Classification of the cardiotocogram data for predicting fetal health using machine learning techniques

Hyun Ko
hko26@wisc.edu

Junghoon Kang
jkang@wisc.edu

Kyle Yeo
yeo9@wisc.edu

Abstract

As infant mortality rate increases, detecting abnormal activities of fetus and preventing it in advance have become important task all around the world. One of the efficient techniques to detect abnormal activities of fetus is cardiotocography (CTG). The cardiotocography (CTG) is a technique monitoring and recording the fetal heart rate and mother's uterine contraction status. The objective of this project is predicting fetal health accurately based on various features and classify it from the range of 1 to 3 ("Normal", "Suspect" and "Pathological"). We used cardiotocography data set with 2126 fetal cardiotocograms (CTGs) and 23 features. We applied kNN, Decision Tree, Random Forest, and various ensemble methods and observed their performance (test accuracy). In addition, feature importance analyzed with Random Forest and boosting algorithms to figure out the most important feature in the process of prediction. As a result of the project, we found that boosting methods (XGBoost and Histogram Gradient Boosting) are the models that have the best performance and MSTV (mean value of short term variability) is the most important feature in the prediction based on XGBoost and ASTV (percentage of time with abnormal short term variability) is the most important feature in the prediction based on Random Forest.

1. Introduction

The decrease in infant mortality rate is the key interest of many countries and international health organizations. One of the main causes of infant mortality is fetal death. However, According to Minnesota Department of Health, fetal death takes up about 1 percent of all pregnancies in the United States, but rarely being focused. [7]

Cardiotocography (CTG) is a simple and economical way of evaluating fetal health and used as a reference for judging child and maternal health state. Developed in the late 1950s by Edward Hon, Roberto Caldeyro-Barcia and Konrad Hammacher, cardiotocography is a machine to keep track of the heartbeat and movement of fetus and uterine contraction while in pregnancy and labour. [2]

Fetus is the next biological phase of an embryo, an unborn offspring of creature after nine weeks of pregnancy. [5] Though definition of fetal death varies, in general, it refers to the intrauterine death of a fetus during pregnancy. [3]

The purpose of this project is to predict fetal health and classify it from the range of 1 to 3 ("Normal", "Suspect" and "Pathological"). With the help of machine learning techniques, we will be able to interpret collected data with better accuracy and this project can also be a great reference for gynecologists or people from related field. Also, in some countries where in lack of medical manpower, machine learning algorithms can contribute to providing relatively better and quicker predictions for mothers.

The academic goal of this project is to apply various machine learning models and techniques discussed during the class to the real dataset and extract meaningful results from it.

1.1. Motivation

Recently there have been concerns regarding declining birth rates around the world. The main factor of low birth rate is economic prosperity. Also, reluctance in pregnancy stems from fear of miscarriage or side effects during and after pregnancy, such as preeclampsia. In fact, the leading cause of death in infants under 5 years of age is complications during labor in infants one month of age. According to CDC, in 2018, about 21,000 infants died in the United States.[1] The five most leading causes were Birth defects, Preterm birth and low birth weight, Injuries (e.g., suffocation), sudden infant death syndrome and Maternal pregnancy complications. Among these five causes, infant death by birth defect, Preterm birth, low birth and Maternal pregnancy complications are known to be prevented by using Cardiotocography. Cardiotocography was introduced in the 1970s, and since then, abortion rate have been decreasing. However, according to the Figure 1, 13.5 deaths per 1,000 recorded from 2017 is still high.

The purpose of Cardiotocography is to check fetal wellbeing, before and during childbirth. Based on this record, the doctor judges fertility, possibility of postpartum prob-

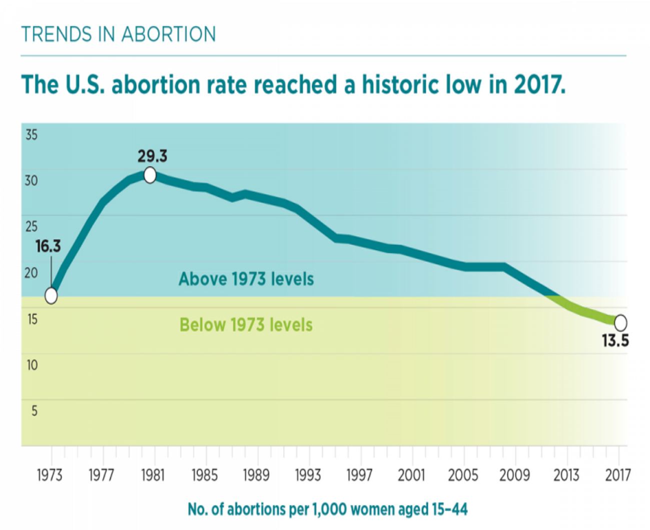


Figure 1. Number of abortions per 1,000 women aged 15-44 years, 1973-2017 from Guttmacher Institute [6]

