Classification of Fetal State and Morphologic pattern using Machine Learning Techniques

Tanmey Rawal

December 20, 2020

1 Abstract

A major contributor to under-five mortality is the death of children in the 1st month of life. Intrapartum complications are one of the major causes of perinatal mortality. The objective of this study was to apply various machine learning techniques on Cardiotocogram data set helping doctors in identifying high-risk fetuses and prevent child and maternal mortality.

2 Introduction

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 mortality rate for same age group to at least as low as 25 per 1,000 live births.

Parallel to notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented. There is a growing tendency to use clinical decision support systems in medical diagnosis. These systems help to optimize medical decisions, improve medical treatments, and reduce financial costs

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. Cardiotocograms is a recording of two distinct signals, fetal heart rate, and uterine activity. It is used for determining the fetal state during both pregnancy and delivery. The aim of the CTGs monitoring is to determine babies who may be short of oxygen (hypoxic); thus further assessments of fetal condition may be performed or the baby might be delivered by Caesarean section or natural birth. The visual evaluation of the CTGs not only requires time but also depends on the knowledge and clinical experience of obstetricians.

2.1 Problem Statement

As per the literature of machine learning, it is a classification task.

- Primary Task: Classify fetal health state into 3 different classes.
- Secondary Task : Classify FHR (fetal heart rate) pattern into 10 classes.

2.2 Objective

The objective of this study was to apply various machine learning classification techniques like SVM, decision tree, etc. on Cardiotocogram data set helping doctors in identifying high-risk fetuses, and compare their precision, recall and other parameters. The dataset contains two different kinds of labels FHR pattern (10 classes) and fetal heath state (3 classes), but our primary focus is to create models which would classify the fetal health state into 3 different classes (i.e normal,suspect, pathological) and secondary task is to create models which would classify FHR pattern into 10 different classes(A,B,C,....)

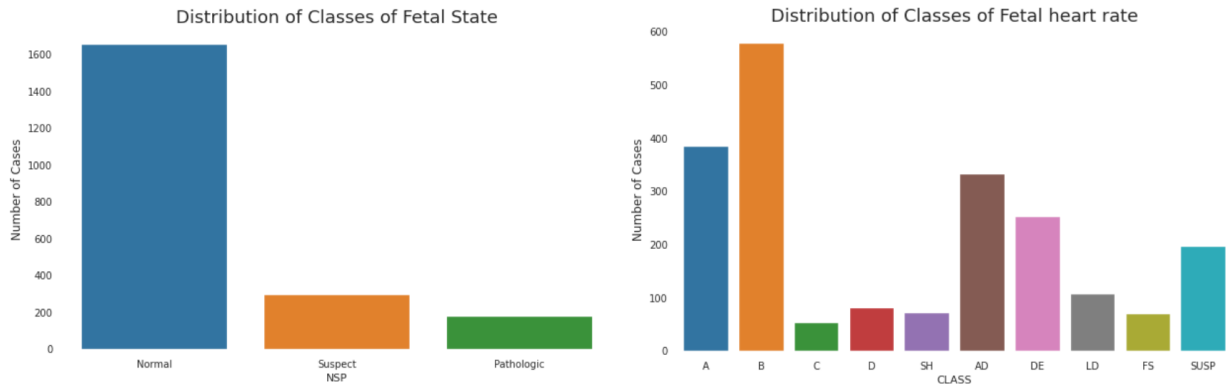
2.3 Cardiotocography Dataset

The Cardiotocography dataset is taken from UCI machine learning repository. 2126 fetal cardiotocograms (CTGs) were automatically processed and the respective diagnostic features measured. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a FHR pattern (A, B, C. ...) and to a fetal state (N, S, P). Therefore the dataset can be used either for 10-class or 3-class experiments.

The original dataset contains 23 attributes, out of which 2 are target variables, the remaining 21 attributes are numerical.

Note to reader: brief description of all attributes and all the different possible states target variable can have are briefly described in appendix, check it out if interested.

