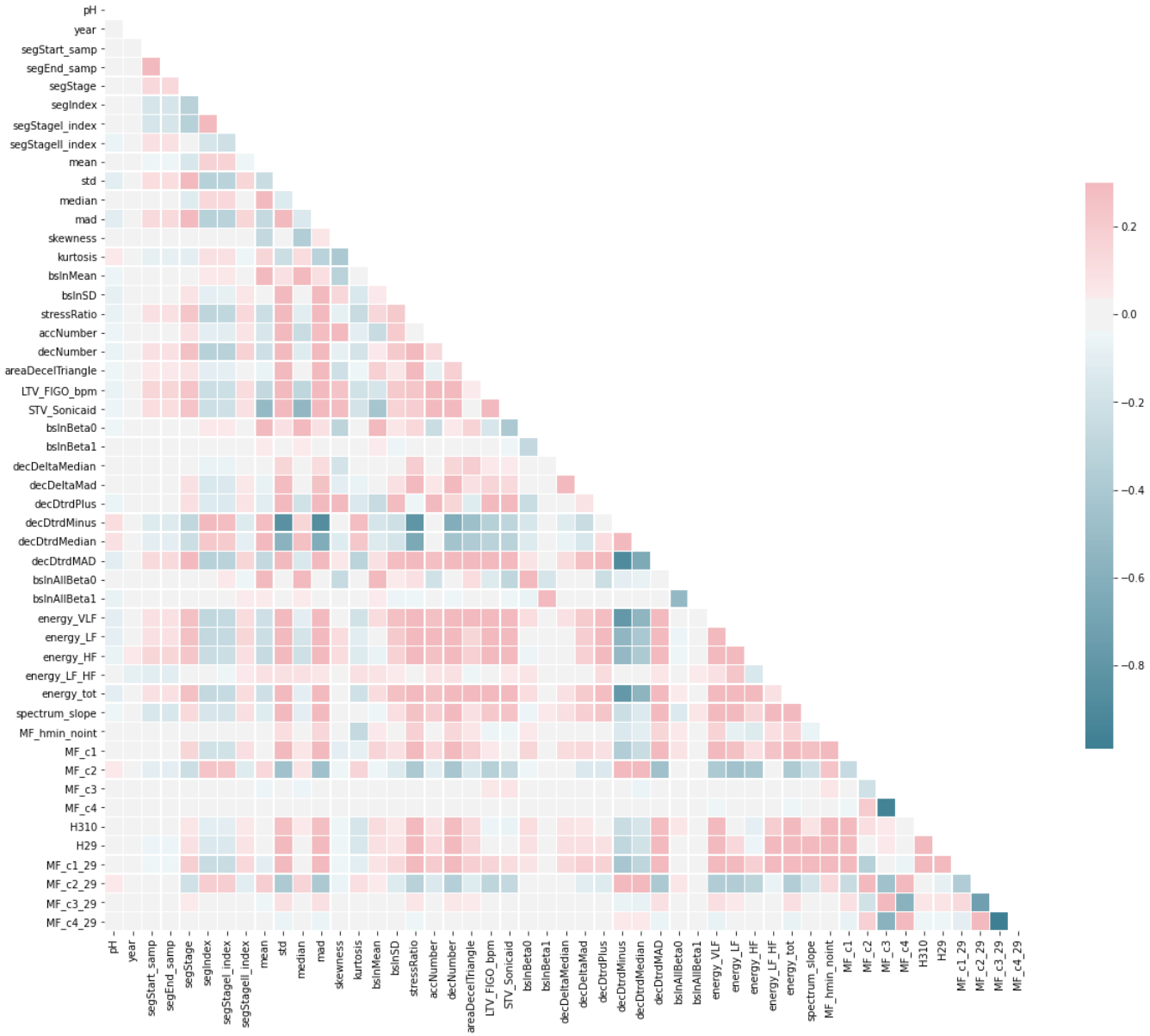


Figure6: Correlation matrix

Some features obtained from the dataset showed a weak correlation with the target vector as seen in Figure 6. Lastly, we inspect the correlation matrix for the features, large values in this matrix indicate serious collinearity between the variables involved, and such high values can make the predictions from a linear model slightly unreliable, we notice there are some slightly correlated features as expected as they come from the same domain. For Example, Figo-based features show some correlations among them similarly to frequency domain features.

IV. Feature importance and ranking

In clinical practice, We are not only interested in classifying patients accurately but also investigate the features that lead to that prediction, as we wonder which factors affect the patient's health the most and it helps us understand the dataset better by comparing features coming from different domains, this could be interpreted by a domain expert and could be used to lead the way of gathering more or different data, In addition to better understanding the dataset, it can help us understand the model better and can help with reducing the number of input variable which can lead to a less complicated model that can be generalized and works well for different datasets. In short, deciding on which features are important the most, we can improve the model by focusing on the important features, could be used in variable selection and finally the importance of the features could be intercepted and help us understand Acidosis better. In our case, two methods were used in determining the feature importance, firstly we measured the AUC score of each feature on its own, it is a simple yet effective technique in evaluating individual features. Information gain is another technique that is used to determine the importance of a feature. Random forest algorithm is one of the easy to use and implement, which is used for feature selection and classification problems providing two main methods, either Gini impurity or information gain entropy. The information gain evaluates an attribute by measuring the amount of information gain concerning a class. The mutual information, termed InfoGain, is computed using entropy H [5]:

$$InfoGain(Class, Attribute) = H(Class) - H(Class \mid Attribute)$$

Therefore random forest algorithm is capable of computing how much each feature decreases the weighted impurity in a tree and then averaged among multiple trees in the forest, which helps us rank the features according to this measure. The following, in Figure 7, are the results of fitting a logistic regression model with a single feature at a time to determine the AUC score for every feature and in Figure 8, we calculated the rank of each feature concerning its information gain entropy:

