Multiclass Classification of Fetal Health using Cardiotocogram Data

JR Ereyi (jrereyi@stanford.edu), Marc Huo (marchuo@stanford.edu)

1 Introduction

Due to the intricacies of the body of a fetus, as well as the rapid rate at which fetuses develop, fetal health care is one of the most difficult medical fields to practice effectively. Due to the challenges around fetal health care, reducing child and maternal mortality rates is of paramount importance to every modern country. The reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals. The UN expects that by 2030, member countries will effectively end preventable deaths of newborns and children under 5 years of age (Child Survival and the SDGs, 2021). All member countries are aiming to reduce neonatal mortality to at least as low as 12 deaths per 1,000 live births and under-5 mortality to at least as low as 25 deaths per 1,000 live births (Child Survival and the SDGs, 2021). Maternal mortality is intrinsically tied with child mortality and has continued to have wide racial/ethnic gaps, even today. Black, American Indian, and Alaska Native (AI/AN) women are two to three times more likely to die from pregnancy-related causes than white women, and that disparity only increases with age (National Center for Health Statistics). Thus, the necessity of mitigating child mortality as much as possible cannot be overstated.

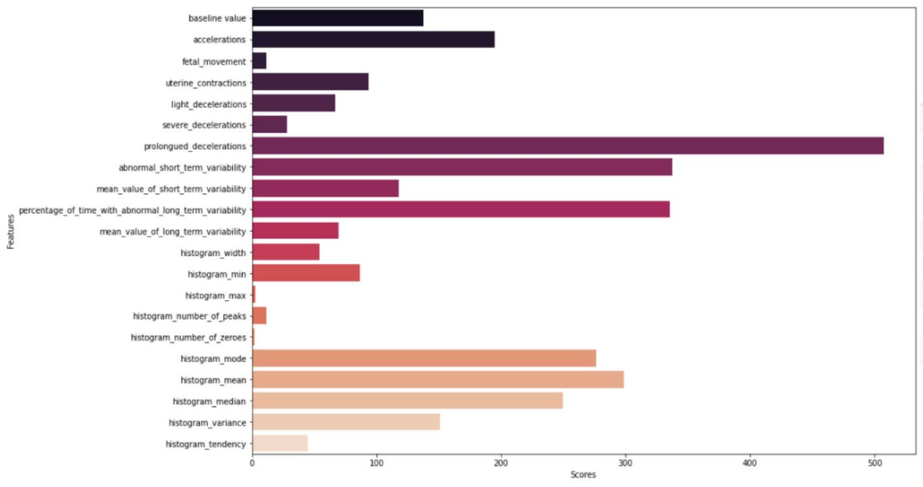
Electronic fetal monitoring or cardiotocography is a "visual representation" of uterine contractions and fetal heart rate, which has been recognized as a prominent indicator of fetal health since the 19th century (Petker, 2018). Certain fetal heart rate patterns are linked to non-reassuring fetal status, and therefore, fetal heart rate monitoring can help prevent poor fetal outcomes (Petker, 2018). The fetal heart rate monitor identifies the normal baseline rate and tracks variability, accelerations, and decelerations to provide insight into a baby's level of stress, oxygenation, acidemia (increase in hydrogen ion blood concentration), and other vital signs (Petker, 2018). A host of models were applied, including XGBoost, LightGBM, Gradient Boosting, Random Forest, Multi-layer Perceptron, Support Vector, Decision Tree, K-Nearest Neighbors, Linear Support Vector, and Logistic Regression, to the cardiotocography data to predict fetal health outcomes and level of care.