lems, and the health of the fetus. However, incorrect interpretation of cardiotocography can put both a mother and a child at risk. For example, there are cases in which a cesarean section is required, but the diagnosis is made to have a natural childbirth. To solve this, recent studies using machine learning are actively being conducted to prevent misdiagnosis. For example, a wearable device monitoring cardiotocography in real time has recently received FDA clearance.[8] , providing high accuracy. Cardiotocography research using such machine learning is an important study because it can reduce infant mortality and reduce maternal side effects and mortality.

Our main motivation is to come up with a machine learning classifier that can accurately predict fetal health state. So, we used 2126 fetal cardiotocography data splitted into three categories ("Normal", "Suspect" and "Pathological") to create a model through an appropriate Transformer and Scikit-learn tool.

2. Related Work

In work from 2015, Hakan and Abdulhamit analyzed cardiotocography data for predicting new-born critical health cases.[9] In this research, the authors evaluated the classification performances of eight different machine-learning methods, including k-nearest neighbors and random forest, on the antepartum cardiotocography (CTG) data. When it comes to splitting data to train and test parts, they used k-Fold cross validation to avoid overfitting. By training many machine learning algorithms, they found that the overall accuracy of the classifiers was approximately 98%, which is extremely high and meaningful. In addition, they trained all classifiers on the data without any feature selection methods.

There is another study that focused on the bagging ensemble machine learning algorithm on CTG data to classify fetal heart rate as normal or abnormal[10]. This study also used k-Fold cross validation without any feature extraction and evaluated performances of ensemble machine learning algorithms. The research reported that the overall classification accuracy was approximately 98%, and the highest classification accuracy observed from bagging model with random forest ensemble classifier, which was 99.02%. Unlike how related studies proceeded, this report will consider feature preprocessing methods such as dimensionality reduction via PCA and feature selection based on correlation coefficient. In addition, we will figure out the most important feature for the prediction.

3. Proposed Method

3.1. Method

For our project, using 22 features, we are going to compare various machine learning models and find the best one. The candidates will be based on models we learned in class: k-Nearest Neighbor Method, Decision Tree and various ensemble methods. We will divide data set into two subsets: train set, and test set with the proportion of 70:30 and 80:20 (depending on method) using train-test-split in sklearn module.

3.2. Metrics

Our goal is to make classification accuracy of our model as close to 100% and F1 Score to be close to 1. Accuracy, F1 score, Precision and Recall are the common methods to evaluate multi-class classification model.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ number\ of\ predictions\ made}$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot recall}{Precision + recall}$$

We aim to get above 90% in Accuracy and 0.9 in F1 score from at least one model.

3.3. kNN

k Nearest Neighbors (kNN) is one of the non parametric supervised machine learning algorithms, which is mostly used for classification and regression. The kNN, in general, called lazy learner, which means that it has no training period except remembering data and it makes the algorithm much faster than any other machine learning algorithms.

However, there are also some obvious disadvantages of using kNN algorithm. The most critical downside of it is that the algorithm is susceptible to noise in the data such as missing values and outliers. That is, it could be complicated to implement kNN on the data that have lots of noises. Also, kNN is not an appropriate for large data because of distance measurement.

In this project, we used *KNeighborsClassifier* for implementing kNN. we applied kNN without any formal grid search methods and only considered euclidean distance for calculating distance between nearest neighbors. Before algorithm implementation, we split the data into train and test sets by using *train-test-split* with the proportion of 80:20. Since we only considered euclidean distance, we did not consider higher value of k. So, in order to figure out the optimal k value, we fitted the algorithm with simply looping over arbitrary k from 1 to 20. Then we reported the test accuracy corresponds to the each k values.

3.4. Decision Tree

Decision Tree is a powerful and easily interpreted supervised machine learning algorithm that can be used both regression and classification. Due to its relatively simple logistics: top-down hierarchy, yes-or-no (boolean) function, it is common to use decision tree at the early stage of machine-learning analysis.

The basic idea of splitting decision tree is information gain. In short, the better the split, the higher the information gain.

There are two ways of evaluating impurity within category. One of them is Entropy, a numerical measurement indicating impurity.

$$Entropy = -\Sigma_i(p_i)\ *\ log_2 p_i$$

, where p refers to the proportion of data being included within category i. As entropy gets closer to zero, it means more pure, while closer to one, meaning impure.