3 Exploratory Data Analysis



(a) Distribution of Classes of Fetal State

(b) Distribution of Classes of FHR

Figure 1: Number of cases in each class of fetal state and Fetal heart rate(FHR) respectively.

We can clearly see from figure 1a in fetal state class, most of the cases have normal fetal state condition followed by suspect fetal state condition and then pathologic fetal state condition.

From figure 1b in fetal heart rate class, most of the cases have state ”B”, followed by state ”A”. We have least number of cases of state ”C”.

The states of classes of fetal heart rate classes are briefly described in Table 5.

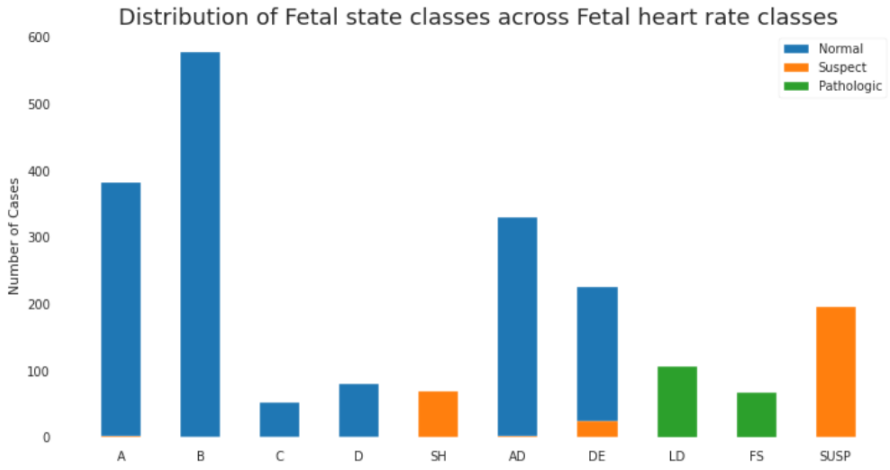


Figure 2: Distribution of Fetal state classes across Fetal heart rate classes

From figure 2 we can clearly see for fetal heart rate classes: ”B”, ”C”, ”D” has normal fetal state. ”A”, ”AD”, ”DE” classes primarily consist of normal fetal state and very few suspect fetal state. Cases having ”LD” FHR has pathologic fetal state. Cases having ”FS” FHR has primarily has pathologic fetal state and very few cases of suspect fetal state. Cases having ”SUSP” FHR has primarily has suspect fetal state and very few cases of pathologic fetal state.

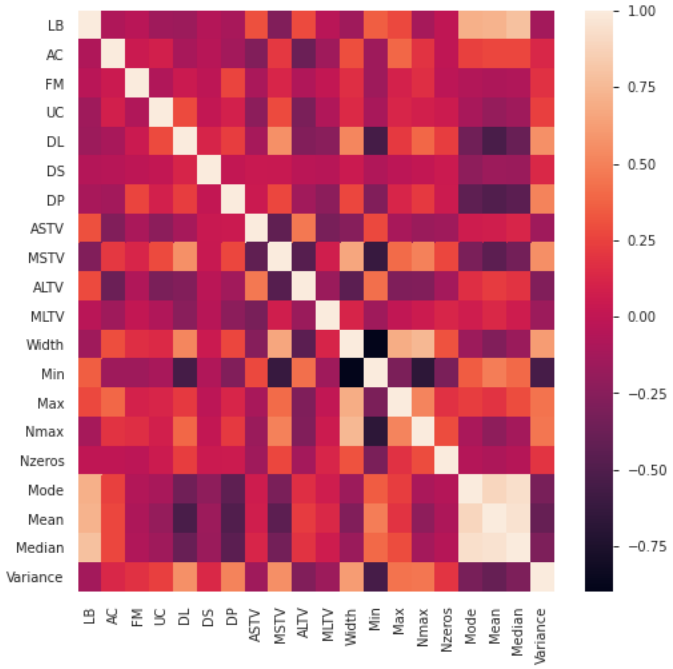
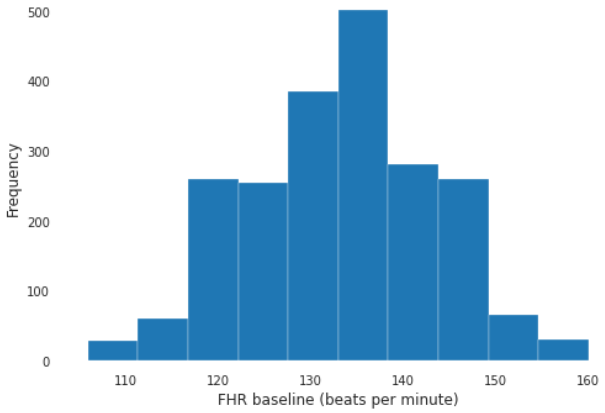
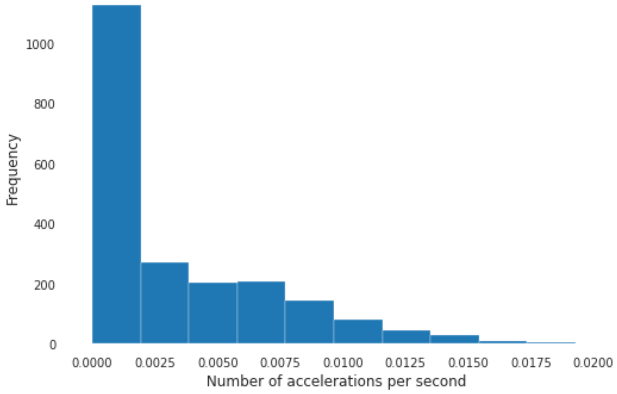