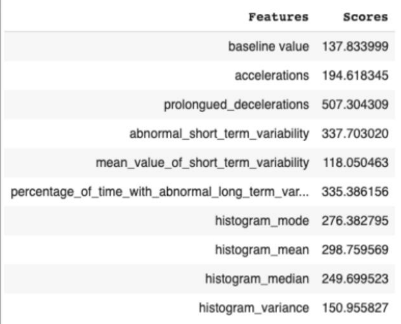
2 Related Work

Cardiotocographic data has previously been used to classify fetal health. Before the widespread adoption of neural networks and advanced machine learning methods, cardiotocograph data was largely analyzed by individual cardiologists, requiring expert interpretation, which bottlenecks its applicability and accessibility in resource-scarce areas and introduces high intra-observer error/variation (Hoodbhoy et al., 2019). As such, machine learning has revolutionized the way cardiotocograph data is being processed. Previous studies in this field of research have generally followed the approach of applying a machine learning model to cardiotocograph data to test the model's efficacy in classifying the fetal health data. Sundar et al. used a supervised artificial neural network (ANN) to classify CTG (cardiotocograph) data similarly separated into Normal, Suspicious, and Pathological, and discovered that the ANN classifier was effectively able to identify the different conditions with high accuracy. Peterek et al. applied the Random Forest, Classification and Regression Tree, and Self-Organizing Map models to cardiotocographic data, and discovered the Random Forest Model was able to classify the data with over 94% accuracy. Aside from just applying common machine learning models, other studies, like the one by Ocak et al. applied a more unique model, based on adaptive neuro-fuzzy inference systems (ANFIS), to the cardiotocographic data separated into normal and pathological examples. This model was able to classify normal and pathological examples with accuracy 97.2% and 96.6%, respectively. All of these approaches were effective in classifying the data, largely highlighting that irrespective of the model chosen, machine learning models were generally effective in successfully classifying the examples.

3 Dataset and Features

The dataset for this experiment was obtained from Ayres-de-Campos et al. and is a database repository of 2,000+ labeled records of Cardiotocograph metrics that includes base ground-truth data separated into 3 classifications: Normal, Suspect, and Pathological. The dataset includes 21 extracted features from the CTG, and whether or not the example was Normal, Suspect, and Pathological. In order to begin applying the models to our data, we started by preprocessing the data, which included removing null and duplicate values from our dataset, and visualizing the raw fetal health data, in order to see how our samples were distributed between the classifications. After our preprocessing, we were able to remove 13 duplicate examples, leaving us with 2113 samples total. Of those 2113 samples, 1646 were classified as “Normal”, 292 were classified as “Suspect”, and 175 were classified as “Pathological” (2012). Samples were separated into a standard 70/30 train/test split.





Features	Scores
baseline value	137.833999
accelerations	194.618345
prolongued_decelerations	507.304309
abnormal_short_term_variability	337.703020
mean_value_of_short_term_variability	118.050463
percentage_of_time_with_abnormal_long_term_var...	335.386156
histogram_mode	276.382795
histogram_mean	298.759569
histogram_median	249.699523
histogram_variance	150.955827

Figure 1: Feature importance scoring produced with KBest algorithm **Table 1: Most important features**

Next, to better understand feature importance and correlation, a confusion matrix was assembled to observe correlation coefficients, to which a KBest algorithm was applied to score the extracted features (Fig. 1) and eventually select the most important features to be used as our model inputs (Table 1). The feature matrix was then standardized by removing the mean and scaling to unit variance.

4 Methods

All classifiers were implemented using sci-kit-learn, except for the LightGBM classifier and the XGBoost classifier, which were implemented using lightgbm and xgboost libraries, respectively.

Logistic Regression

Logistic Regression allows us to classify our data by modeling our prediction, that is $p(y = 1 \mid x; \theta)$ using a logistic function dependent on x, θ. From Lecture 3,

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

This logistic function squashes the output between 0 and 1, which allows us to predict 1 when $h_\theta(x) > 0.5$, and 0 if $h_\theta(x) \leq 0.5$ (Ng, 2018).

Random Forest / Decision Tree

The Random Forest classifier works by establishing a group of decision trees, where each decision tree classifies the example, and the class with the most decision tree predictions becomes the model's prediction for the example. The significance of Random Forest lies in the fact that the individual tree models, as they are uncorrelated, allow for the models to cover the errors of the others. These individual decision trees work by learning decision rules from the features of the data, then using those decision rules to predict, by following down the nodes of a decision tree (Yiu, 2019). In particular, Random Forest is best suited for our multiclass classification of fetal health given the skewed/unbalanced dataset.