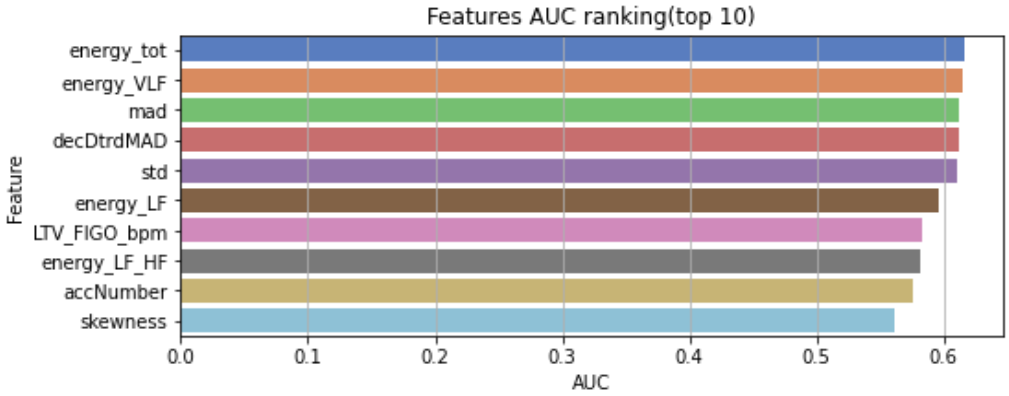


Figure7: AUC ranking for features

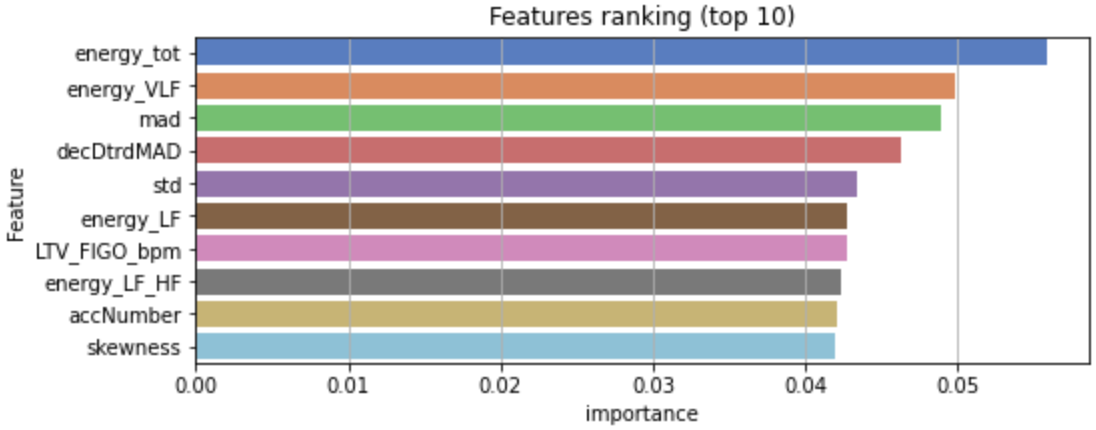
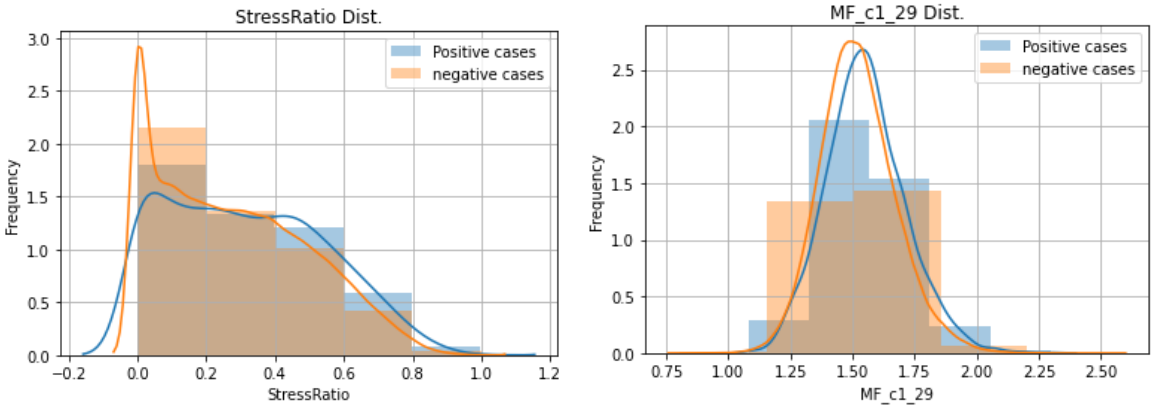