From entropy of parent node and child node, we can calculate information gain:

$$Information\ gain =$$

$$entropy_{parent} - weighted\ average * entropy_{children}$$

Decision tree continues training process in the way of maximizing information gain from each split.

However, the downside of decision tree is susceptibleness to overfitting: a profound performance difference between training and test data. Without any parameters setting, the size of tree gets infinitely large. Therefore, additional pre-pruning and post-pruning is recommended to avoid overfitting.

In this project, we used decision tree in conjunction with GridSearch to find best hyperparameter pairs for pre-pruning. We set minimum samples split ranging from 2 to 4, and maximum depth from 6 to 16. Also, used 10-fold cross validation to reduce overfitting.

3.5. Random Forest

Random forest is an ensemble method consisting of multiple decision trees. Random forest is used for various problems such as detection, classification, and regression. The primary reason for choosing random forest is there are few variables that p-value is not significant indicating it might contain various noises in our data and model. The random forest method is that decision trees have slightly different characteristics due to randomness, so that the predictions of each tree are uncorrelated. Therefore, it guarantees high accuracy even when noise is included and enables relatively simple and fast learning. In Random forest model, we used ‘n estimators’, ‘criterion’, ‘max depth ’ as our hyperparameters and to find best parameter, we used Grid search.

3.6. Gradient Boost

Gradient Boosting is a original boosting technique that progressively improves the residual error by passing the error from the previous learning result to the next learning.

$$Error = h_1(x) + Error2$$
$$Error2 = h_2(x) + Error3$$
$$...$$
$$y = h_0(x) + h_1(x) + h_2(x) + ... + smallError$$

Gradient boosting should be set carefully because it responds sensitively to parameters. However, it can be applied to large data, does not require a separate scale, and the performance is relatively good.

3.6.1 Histogram Gradient Boosting

Histogram-based gradient boosting is an improved version of gradient boosting in terms of speed and performance. It divides variables into 256 sections. Therefore, when partitioning a node, the optimal partition can be found in a quickly manner, and affect from missing data is negligible. For our model, we used grid search to find best hyperparameter using learning_rate of 0.01,0.1,0.3 and max_depth of 1,5,10,20.

3.6.2 XGBoost

XGBoost is one of the efficient implementations of gradient boosting. However, XGBoost considered as more powerful and advanced algorithm because it is more regularized than any types of implementation of gradient boosting. XGBoost also can be interpreted as a tree based ensemble machine learning algorithm. In the process of dealing with errors, newly created trees, in general, detect and correct errors of existing trees in the model and trees keep added for the purpose of performance improvement. For this project, we optimized the model’s hyper parameter settings with optuna and formal grid search. Then, we evaluated the optimal values of each hyper parameter and classification accuracy of the model.

We used *XGBClassifier* for the model setting. For the hyperparameters setting, we tried to find the best n-estimator by setting arbitrary n-estimators of 30, 50, 100, 300, 500, and 1000. Also for the learning rate, we randomly set the values as 0.01, 0.03, 0.05, and 0.1, which are mostly used values for the learning rate and tried to find the best among them. Lastly, we used optuna for the model optimization. We created study with maximization direction by *optuna create study* and optimized the model with it by 50 trials.

3.7. Voting

Bagging is a method of voting by combining samples, but Voting is of selecting the final result through a method of final voting on the results derived by different algorithms method.

Our model uses the soft vote method, because rather than the ”hard vote” method, which determines the final value of the result by voting, the ”soft vote” which determines the final value with highest probability from the average value of each calculated probability is more reasonable.

The models we used for Voting are ’Random Forest’,’XGboost’,’Decision Tree’,’Gradient Boost’ and ’Histogram based Gradient’ with best hyperparameter tuning from each algorithm . This model was selected because it is easier to compare with the individual models because they have already been used individually previous model.

3.8. Stacking

Stacking is also the type of ensemble methods. Unlike other ensemble methods that take on orthodox approach(train - predict - evaluate), Stacking re-trains the results from individual ’base learners’ as training sets. Then, the final model (meta-classifier) provides the final result.

Despite its powerful performance, stacking is highly likely vulnerable to overfitting. If individual base learner overfits, that is also the case of meta-classifier. Therefore, in this project, we used 10-fold cross validation to avoid overfitting. Also, we implemented stacking algorithms(XGBoost, Random Forest), each with two base learners. In that model, we used logistic regression method as a meta-classifier, which is a convention in classification.

The models we used for stacking were ’Random Forest’, ’Xgboost’, ’Decision Tree’, ’Gradient Boost’ and ’Histogram based Gradient boost’ with best hyperparameter tuning from each algorithm. The model we used in stacking were same with model we used in voting which we can also compare the performances between Stacking and Voting.