K-Nearest Neighbors

The K-Nearest Neighbors classifier works off of the assumption that examples close in distance to one another belong to the same class. It uses this assumption to classify examples by looking at the k nearest neighbors for every example and assigning that example to the most common class of its neighbors (Harrsion, 2019).

Support Vector Classifiers (Linear and Nonlinear)

Support Vector Machines work to classify examples by attempting to find a hyperplane decision boundary to divide our data effectively into the individual classes of data. The goal of an SVM classifier is to maximize the distance from this hyperplane boundary (it is a line in the case that the SVM is linear) to the examples in the two classes (Ng, 2020).

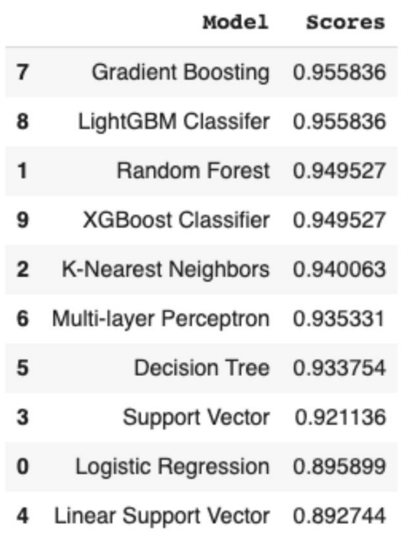
Multi-Layer Perceptron (Neural Network)

Neural networks are models that are non-linear both in $\theta$and in x, the building block of which are neurons. Neurons take in weighted inputs and can produce an output based on an activation function. Neural networks work by stacking neurons so that one neuron takes in a weighted input and feeds it to the next neuron, applying those activation functions in every hidden layer, until it reaches the output layer (Ma, 2020).

Gradient Boosting Classifiers (LightGBM and XGBoost included)

Gradient Boosting classifiers work by first using a weak learning algorithm to make a prediction, typically a decision tree, then combining subsequent machine learning models to improve the power of the initial model. Typically, the way it works is it fits the first model to the data, then a second model is built to focus on predicting the cases where the first model failed. This process continues, where each new model is attempting to fill the holes in knowledge of the previous models (Singh, 2018).

5 Experiments/Results/Discussion

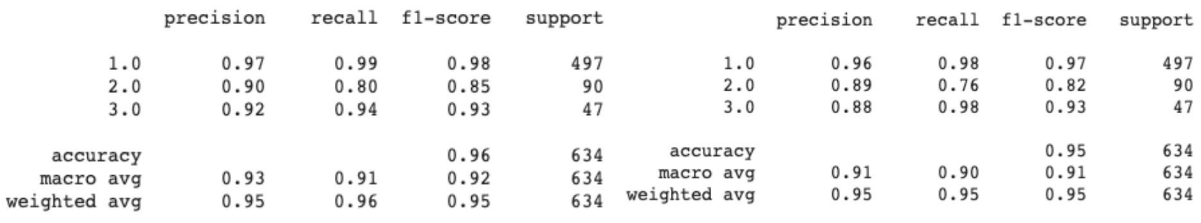


	Model	Scores
7	Gradient Boosting	0.955836
8	LightGBM Classifer	0.955836
1	Random Forest	0.949527
9	XGBoost Classifier	0.949527
2	K-Nearest Neighbors	0.940063
6	Multi-layer Perceptron	0.935331
5	Decision Tree	0.933754
3	Support Vector	0.921136
0	Logistic Regression	0.895899
4	Linear Support Vector	0.892744

Table 2: Classifier Model Scoring

The results for our models corroborated what was found in similar studies discussed in the "Related Works" section, that irrespective of the model chosen, machine learning models were generally effective in successfully classifying the examples as either Normal, Suspected, or Pathological.