Figure8: Feature ranking using RF

As seen from the above results, both techniques lead to the same ranking, showing the top 10 ranked features, having the energy_tot feature at the top of rank with an AUC score of 0.61, also it is noticeable that the top 10 ranked features are coming for multiple domain which might imply that depending on one domain for classification might not be sufficient.Lastly, we show the distribution of the continuous features that were ranked high as follows, to help us better visualise and understand the data, as seen from Figure 9 below:



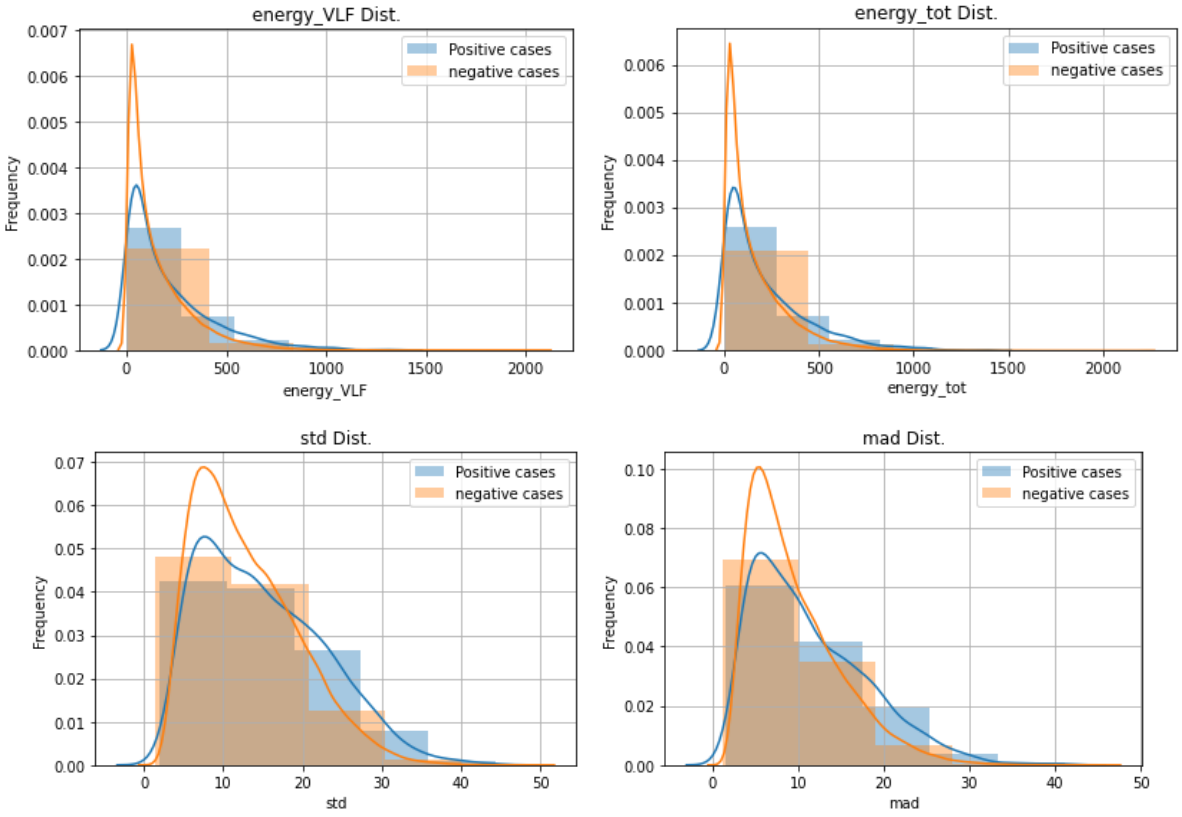


Figure9: Distribution of highly ranked features

Methods of classification

I. Statistical models

In this paper, we are going to discuss the performance of three popular algorithms used for classification problems which are: Logistic Regression, Support Vector Machine, and K-Nearest Neighbor. A subset of the dataset is used to train the data, only records of the first stage were taken into account and we also dropped redundant features (correlated or irrelevant [7]) leaving only 26 features, taking into account only the last segment.

Logistic Regression

Logistic regression is used to obtain the odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial. The result is the impact of each variable on the odds ratio of the observed event of interest[8]. Which is computed as:

$$logit\,(p\,(Y\,=1|x))\;=\;w_o x_o + w_1 x_1 + ... + w_m x_m \;=\; \sum_{i=0}^{m} w_i x_i = w^T x$$

Where $(p\,(Y\,=1|x)$ is the conditional probability of class variable Y for the feature x and the linear combination of weights and sample features $w^T x$ is squished by the sigmoid function

$$\vartheta(Z) = \frac{1}{1 + e^{Z}}$$

Here, z is the linear combination of weights and features, $Z = w_o x_o + w_1 x_1 + ... + w_m x_m$ and w_o is called bais, which is an additional input we provide together with x0 that is a vector of ones. Then, to fit the parameters to the model, we define the sum squared-error cost function as follows:

$$J(w) = \sum_i \frac{1}{2}(\vartheta(z^{(i)}) - y^{(i)})^2$$

We minimize the previous cost function by maximizing the natural log of the likelihood of the function, and rewriting it as follows:

$$J(w) = \sum_{i=1}^{n}[-y^{(i)} log(\vartheta(z^{(i)})) - (1 - y^{(i)}) log(1 - \vartheta(z^{(i)}))]$$

The natural log is used to reduce the potential for numerical underflow and it is easier to compute in practice, and now maximizing the likelihood using *gradient descent* will minimize the cost function $J(w)$.