4. Experiments

4.1. Hardware and Software

We used Jupyter Notebooks running Python 3.8 or later versions. We mainly used pandas and numpy libraries for cleaning and analyzing data, matplotlib for visualization, and sklearn, xgboost, and optuna for implementing machine learning models.

4.2. Dataset

We will use “Cardiotocography Data Set” provided by UCI Machine Learning Repository[4] Machine Learning Repository. Cardiotocography Data Set contain 2126 fetal cardiotocograms(CTGs) with 23 features. The CTGs were classified by three expert obstetricians and a consensus classification label assigned to each of them. The 23 features consist of the following data fields. In order to find out noise and unnecessary features, we used correlation coefficient and p-value between “Fetal health” and the rest of the features(shown in figure 2 and figure 3). The correlation coefficient was not relatively high, and a few p values were significant. The reason for low correlation coefficient is there are too many repetitive zeros found in our variables. Since we don’t have expert knowledge on specific variable,we couldn’t decide whether 0 is important indicator or not. Therefore,we decide to use all the variable in our model.

Finally, in our dataset response variable “fetal health”, the ratio of 1 is far superior (Figure 4). So, when the number

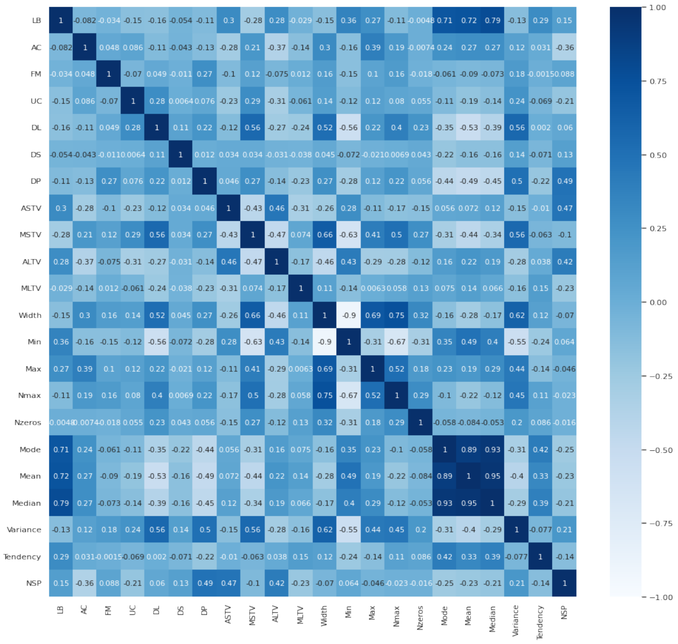


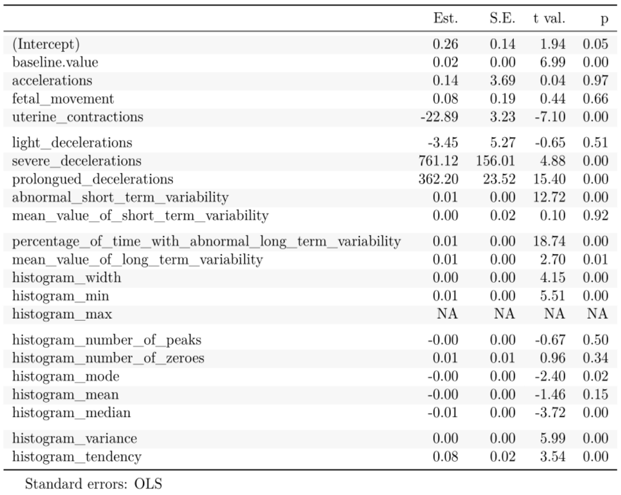
Figure 2. Heatmap across all features

of test sets of labels 2 and 3 is absurdly small compared to the training set, sampling bias may occur. To prevent that, we did stratify=y in the ”train test split function to divide the data according to the class ratio.

5. Results and Discussion

5.1. Results

First, we used kNN, and the best results were obtained 89.9% when K=3. The accuracy did not exceed 90%. As shown in Figure (5), the accuracy decreases as K increases. This is a weakness of kNN Algorithm for high dimension data set. We use Euclidean distance, and usually if dimension of model higher than 4, the performance is relatively poor.