Figure 3: Heatmap of numerical variables

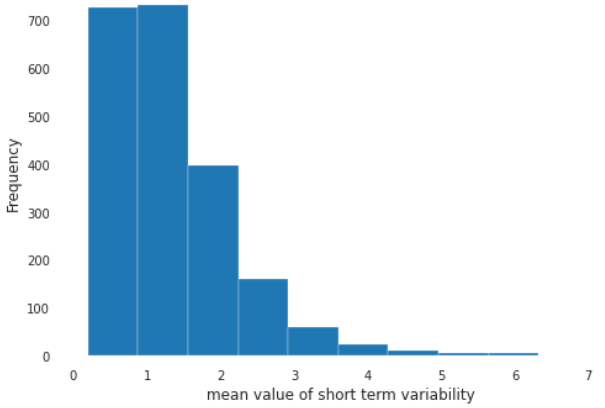
From figure 3 we can see the correlations between the variables. LB is highly correlated with mean,mode and median of histogram. Width is highly correlated with min, nmax, max of histograms.



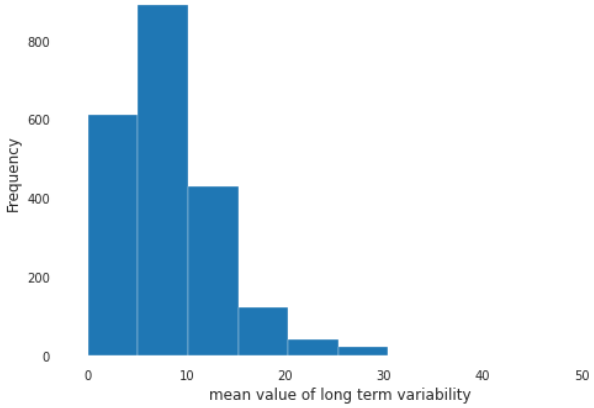
(a) Histogram of FHR baseline (beats per minute)



(b) Histogram of accelerations per second



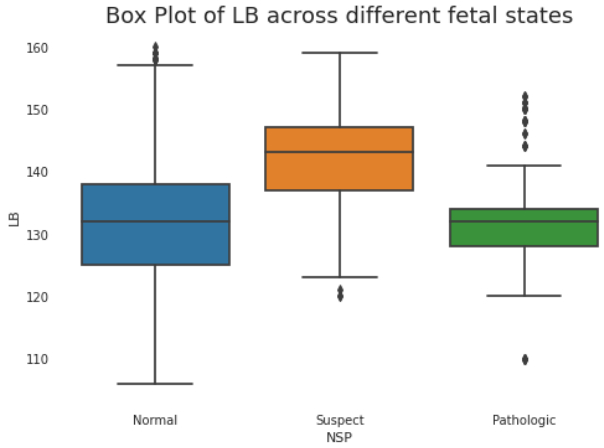
(c) Histogram of mean value of short term variability