Support Vector Machine

The Support Vector Machine model is an algorithm used for analyzing data that is usually for regression analysis and classification. The model is also used to classify non-linear sets of data and relationships. This is usually done using kernel trick, whereby inputs are mapped into high dimensional spaces. The model further develops a hyperplane that helps separate the data in a higher dimension. Moreover, the element of separation is usually developed through the distance between the hyperplane and the training data points. This idea is derived from the concept that when the margin is large, the generalization error becomes lower. The model can also be used for multidimensional classification of vectors. In this process, the first step is to develop the given into a vector. This further allows the operator to develop features that they can use for the classification process. To find the optimal hyperplane we have to solve the primal optimization task by minimizing the following equation:

$$(w, b) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i \quad [5]$$

Subjected to

$$< w.\vartheta(x_i) > + b \geq +1 - \xi, \quad y_i = +1$$
$$< w.\vartheta(x_i) > + b \leq -1 + \xi, \quad y_i = -1$$

where ξi are called slack variables that allow the margin constraints to be violated and $\vartheta(\cdot)$ is a kernel providing nonlinear feature mapping. Constant C is a trade-off between maximization of margin and minimization of error[9].

K-nearest Neighbour

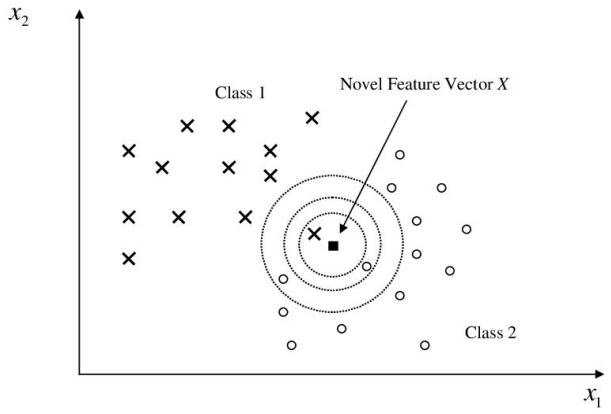


Figure10: Classifying using kNN

This is a model used to map inputs into outputs for regression and classification. The value K is a constant, which represents the nearest training samples. On the same note, the output is usually dependent on the purpose of the model. The output for classification is a class membership. This means that objects are allocated classes that have the most common features. Sometimes, the model weighs the average of the nearest neighbors to K, by finding its inverse. However, one thing to note is that the KNN classification model uses mainly the distance between points which could be the Euclidean, Manhattan, or Minkowski distance. In short, the KNN algorithm doesn't learn a discriminative function from the training data, but memorizes the training data instead and follows the following steps:

I. Choosing the number of K and the distance metric
II. Find the K-nearest neighbour for the sample we want to classify
III. Assign the class label by majority of voting

Evaluation metrics and performance estimation

A successful classification model should be able to generalize its prediction and perform well on unseen data, for that purpose a proper evaluation metric should be implemented to help assess the model. There are multiple evaluation metrics, among the popular one is the accuracy of the model, which is simply the ratio between the correct classified samples over the total number of samples, although it might look intuitive the accuracy metric fails to give a proper assessment in some of the cases, including our problem. In case of highly imbalanced data, using the accuracy metric could be misleading as the model can successfully predict the majority class while failing at predicting most of the minority class, this can lead to high accuracy but low performing model as in many cases, we are more interested in classifying the minority class correctly, and for that reason, other evaluation metrics should be considered.

The Confusion matrix is another popular evaluation matrix, that is more reliable in our case, The confusion matrix provides a more insightful picture which is not only the performance of a predictive model but also which classes are being predicted correctly.

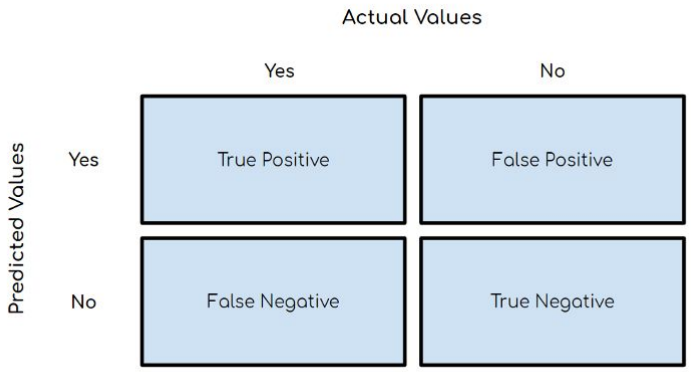


Figure11: Confusion matrix

Where:

- **True Positive (TP)**: The model correctly predicts the positive class
- **False Positive (FP)**: The model incorrectly predicts the positive class
- **True Negative (TN)**: The model correctly predicts the negative class
- **False Negative (FN)**: The model incorrectly predicts the negative class

We can extract insights from the confusion matrix like the sensitivity (also called the true positive rate) which measures the proportion of positive cases that are correctly identified, as well as the specificity (also called true negative rate) which measures the proportion of negative cases that are correctly identified. which can be calculated as follows:

$$\begin{aligned} Sensitivity\ (SE) &= TP/(TP+FN) \\ Specificity\ (SP) &= TN/(TN+FP) \\ gmean &= \sqrt{SE\ .\ SP} \end{aligned}$$

The receiver operation characteristic is another classification metric that is widely used to assess the performance of the model, which helps show the tradeoff between the sensitivity and specificity as the model's discrimination threshold is varied.

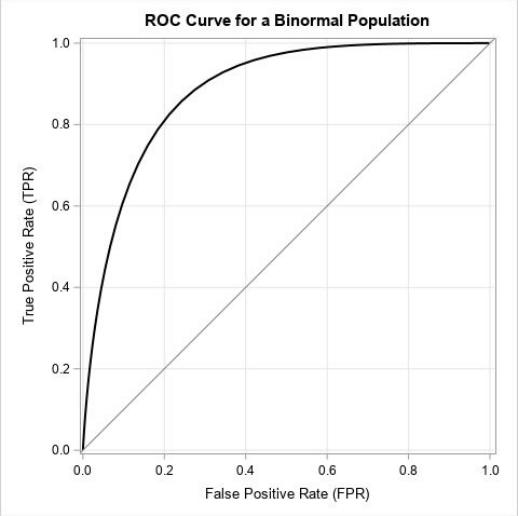


Figure12: ROC curve

The ROC curve shown in Figure 12 is graphical plot of the true positive rate plotted as a function of false positive rate, any successful model is placed above the diagonal line(random guess) and the area under the curve, referred as the AUC, which expresses probability that classifier rank randomly chosen positive instance higher than randomly chosen negative instance [5]. We chose the AUC metric to evaluate our models' performance.

As for the means of model validation, we used nested cross-validation with five folds for both the outer and inner layers. This choice was preferred over the train test split as in the latter option a portion of the data is not used for training the model. Nested cross-validation was also preferred on the single-layer cross-validation, as it overcomes the problems that occur with single layer CV like data leakage as it estimates the error of the model with the same data used to tune it in the first place which increases the bias of the model. It is implemented as follows:

Algorithm 1: K-Fold Nested Cross-Validation with Random Search

Require: K_1, K_2, where K_1 is number of outer folds and K_2 inner folds
Require: $\mathcal{D}$, dataset containing input features X and output feature y
Require: P_{sets}, set of hyperparameters with different values
Require: $\mathcal{M}$, a single estimator, model.

```
for i = 1 to K_1 splits do
    Split D into D_i^train, D_i^test for the i'th split
    for j = 1 to K_2 splits do
        Split D_i^train into D_j^train, D_j^test for the j'th split
        foreach p in RandomSample(P_sets) do
            Train M on D_j^train with hyperparameter set p
            Compute test error E_j^test for M with D_j^test
    Select optimal hyperparameter set p* from P_sets, where E_j^test is best
    Train M with D_i^train, using p*
    Compute test error E_i^test for M with D_i^test
```

For our logistic regression model, the inverse regularization parameter C is tuned from the range of 10^{-12} to 1, as for the kNN model, two hyperparameter were tuned, firstly the number of neighbours K which was ranging from 200 to 400, secondly was the weight function which was either uniform(all points in each neighborhood are weighted equally) or distance weighted (weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away). Lastly our SVM had only one hyperparameter to tune which was the kernel used, either linear or polynomial kernel.

Models Results and hypotheses testing

I. Results

We fitted the previously mentioned models, a logistic regression, support vector machine, and k-nearest neighbor, using python 3.7 library sklearn version 0.22.2, while having the AUC as our performance metrics while documenting the confusion matrix results as well as the Gmean.

the following are the results obtained from the three models :

Classifier	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
Logistic Reg.	0.62	1618	46	40	2758	0.53	0.63	0.57
KNN	0.57	0	0	86	4376	0	1	0
SVM	0.58	1347	37	49	3029	0.43	0.69	0.54

Table 2: Results from fitting LR, kNN and SVM

We notice that kNN performed appallingly, without predicting a single positive case. If we examine closely, it appears as kNN outputs the negative class only. The existing kNN algorithm is equivalent to using only local prior probabilities to predict instance labels, and hence it does not take into account the class distribution around the neighborhood of the query instance, which results in undesirable performance on imbalanced data [10]. Logistic regression outperforms SVM with an AUC of 0.62 with much higher sensitivity.

II. Proposed hypotheses

In this thesis, we examine several hypotheses and investigate whether it improves the performance of a statistical model. The experiments that test those hypotheses are done independently and compared to a baseline estimate, in our case, this estimate is the results from the previous logistic regression mentioned in Table2. All hypotheses are tested using a logistic regression model.