	Est.	S.E.	t val.	p
(Intercept)	0.26	0.14	1.94	0.05
baseline.value	0.02	0.00	6.99	0.00
accelerations	0.14	3.69	0.04	0.97
fetal_movement	0.08	0.19	0.44	0.66
uterine_contractions	-22.89	3.23	-7.10	0.00
light_decelerations	-3.45	5.27	-0.65	0.51
severe_decelerations	761.12	156.01	4.88	0.00
prolongued_decelerations	362.20	23.52	15.40	0.00
abnormal_short_term_variability	0.01	0.00	12.72	0.00
mean_value_of_short_term_variability	0.00	0.02	0.10	0.92
percentage_of_time_with_abnormal_long_term_variability	0.01	0.00	18.74	0.00
mean_value_of_long_term_variability	0.01	0.00	2.70	0.01
histogram_width	0.00	0.00	4.15	0.00
histogram_min	0.01	0.00	5.51	0.00
histogram_max	NA	NA	NA	NA
histogram_number_of_peaks	-0.00	0.00	-0.67	0.50
histogram_number_of_zeroes	0.01	0.01	0.96	0.34
histogram_mode	-0.00	0.00	-2.40	0.02
histogram_mean	-0.00	0.00	-1.46	0.15
histogram_median	-0.01	0.00	-3.72	0.00
histogram_variance	0.00	0.00	5.99	0.00
histogram_tendency	0.08	0.02	3.54	0.00

Standard errors: OLS

Figure 3. A table with regression coefficients, standard errors, t-values, and p values.

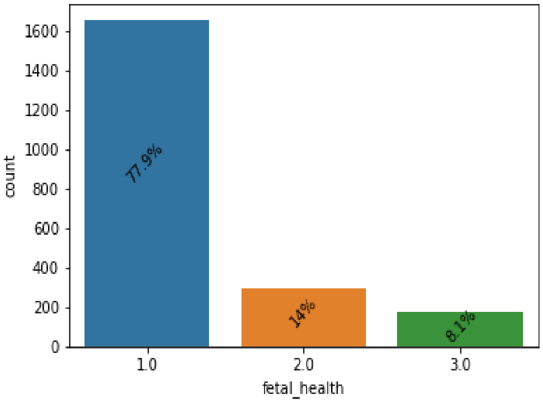


Figure 4. Proportion of Fetal health feature labels

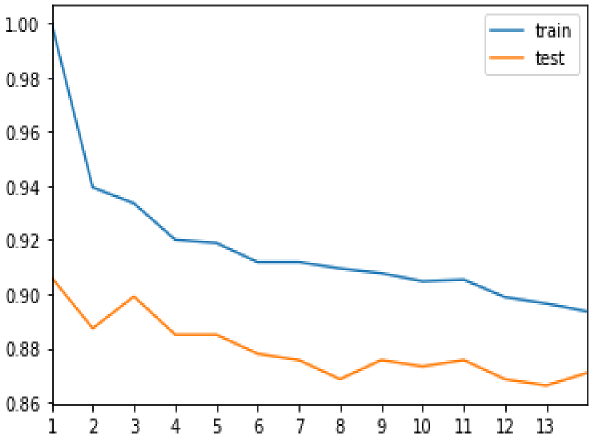


Figure 5. k-NN varing number of neighbors

Next, the decision tree algorithm using grid search was used. We recorded 92.49% accuracy using *’max_depth’=16* and *’min_samples_split’=4* parameters. This was higher than KNN. However, due to the characteristics of the tree model, it tends to overfit and the generalization performance is not good.

LB	FHR baseline (beats per minute)
AC	# of accelerations per second
FM	# of fetal movements per second
UC	# of uterine contractions per second
DL	# of light decelerations per second
DS	# of severe decelerations per second
DP	# of prolongued decelerations per second
ASTV	percentage of time with abnormal short term variability
MSTV	mean value of short term variability
ALTV	percentage of time with abnormal long term variability
MLTV	mean value of long term variability
Width	width of FHR histogram
Min	minimum of FHR histogram
Max	Maximum of FHR histogram
Nmax	# of histogram peaks
Nzeros	# of histogram zeros
Mode	histogram mode
Mean	histogram mean
Median	histogram median
Variance	histogram variance
Tendency	histogram tendency
CLASS	FHR pattern class code (1 to 10)
NSP	fetal state class code (N=normal; S=suspect; P=pathologic)

Table 1. Data Set Variable Information

To get higher accuracy, we used ensemble method. First, when using the random forest without hyper parameters, it recorded 93.7%, which is slightly higher than the decision tree, but the performance did not improve significantly. The reason for this is that the random forest is also based on a tree.So, we found the best parameter through grid search, and found the accuracy 94.6% using *'n_estimators'=500*, *'criterion','entropy'*, *'max_depth=20'*. We were able to confirm that it came out better than kNN, Decision Tree, and Random Forest without tuning.Furthermore, regardless of the criterion, *'n_estimators'=500, 'max_depth'=20* were the most optimal cases. When the criterion is gini, it can be seen that 'max depth' affects model performance, and when 'entropy', 'n estimators' have a relatively affect to model performance.