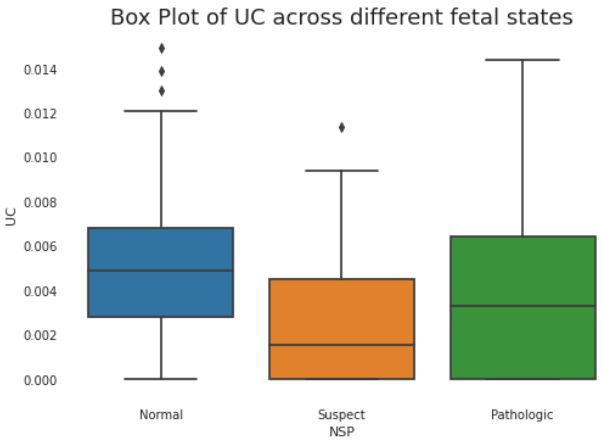
(d) Histogram of mean value of long term variability

Figure 4: Histogram of different features of Cardiotocogram data set

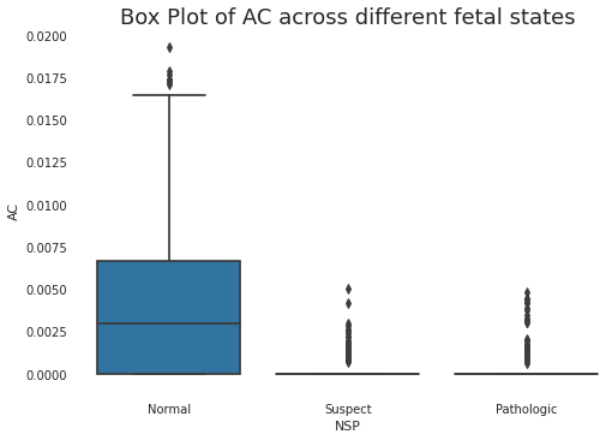
From figure 4a we can see FHR baseline is almost symmetrical with centre in interval in (130,140). From figure 4b we can see number of accelerations per second is negatively skewed and we have majority of values in range (0.0000-0.0025). From figure 4c and 4d we can see the distribution of mean value of short term variability and mean value of long term variability respectively, this data is also negatively skewed, for short term variability we have most of the values in range(0-2) and for long term variability we have maximum values in range(5-10).



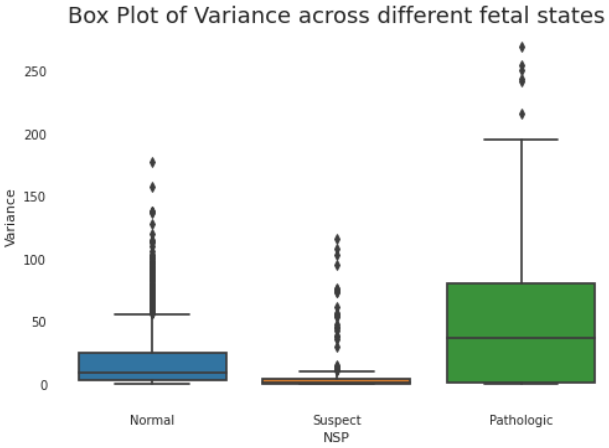
(a) Box Plot of FHR baseline (beats per minute) across different fetal states



(b) Box Plot of uterine contractions across different fetal states



(c) Box Plot of Number of accelerations per minute across different fetal states



(d) Box Plot of Variance across different fetal states

Figure 5: Box Plot of features across different fetal states

We had plotted boxplot for all features, but we have shown only few boxplots which show us some interesting facts about data. From figure 5a, we can see suspect fetal states has median more than both normal and pathologic fetal state and patholic has less spread as compared to other states. From figure 5b we can see median of uterine contractions is minimum as compared to other states. From 5c we can see normal cases have more number of acceleration per minute as compared to other states. .From 5d we can see suspect cases have least variance in CTG histogram and pathologic cases has largest variance in CTG histogram.