Addition of the year Feature

The dataset collected by Brno university hospital over the span of multiple years, from 2010 until 2017. Our first proposition is including the year feature into the model and checking whether it helps the model, as seen from the following:

Year Incl.	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
No.	0.62	1618	46	40	2758	0.53	0.63	0.57
Yes	0.62	1202	33	53	3174	0.38	0.73	0.44

Table3: Results from fitting LR with the feature 'year' included, and compared with our base estimation

the year feature, although it didn't change the AUC score, but made a noticeable trade-off by increasing the model's specificity for decreasing its sensitivity. This could be undesirable as we care more for classifying the minority class correctly as well as the decrease of the gmean which indicates that the year feature doesn't contribute to the classification.

Features included in the model

As mentioned before, only 26 features out of 49 were used to train the model, and highly correlated features were dropped. However, our second proposition is to investigate the performance of a model trained on all features .

Features inc	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
Subset.	0.62	1618	46	40	2758	0.53	0.63	0.57
All	0.68	1244	51	35	3132	0.59	0.72	0.65

Table4: Results from fitting LR with all features included, and compared with our base estimation

The above table shows that including all features highly benefits the model increasing its performance by 5% and was able to correctly classify 51 positive cases, which is a huge improvement to the model. We decided to measure the weights of the all the features (including the ones that were recommended to be dropped) and compare the highly ranked features to the one obtained before previously in Figure 7 and the weights obtained from fighting the logistic regression to the recommended subset of features, the following are the results:

Recommended Subset of Features

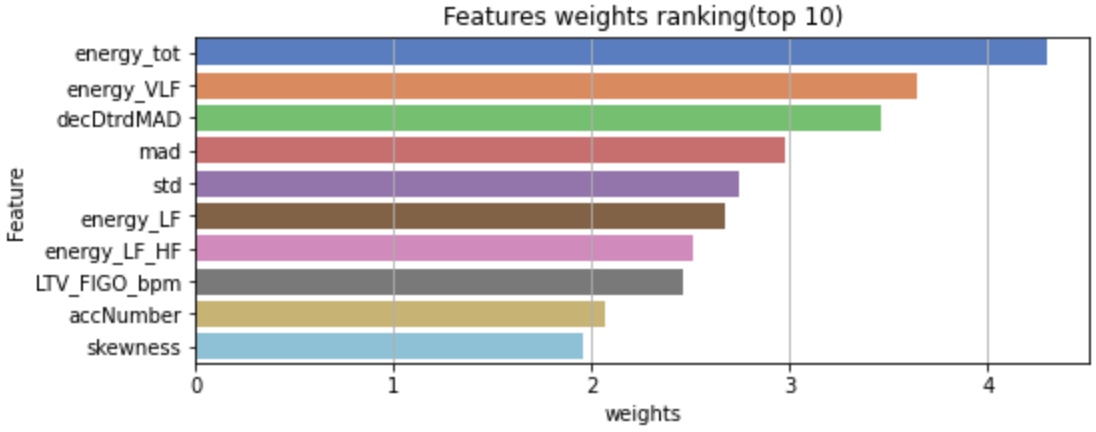


Figure 13: Features weights(LR is fitted for the recommended subset) Weights are multiplied by 10^{10}

All Features

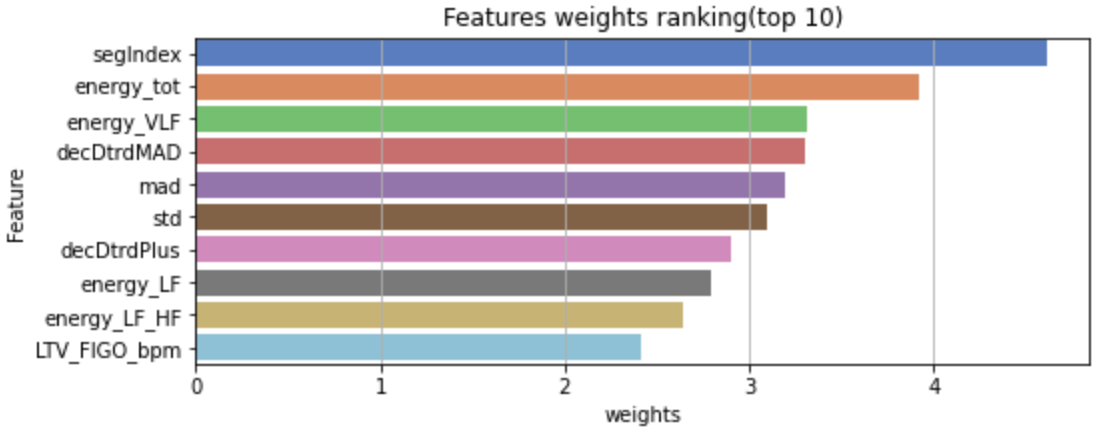


Figure 14: Features weights(LR is fitted for the all features) Weights are multiplied by 10^{10}

SegIndex which was initially dropped has the highest weight as shown in Figure 14 , followed by the energy_VLF and energy_tot (similarly to the results in Figure 13). Both Figures show similar results but some dropped features, segIndex and decDtrdMAD had high weights, which might indicate that increasing the features space and including more features can help with the performance.