From our model scores, it is clear that all of the models performed well on our dataset (Table 2), all with scores above 0.89. We also evaluated all the algorithms' performance with confusion matrices, which shows the relation between correct and incorrect predictions (Fig. 2). Of all our classifiers, Gradient Boosting and LightGBM performed the best, showing that there is a performance increase from using a gradient boosting algorithm, where new models are added to offset the errors from previous models. Interestingly enough, Random Forest performed just as well on our dataset as the XGBoost Classifier. Random Forest and XGBoost both work by combining weaker learning models to make stronger predictions, but differ in that the decision trees in Random Forest are all independent of one another, but the decision trees that get added in XGBoost are added to fill holes in previous models. The act of adding multiple models to strengthen our prediction decreases the variance of our prediction, and for XGBoost, applying the models to offset errors in previous models decreases the bias of our prediction. As the scores are the same for XGBoost and Random Forest, this implies that the decision trees that got added in XGBoost were independent of one another, and the added new decision trees didn't decrease the bias of the model, only the variance. Therefore, the models added during XGBoost did not offset the error of previous models, as is the intention of XGBoost.



```
              precision    recall  f1-score   support

         1.0       0.97      0.99      0.98       497
         2.0       0.90      0.80      0.85        90
         3.0       0.92      0.94      0.93        47

    accuracy                           0.96       634
   macro avg       0.93      0.91      0.92       634
weighted avg       0.95      0.96      0.95       634
```

```
              precision    recall  f1-score   support

         1.0       0.96      0.98      0.97       497
         2.0       0.89      0.76      0.82        90
         3.0       0.88      0.98      0.93        47

    accuracy                           0.95       634
   macro avg       0.91      0.90      0.91       634
weighted avg       0.95      0.95      0.95       634
```

Table 3: Classification Tables for LightGBM Before vs. After Optimization

As LightGBM was one of our two best-performing models, we chose this model to optimize and tune the hyperparameters - we did so using 3-Fold grid search cross-validation. In doing so, however, we noticed that the score for the classifier decreased after the hyperparameters were tuned (Table 3). This could be because Gradient Boosting models are very prone to overfitting, and tuning the hyperparameters might have caused the model to overfit the training data, decreasing the prediction accuracy of the testing data.

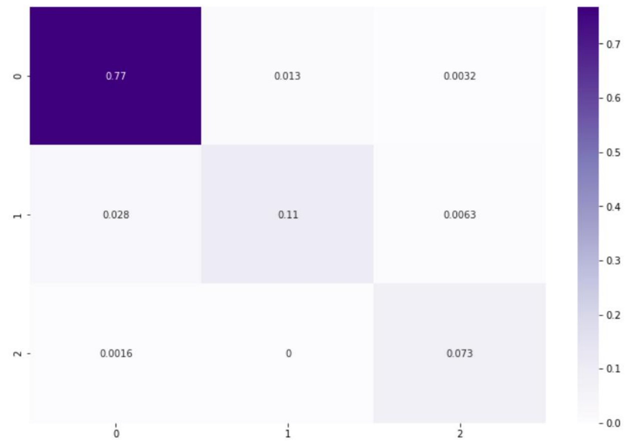


Figure 1: Confusion matrix for optimized LightGBM model

6 Conclusion

In our applications project, we set out to measure the efficacy of several different machine learning models on classifying a dataset of cardiotocographic data as either Normal, Suspect, and Pathological. After we preprocessed the dataset, we were able to remove 13 duplicate examples, leaving us with 2113 samples total. Of those 2113 samples, 1646 were classified as “Normal”, 292 were classified as “Suspect”, and 175 were classified as “Pathological” (2012). Samples were separated into a standard 70/30 train/test split. A total of 10 classifiers were run on this dataset, including XGBoost, LightGBM, Gradient Boosting, Random Forest, Multi-layer Perceptron, Support Vector, Decision Tree, K-Nearest Neighbors, Linear Support Vector, and Logistic Regression, all of which, barring XGBoost and LightGBM were implemented using sci-kit-learn, whereas those two were implementing using xgboost and lightgbm respectively. Analysis of our models showed us that LightGBM and Gradient Boosting performed the best out of all of our models, due to the ability of Gradient Boosting models to use subsequent models to offset the errors of previous models. However, in trying to optimize LightGBM and tune our hyperparameters, we saw that doing so actually decreased the effectiveness of our classifier. This could be due to the propensity of Gradient Boosting models to overfit training data, leading to worse performance on the testing data after the model was optimized. Further steps include tuning the hyperparameters for other models that we looked at, to see if there was a significant benefit to tuning the hyperparameters of models not as prone to overfitting as Gradient Boosting.