4 Machine Learning algorithms

4.1 Dummy Classifier

I have created a dummy classifier which would give majority class as output as for both fetal state and fetal heart rate. This would act as my benchmark model and performance of each classifier is evaluated against this benchmark model.

4.2 Logistic Regression

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring that they sum to one and remain in [0, 1]. The model has the form

$$log\frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$

$$log\frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$

...

$$log\frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

The model is specified in terms of K-1 log-odds or logit transformations (reflecting the constraint that the probabilities sum to one)

4.3 Decision Tree

Decision Trees are intuitive, and their decisions are easy to interpret. Such models are often called white box models. It contains series of splitting rules, starting at the top of tree and the prediction is the majority class of the node We take greedy approach to decide which variable would be used for splitting. We consider all predictors $X_1 \ldots, X_p$, and all possible values of the cut-off point s for each of the predictors, and then choose the predictor and cut-off point such that cost function is minimised. Scikit-Learn library in python uses the Classification and Regression Tree (CART) algorithm to train Decision Trees. Once the CART algorithm has successfully split the training set in two, it splits the subsets using the same logic, then the sub-subsets, and so on, recursively. It stops recursing once it satisfies stopping criteria. CART classification cost function is :

$$J = \frac{m_{left}}{m}G_{left} + \frac{m_{right}}{m}G_{right}$$

where m is the total number of data points in dataset.

$m_{left/right}$ is the total number of data points in left/right node.

$G_{left/right}$ measures the impurity of the left/right node.

4.4 K Nearest Neighbour

K-nearest neighbor (kNN) is a lazy learning method in the sense that no model is learned from the training data. Learning only occurs when a test example needs to be classified. Given a query point x_0, we find the k training points x_r, r = 1, . . . , k closest in distance to x_0, and then classify using majority vote among the k neighbors.

4.5 Linear and Quadratic Discriminant Analysis

We need to know the class posteriors for optimal classification. Suppose $f_k(x)$ is the class-conditional density of X in class G = k, and let π_k be the prior probability of class k, with $\sum_{k=1}^{K} \pi_k = 1$. Using Bayes Theorem we

$$P(G=k|X=x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

We see that in terms of ability to classify, having the $f_k(x)$ is almost equivalent to having the quantity Pr(G = k | X=x). Linear and Quadratic discriminant analysis use Gaussian densities to model $f_k(x)$. Linear discriminant analysis (LDA) arises in the special case when we assume that the classes have a common covariance matrix $\Sigma_k = \Sigma \forall k$. This causes the normalization factors to cancel, as well as the quadratic part in the exponents.The decision boundary between two classes are linear. Linear discriminant functions:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k$$

Best decision rule

$$G(x) = argmax_k \delta_k(x)$$

We can estimate the gaussian parameters from MLE method using our training data.

For Quadratic discriminant analysis there is atleast one k for which $\Sigma_k \neq \Sigma$. Then the convenient cancellations in do not occur; in particular the pieces quadratic in x remain Quadratic discriminant functions

$$\delta_k(x) = -\frac{1}{2}|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log \pi_k$$

The decision boundary between each pair of classes k and l is described by quadratic equation $\{x: \delta_k(x)=\delta_l(x)\}$.

The estimates for QDA are similar to those for LDA, except that separate covariance matrices must be estimated for each class. When number of features are large this can mean a dramatic increase in parameters.