Number of segments

We mentioned that we are taking into account only the first stage and the last segment of each patient, so thirdly we investigate the model's behavior if trained on multiple segments. We included data computed from more segments 2, 4, 6, and 8 to the dataset and trained the model again. Note that when using more segments, we did not separate test subjects in cross-validation. It can happen that segments from the same subject are in the training set and in the testing set. This leads to an optimistic bias in performance estimation. However, for the sake of simplicity, we decided not to neglect that in the current analysis.

Segment inc	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
1.	0.62	1618	46	40	2758	0.53	0.63	0.57
2	0.64	2723	87	84	6002	0.51	0.69	0.59
4	0.65	5303	175	165	11997	0.51	0.69	0.60
6	0.63	8350	265	234	17256	0.53	0.67	0.60
8	0.63	11677	343	296	21596	0.54	0.65	0.59

Table 5: Results from fitting LR with multiple number of segments, and compared with our base estimation

In Table 5 we can see an increase in AUC value and G-mean when more segments are included. For the four segments, the AUC is reaching the highest value. However, we would expect that increasing the number of segments would lead to significant improvement in performance, considering that more data are used for learning and, even more, that we are neglecting the optimistic bias introduced in evaluation.

Principal Component Analysis

Feature extraction methods are algorithms that help reduce the feature space while keeping relevant and non-redundant information. Unlike feature selection which keeps original features, feature extraction algorithms are used to project the data into new feature space.PCA is one of the most popular feature extraction algorithms, mentioned excessively in the literature.

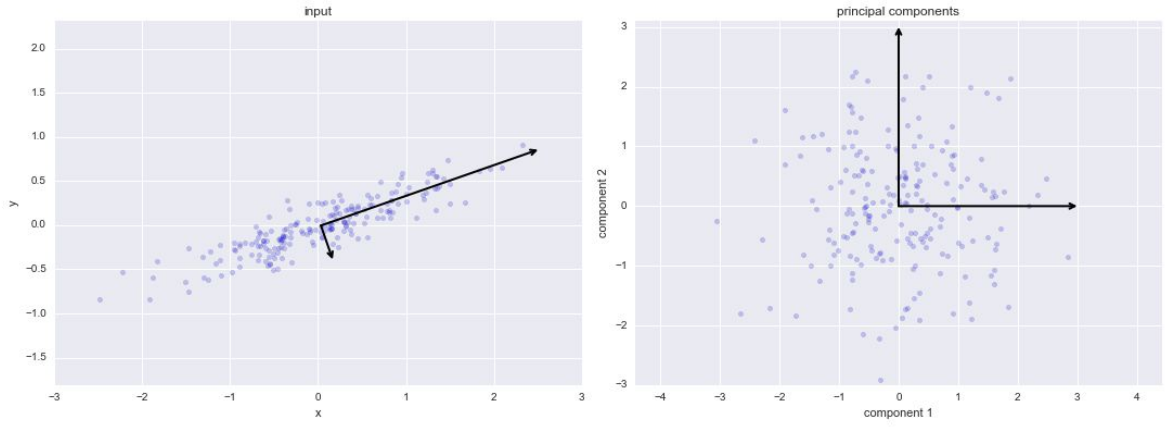


Figure 15: Example of Principal Component Analysis

The covariance between two features is defined by the following:

$$\sigma_{jk} = \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

Where μ_j and μ_k are sample means of feature j and k respectively, then we form the covariance matrix of the features and obtain the eigenpairs of the matrix as the eigenvectors of the covariance matrix represent the principle component in which

$$\Sigma v = \lambda v$$

Where is the covariance matrix, and v represent the eigenvector with an eigenvalue, we then can reduce the dimension of the feature space by choosing only a subset of the eigenvector that contains most information(i.e., high variance, large eigenvalue).

Our fourth proposition is implementing PCA into our model, which can help reduce the feature space and get rid of the problem accompanied by having high dimensions(the curse of dimensionality).

PCA applied	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
No.	0.62	1618	46	40	2758	0.53	0.63	0.57
Yes	0.56	1160	31	55	3216	0.36	0.73	0.43

Table 6: Results from fitting LR with PCA applied, and compared with our baseline performance.

As seen from the results of fitting PCA, the model's performance got worse, it appears that(also from our second experiment) the model behaves better while having more features. There are multiple reasons why PCA might decrease the performance of a classifier. PCA is agnostic to the target Y, this means that even while extracting a new axis that explains most of the Variance from two given features, we could be getting rid of vital information for our target, and since we can not include the target in the PCA analysis(due to data leakage), it might happen that PCA decreases the model's performance. In other words, the first principal component is the direction in space along which projections have the largest variance, but that doesn't mean it is optimal for classifying our target, as the direction for instance could be horizontal but classes are separable vertically.