7 Contributions

Marc Huo - Feature Analysis and implementing LightGBM, Gradient Boosting, K-Nearest Neighbors, Linear Support Vector, and Decision Tree models. Collaborated on tuning hyperparameters.

JR Ereyi - Preprocessing and implementing XGBoost, Random Forest, Multi-layer Perceptron, Support Vector, and Logistic Regression models. Collaborated on tuning hyperparameters.

Our work can be found at:
https://colab.research.google.com/drive/1Au6JpBeGQjgBKgoOVTPscjKFy6To8fiz?usp=sharing

8 References

Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms. J Matern Fetal Med 5:311-318

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

“Child Survival and the SDGs.” UNICEF DATA, 6 Jan. 2021, data.unicef.org/topic/child-survival/child-survival-sdgs/.

Harrison, O. (2019, July 14). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Medium.
https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

Hoodbhoy, Z., Noman, M., Shafique, A., Nasim, A., Chowdhury, D., & Hasan, B. (2019). Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data. International journal of applied & basic medical research, 9(4), 226–230.
https://doi.org/10.4103/ijabmr.IJABMR_370_18

“Intrapartum Complications,” Family Medicine Obstetrics, 3rd Edition, 2008, pp. 454-499. Edited by Stephen D. Ratcliffe, Elizabeth G. Baxley, Matthew K. Cline, and Ellen L. Sakornbut.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.

Ma, T. (2020). Lecture on Deep Learning. Personal Collection of T. Ma, Stanford University, Stanford, CA.

National Center for Health Statistics. Racial and Ethnic Disparities Continue in Pregnancy-Related Deaths, 2019. Public-use data file and documentation. https://www.cdc.gov/media/releases/2019/p0905-racial-ethnic-disparities-pregnancy-deaths.html. 2019.
Ng, A. (2018). Lecture on Supervised Learning and Logistic Regression. Personal Collection of A. Ng, Stanford University, Stanford, CA.

Ng, A., Ma, T. (2020). Lecture on Kernel Methods and Support Vector Machines. Personal Collection of A. Ng & T. Ma, Stanford University, Stanford, CA.

Ocak, H., Ertunc, H.M. Prediction of fetal state from the cardiotocogram recordings using adaptive neuro-fuzzy inference systems. Neural Comput & Applic 23, 1583–1589 (2013). https://doi.org/10.1007/s00521-012-1110-3

Peterek T., Gajdoš P., Dohnálek P., Krohová J. (2014) Human Fetus Health Classification on Cardiotocographic Data Using Random Forests. In: Pan JS., Snasel V., Corchado E., Abraham A., Wang SL. (eds) Intelligent Data analysis and its Applications, Volume II. Advances in Intelligent Systems and Computing, vol 298. Springer, Cham. https://doi.org/10.1007/978-3-319-07773-4_19

Pettker, Christian M., and Katherine H. Campbell. "Antepartum Fetal Assessment," Avery's Diseases of the Newborn, 10th Edition, 2018, pp. 145-157.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Singh, H. (2018, November 4). Understanding Gradient Boosting Machines. Medium. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab.

Sundar, C., Chitradevi, M., & Geetharamani, G. (2012). Classification of cardiotocogram data using neural network based machine learning technique. International Journal of Computer Applications, 47(14).

Yiu, T. (2019, August 14). Understanding Random Forest. Medium. https://towardsdatascience.com/understanding-random-forest-58381e0602d2.