4.6 Support Vector Machine

A Support Vector Machine (SVM) is a powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification. In its most simple type SVM are applied on binary classification, dividing data points either in 1 or 0. For multiclass classification, the same principle is utilized. The multiclass problem is broken down to multiple binary classification cases, which is also called one-vs-one. In scikit-learn one-vs-one is not default and needs to be selected explicitly. One-vs-rest is set as default. It basically divides the data points in class x and rest. Consecutively a certain class is distinguished from all other classes. The following formula poses the optimization problem that is tackled by SVMs.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

subject to

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i$$

where w is the normal vector to hyperplane, b is the bias or offset scalar, ξ_i are the slack parameters which are used to allow soft margins, C is the penalty parameter which controls the trade-off between minimizing the error and maximizing the margin, and $\phi(x_i)$ is a nonlinear mapping from the input space to the higher dimensional feature space. The corresponding dual problem is:

$$\max_{\alpha} J(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

subject to

$$\sum_{i=1}^{N}\alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq C \quad \forall i$$

where α_i are Lagrange multipliers, the term $K(x_i,x_j)$ is a kernel function representing the inner product of two vectors in the feature space, that is,$\phi^T(x_i)\phi(x_j)$ Kernel function must satisfy the well known Mercer's condition.

4.7 Passive Aggressive Classifier

Passive Aggressive Classifier is one of the few online-learning algorithms. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. Passive-Aggressive algorithms are somewhat similar to a Perceptron model, in the sense that they do not require a learning rate. However, they do include a regularization parameter.Passive-Aggressive algorithms are called so because :

- Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.
- Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

4.8 Random Forest

Random forest classifier is an ensemble algorithm. Let D is the data set comprising of N data points and p features. k bootstrap samples D_1, D_2,. . ., D_k are created from the dataset D. For each D_i, build a Decision tree T_i. We fix the number of attributes (say m), At each level, choose a random subset of available attributes of size m. Evaluate only these m attributes to choose next query. Final prediction is decided by the vote on the results returned by Combination of various such trees T_1, T_2, . . . , T_k is used to predict the class of a new data point by majority voting.

4.9 Gradient Boosting

Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor.This method tries to fit the new predictor to the residual errors made by the previous predictor. If we use Decision Trees as the base predictors then it is called Gradient Tree Boosting.We have used two Gradient Tree boosting Methods Light Gradient Boosting Machine(LGBM) and Extreme Gradient Boosting(XGBoost). LGBM is much faster than XGBoost. XGBoost generally produces better results than LGBM

4.9.1 LGBM

Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm.Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit.So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms.LightGBM uses a novel technique of Gradientbased One-Side Sampling (GOSS) to filter out the data instances for finding a split value.GOSS (Gradient Based One Side Sampling) is a novel sampling method which down samples the instances on the basis of gradients. As we know instances with small gradients are well trained (small training error) and those with large gradients are undertrained.A naive approach to downsample is to discard instances with small gradients by solely focussing on instances with large gradients but this would alter the data distribution. In a nutshell, GOSS retains instances with large gradients while performing random sampling on instances with small gradients.

4.9.2 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.XGBoost uses presorted algorithm & Histogram-based algorithm for computing the best split.In simple terms, Histogram-based algorithm splits all the data points for a feature into discrete bins and uses these bins to find the split value of the histogram. While it is efficient than the presorted algorithm in training speed which enumerates all possible split points on the presorted feature values.

5 Results

5.1 Result for Fetal State Classification

Model	N	S	P	Weighted F1 Score
XGboost	0.97	0.84	0.97	0.95
LGBM	0.97	0.85	0.95	0.95
Random Forest Classifier	0.97	0.81	0.97	0.94
Passive Aggressive Classifier	0.95	0.76	0.78	0.91
Support Vector Classifier	0.95	0.72	0.81	0.91
Quadratic Discriminant Analysis	0.92	0.69	0.73	0.88
Linear Discriminant Analysis	0.92	0.64	0.69	0.86
K Nearest Neighbours	0.96	0.74	0.91	0.92
Decision Tree	0.96	0.79	0.93	0.93
Logistic Regression	0.95	0.74	0.73	0.90
Dummy Classifier	0.88	0	0	0.69

Table 1: Comparison of F1 score of various machine learning models predicting fetal state on test data