Synthetic Minority Oversampling Technique

(SMOTE) is a popular technique to compensate for an imbalance in data. It operates on the minority class creating artificial data. SMOTE is based on real data belonging to the minority

class and it operates in the feature space rather than the data space [5]. The minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement. This approach is inspired by a technique that proved to be successful [6, 10]The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors [6, 10]. As it is clear, our data sufferers from high imbalance rate between the two target classes, so our final proposal is applying SMOTE and compares the results as follows:

SMOTE	*AUC*	*FP*	*TP*	*FN*	*TN*	*SE*	*SP*	*Gmean*
No.	0.62	1618	46	40	2758	0.53	0.63	0.57
Yes	0.61	1276	35	51	3100	0.42	0.71	0.44

Table 7: Results from fitting LR with SMOTE analysis applied, and compared with our base estimation

However, in our case, the application of SMOTE didn't improve the results of the model. This might happen due to the fact that the logistic regression already had weights associated with classes and automatically adjusts the weights inversely proportional to class frequencies in the input data, so even if previously(before applying SMOTE) the majority class was dominating, the model was still heavily penalized for every wrong classification it makes for the minority class.

Conclusion and discussion

In the thesis we discussed the process of classifying intrapartum fetal heart rate to successfully detect fetal acidosis, using features obtained from FHR signals. We discussed the importance of machine learning in the medical field and discussed arguments for its use for fetal heart rate classification. We explored the intrapartum CTG dataset provided by the university hospital Brno and briefly described the process behind collecting the data using STAN devices and features extracted from the FHR signals. We used features from multiple domains ranging from typical clinical features to features based on scale invariance measures. We analyzed the features obtained, and showed their importance toward the target (pH value obtained after delivery) and concluded that energy_tot is the most relevant feature with AUC score of 0.61 (estimated by logistic regression) followed by energy_VLF and maximum absolute deviation MAD, both scoring 0.60. Also, we plotted the distribution of the highly ranked (AUC wise) features for further analysis. Further, we investigated the feature correlation matrix. We showed the high

imbalance found in the data as only 86 samples were positive (only 0.02%) which was one of the reasons that made us discard the accuracy as an evaluation matrix. As a result, we specified another evaluation matrix(AUC) and explained why we think it is better suited for classification, (beside that SE, SP, and G-mean were also reported as complementary measures). We then proceeded by explaining the workflow behind the nested cross-validation which was used to assess our models and explained why we favored it over the single loop cross-validation. We described three popular classification algorithms, logistic regression, SVM and kNN. We trained them on the data and showed the obtained results and analyzed the results (the three models scored 0.62, 0.57 and 0.58, respectively) and interpreted the models. We proposed multiple hypotheses and further investigated their effect on the performance of our model. We implemented PCA to our model and documented the results (the AUC decreased to 0.56). In addition to that, we tried oversampling the minority class using SMOTE, and finally added extra features to check the relevance and concluded that the year feature might not be relevant but including more features might be beneficial as it increased the AUC score to 0.68, which made us investigate other features that were dropped by measuring their AUC and found that some of them might be beneficial for our model, as three initially dropped features which are segIndex, decDtrdPlus and bslnAllBeta1 scored relatively high AUC scores of 0.63, 0.60 and 0.58 respectively.

References (MLA8 style)

[1] "Ascent of machine learning in medicine." Nature materials vol. 18,5 (2019): 407. doi:10.1038/s41563-019-0360-1

[2] Georgieva, Antoniya, et al. "Computer‐Based Intrapartum Fetal Monitoring and beyond: A Review of the 2nd Workshop on Signal Processing and Monitoring in Labor (October 2017, Oxford, UK)." Acta Obstetricia Et Gynecologica Scandinavica, vol. 98, no. 9, 2019, pp. 1207–1217., doi:10.1111/aogs.13639.

[3] Bobrow, C. S, and P. W Soothill. "Causes and Consequences of Fetal Acidosis." *Archives of Disease in Childhood - Fetal and Neonatal Edition*, vol. 80, no. 3, 1999, doi:10.1136/fn.80.3.f246.

[4] Georgoulas, George, et al. "Investigating PH Based Evaluation of Fetal Heart Rate (FHR) Recordings." *Health and Technology*, vol. 7, no. 2-3, 2017, pp. 241–254., doi:10.1007/s12553-017-0201-7.

[5] Spilka, J.: Complex approach to fetal heart rate analysis: a hierarchical classification model. Ph.D. Thesis, Czech Technical University in Prague Department of Cybernetics (2013)

[6] Abry, P, et al. "Sparse Learning for Intrapartum Fetal Heart Rate Analysis." *Biomedical Physics & Engineering Express*, vol. 4, no. 3, 2018, p. 034002., doi:10.1088/2057-1976/aabc64.

[7] Spilka, Jiri, et al. "Sparse Support Vector Machine for Intrapartum Fetal Heart Rate Classification." *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, 2017, pp. 664–671., doi:10.1109/jbhi.2016.2546312.

[8] Sperandei, Sandro. "Understanding Logistic Regression Analysis." *Biochemia Medica*, 2014, pp. 12–18., doi:10.11613/bm.2014.003.

[9] Vapnik, Vladimir Naumovich. *The Nature of Statistical Learning Theory*. Springer, 2010.

[10] Dubey H., Pudi V. (2013) *Class Based Weighted K-Nearest Neighbor over Imbalance Dataset.* In: Pei J., Tseng V.S., Cao L., Motoda H., Xu G. (eds) *Advances in Knowledge Discovery and Data Mining*. PAKDD 2013. Lecture Notes in Computer Science, vol 7819. Springer, Berlin, Heidelberg.