Another ensemble method we used is boosting method. We used Histogram based Gradient boost and Xgboost, which are known as the latest technologies among boosting algorithm. The boosting method requires more learning time because it is more sensitive to parameter variables than other methods. First, histogram based gradient boosting also found a hypermeter through grid search. As a result, with *'learning_rate'=0.3*, *'max_depth'=5*, performance 95.77% accuracy. In addition, the Xgboost model optimized with Optuna, shows 95.45% accuracy. Moreover, Optuna framework provide module that shows importance of hyperparameters and from figure 6 we could find 'max_depth' is the most importance hyperparemeter in this model.Both gradient boosting models showed an accuracy of over 95% which were the best results among the previous models.

Finally, we used Stacking and Voting method which both methods using various different models on base. First, the accuracy of stacking was 95.54% and voting was 95.31%. The performance of both models are lower than some of individual model that used for base. It can be seen because some individual models like boosting based model are relatively stronger than others.

Model	**Accuracy**	**F1 Score**
KNN	89.9%	0.90
Decision Tree	92.49%	0.92
Random Forest	**94.6%**	0.95
HisGradient	**95.77%**	0.96
XGboost	**95.45%**	0.96
Voting	**95.31%**	0.95
Stacking	**95.54%**	0.96

Table 2. Accuracy and F1 Score for each model

5.2. Discussion

Although we could extract meaningful results from this project, there still exists potential shortcomings.

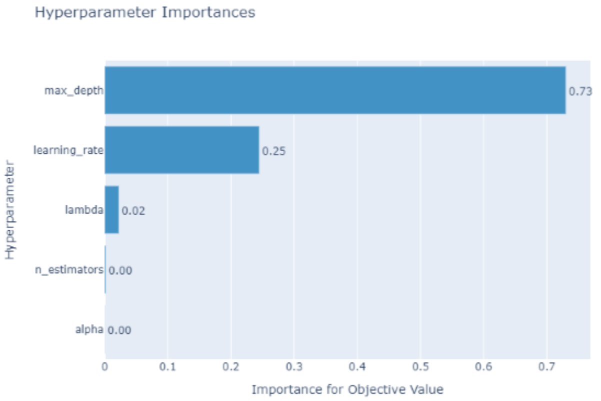


Figure 6. plot of the impact of hyperparameters on the accuracy of the predictions

First, the dataset lacks enough observations: only 2126. The lesser observation a dataset has, the lesser data an algorithm can train and predict. This might lead to overfitting: a huge difference between training accuracy and test accuracy. Despite the implementation several counter-overfitting measures, we cannot affirm that there was no possibility of overfitting at all.

Second, from our data, the correlation coefficient was not relatively high, but a few p values were significant. Some variable shows an inverse relationship between correlation coefficient and p values. However, in our data, there were cases where variables with high p values recorded relatively high correlation coefficients. In Figure 7 the correlation coefficient table, a strong correlation was found between histogram variables. We call this multicollinearity: a correlation between independent variables. There are some methods to solve multicollinearity like variable elimination or dimension reduction. However, in terms of variable elimination, as the elimination process is based on correlation, it can possibly lead to a bias due to a large number of meaningful zeros in our data. And for dimension reduction method like PCA(Principal component analysis), we didn't have enough data for it. In fact, according to Figure 8, the accuracy of random forest across number of components vary. In short, in terms of solving multicollinearity problem, we were unsure of the effectiveness two mentioned methods to troubleshoot multicollinearity.

Third, as mentioned above, there were too many observed 'zeros' on two particular columns: severe decelerations, prolongued decelrations. According to Figure 9, the proportion of observed zeros from severe decelerations and prolongued decelartions far outweighted those of the rest. However, from our knowledge, it was impossible to judge whether these zeros genuinely affected the overall perfor-