5.2 Result for Fetal Heart Rate Classification

Model	A	B	C	D	SH	AD	DE	LD	FS	SUSP	Weighted F1 score
XGboost	0.88	0.92	0.71	0.93	0.54	0.89	0.92	0.97	0.90	0.88	0.89
LGBM	0.87	0.92	0.71	0.86	0.52	0.91	0.92	0.95	0.90	0.85	0.88
Random Forest Classifier	0.85	0.89	0.71	0.87	0.40	0.91	0.90	0.97	0.80	0.85	0.86
Passive Aggressive Classifier	0.69	0.86	0	0.70	0.63	0.81	0.79	0.74	0	0.60	0.73
Support Vector Classifier	0.81	0.89	0.62	0.80	0.65	0.90	0.89	0.92	0.74	0.69	0.84
Quadratic Discriminant Analysis	0.72	0.80	0.61	0.86	0.48	0.70	0.82	0.89	0.53	0.70	0.75
Linear Discriminant Analysis	0.67	0.88	0.62	0.71	0.52	0.77	0.76	0.85	0.29	0.62	0.75
K Nearest Neighbours	0.78	0.84	0.44	0.69	0.48	0.80	0.85	0.97	0.74	0.69	0.79
Decision Tree	0.75	0.84	0.46	0.71	0.39	0.87	0.83	0.86	0.76	0.74	0.79
Logistic Regression	0.78	0.89	0.62	0.77	0.65	0.91	0.89	0.89	0.50	0.65	0.82
Dummy Classifier	0	0.46	0	0	0	0	0	0	0	0	0.14

Table 2: Comparison of F1 score of various machine learning models predicting fetal heart rate on test data

5.3 Conclusion

We can clearly see using metric weighted average F1 score from Table 1 XGboost, LGBM, and Random Forest classifier are performing best while predicting fetal health state on test dataset. Similarly from Table 2 XGboost, LGBM, and Random Forest classifier are performing best while predicting fetal heart rate on test dataset. This just shows the power of ensemble models.

Appendix

Brief Description of dataset

S.No	Name of feature	Description	Type of variable
1	LB	Fetal heart rate (FHR) baseline (beats per minute)	Numerical
2	AC	No. of accelerations per second	Numerical
3	FM	No. of fetal movements per second	Numerical
4	UC	No. of uterine contractions per second	Numerical
5	DL	No, of light decelerations per second	Numerical
6	DS	No. of severe decelerations per second.	Numerical
7	DP	No. of prolongued decelerations per second	Numerical
8	ASTV	percentage of time with abnormal short term variability	Numerical
9	MSTV	mean value of short term variability	Numerical
10	ALTV	percentage of time with abnormal long term variability	Numerical
11	MLTV	mean value of long term variability	Numerical
12	Width	width of FHR histogram	Numerical
13	Min	minimum of FHR histogram	Numerical
14	Max	Maximum of FHR histogram	Numerical
15	Nmax	No. of histogram peaks	Numerical
16	Nzeros	No. of histogram zeros	Numerical
17	Mode	histogram mode	Numerical
18	Mean	histogram mean	Numerical
19	Median	histogram median	Numerical
20	Variance	histogram variance	Numerical
21	Tendency	histogram tendency (-1=left assymetric; 0=symmetric; 1=right assymetric)	Numerical
22	CLASS	FHR pattern class code (1 to 10) (Target variable for secondary task)	Categorical
23	NSP	fetal state class code (N=normal; S=suspect; P=pathologic) (Target variable for primary task)	Categorical

Table 3: Cardiotocography Dataset

The three classes of NSP :

Class No.	Symbol	Description
1	N	Normal
2	S	Suspect
3	P	Pathologic

Table 4: Classes of NSP

The ten classes of FHR pattern are :

Class No.	Symbol	Description
1	A	calm sleep
2	B	REM sleep
3	C	calm vigilance
4	D	active vigilance
5	SH	shift pattern
6	AD	accelerative/decelerative pattern (stress situation)
7	DE	decelerative pattern (vagal stimulation)
8	LD	largely decelerative pattern
9	FS	flat-sinusoidal pattern (pathological state)
10	SUSP	suspect pattern

Table 5: Classes of Morphologic pattern/ FHR pattern