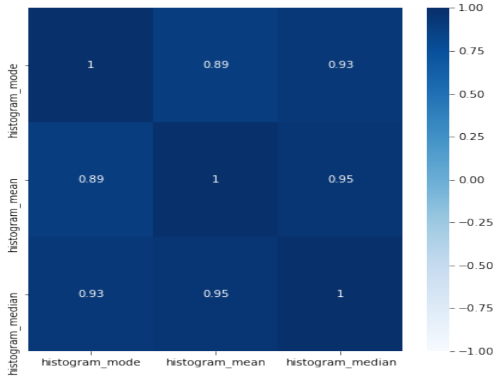


Figure 7. Correlation among histogram features

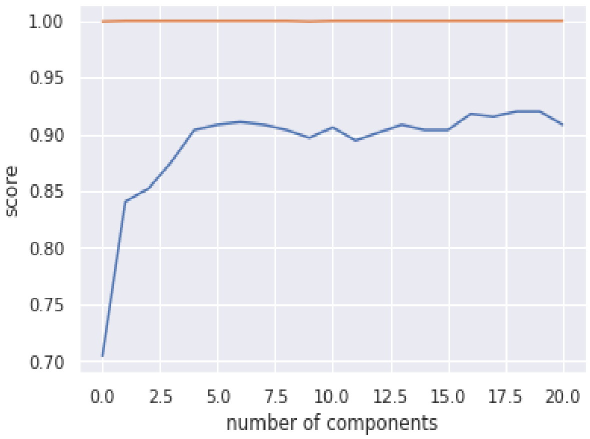


Figure 8. Score vs Number of Component using PCA with Random Forest

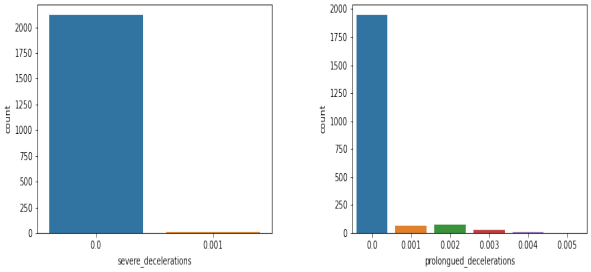


Figure 9. A bincount of deceleration features

mance of models and led to further confusions.

In short, the lack of enough observation data, unsureness regarding multi-colinearity among histogram features and over-repetitive appearance of zeros were our potential weakness of our project and left room for improvements.

5.2.1 Feature Importance

Based on all the results that we got from the experiments, we concluded that beyond predicting child mortality and prediction accuracy of each machine learning models, it would be interesting to provide what the most important features are for the prediction. Since we observed the overall meaningful observations from both Random Forest and XGBoost models, we studied impurity based feature importance analysis with both models.

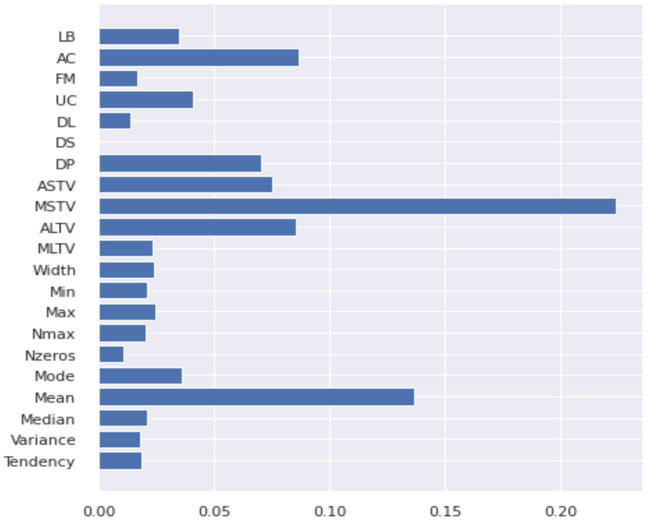


Figure 10. Feature Importance of XGBoost Optuna

First, we have the feature importance analyzed and determined by XGBoost model. According to the Figure 10, the features that observed to be important in predicting fetal health are MSTV(mean value of short term variability), Mean(histogram mean), ALTV(percentage of time with abnormal long term variability), AC(num. of accelerations per second), ASTV(percentage of time with abnormal short term variability), and DP(num. of prolongued decelerations per second). Overall observations are what we expected to see and seemed somewhat reasonable, but we were surprised to observe 'Mean' feature as one of the important features because we didn't expect any histogram related features would have meaningful importance in prediction. In addition, there is one thing to note that based on common knowledge, 'DS' (num. of severe decelerations per second) feature should have significant importance in the prediction, but the values in the DS feature are almost zeros in our data set as shown in Figure 9 so that the importance value also converges to 0.

To make our analysis more convincingly, we also determined the feature importance by Random Forest algorithm to compare with the observations made by XGBoost above. We also analyzed impurity based feature importance as we

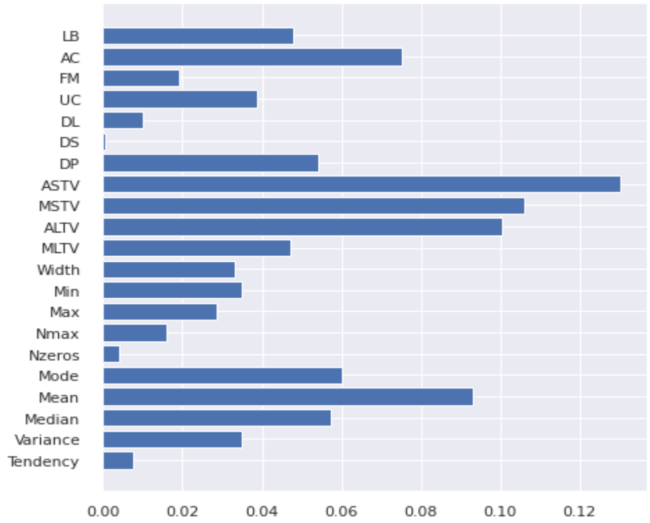


Figure 11. Feature Importance of Random Forest

did with the XGBoost model. Most importantly, we found that the overall distributions of the feature importance based on both models are similar. According to Figure 11, however, we saw that the ASTV is the most important feature and MSTV is the second most important feature, which is different from the earlier result. In addition, MODE (histogram mode) feature observed as somewhat important feature at this time, which is unexpected result.

Even though there is difference between the distributions of the values of feature importance determined by both XGBoost and Random Forest models, both models shared many common features that are important. However, we only considered and obtained impurity based feature importance and so for the future direction on this feature importance analysis could be advanced with applying permutation feature importance with more various algorithms.

6. Contributions

As a group, all members participated across the entire pipeline of the project, from data cleaning to conducting, deriving and discussing meaningful observations. For the writing process, Hyun was responsible for the introduction, proposed methods, discussion and conclusions part, Junghoon was responsible for motivation, dataset, and results sections, and Kyle was responsible for abstract, related work, proposed methods, and experiment sections.

When it comes to the modeling process, Hyun was responsible for works in the Decision Tree and Stacking, Junghoon was responsible for the works in Random Forest, Gradient Boosting, and Voting, and Kyle was responsible for the works in kNN and XGBoost Optuna. All members contributed to visualizations and feature importance analysis.

As such, under the evenly distributed tasks, every members of the group well collaborated with each other under any circumstances across the board.

7. Conclusions

The initial motivation of this project was to come up with an effective machine learning model to classify fetal health which provides similar results to those of actual one. Accordingly, we implemented several potential algorithms: kNN, Decision Tree, Random Forest, Boosting, Stacking. From the vast majority of our models, we were able to reach our baseline accuracy level, 90% and reliable F1 score.

In addition, we strived for figuring out which was the most important feature explaining fetal health state. In order to compare between two highest-accuracy providing algorithms, we went on feature importance analysis on both XGBoost and Random Forest.

Compared to ours, previous studies had achieved higher accuracies. Sahin,H. and Subasi, A. had provided 99.2% accuracy. What differentiated us from previous studies was that we more focused on the ways dealing with features rather than just focusing on prediction such as each feature's correlation with fetal health, multi-colinearity among histogram columns, feature importance.

Future direction on this study could be done in the way of more bountiful observations coupled with outside of cardiotcography data such as more related information of a mother and genetic information. Plus, various multi-colinearity troubleshooting methods among histogram columns will also be a great idea. Lastly, beyond just figuring out mere importance of features, applying various feature extraction techniques to explain feature health would be ideal.

8. Code

All the associated such as code, notebooks, project proposal, and final report of this project can be accessed at the Kyle's GitHub repository: `https://github.com/hoonkiyeo/FetalHealth_Prediction`

References

[1] Causes of infant mortality:infant mortality. Centers for Disease Control and Prevention, 2021.

[2] Cardiotocography. Cardiotocography — Wikipedia, the free encyclopedia, 2021. [Online; accessed 05-December-2021].

[3] Centers for Disease Control and Prevention. Fetal deaths, 2020.

[4] D. Dua and C. Graff. UCI machine learning repository, 2017.

[5] S. Ghosh, M. Raghunath, and J. K. Sinha. Encyclopedia of animal cognition and behavior, 2017.

[6] R. K. Jones and E. Witwer. Induced abortion in the united states, May 2020.

[7] Minnesota Center for Health Statistics. Infant and fetal mortality, 2021.

[8] Reuter. Startup gets fda clearance for device that can monitor baby's heart rate, Mar 2020.

[9] H. Sahin and A. Subasi. Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques, 2015.

[10] A. Subasi, B. Kadasa, and E. Kremic. Classification of the cardiotocogram data for anticipation of fetal risks using bagging ensemble classifier. *Procedia Computer Science*, 168:34–39, 2020. "Complex Adaptive Systems"Malvern, PennsylvaniaNovember 13-15, 2019.