Towards Automated Fetal Health Classification from Cardiotocograms

Ezgi Siir Kibris, Gabrielle Kosoy, Julie Fleischman, and Siladitya Khan
Goergen Institute for Data Science
University of Rochester, Rochester, NY, USA

Abstract— This project, uses the Kaggle dataset "Fetal Health Classification" to identify Cardiotocograms (CTG) output as normal, suspect, or pathological. We report classification models such as Support Vector Machine (SVM), Multinomial Logistic Regression, and XG-Boost that achieve ~90 % accuracy and an F1 score of ~0.8. Additionally, we performed SHapley Additive exPlanations (SHAP) to assess the importance of individual features to model prediction. Subsequently, we use Uniform Manifold Approximation and Projection (UMAP) to report cluster formation and offered insights into cluster memberships. Notably, we achieve 83% and 100% cluster membership allocations by using two variants of U-MAP alongside density-based clustering methods. The overarching goal of this project is to explain child and maternal mortality from biophysical markers like CTG data.

Index Terms—Cardiotocograms (CTGs), Fetal Health, Support Vector machines (SVM), Gradient-Boosting, SHAP, UMAP.

I. INTRODUCTION

Cardiotocograms (CTGs) [1] are a method for tracking fetal health during pregnancy. Ultrasound captures the fetal heart rate from transducers placed on the mother. The output from the machine is a cardiotocograph. This graph shows variability in heart rate such as decelerations, accelerations, called bradycardia and tachycardia. While some variability in heart rate is normal, persistent accelerations and decelerations in heart rate can indicate problems during labor and intervention may be necessary. Changes in fetal heart rate have always been observed by eye by doctors. The graph provides other relevant information as well such as the features that describe the shape of the peaks associated with the heartbeat. It is challenging for a doctor to observe all these features at once during labor while trying to make periodic decisions about the fetuses' health. Our goal is to use the Kaggle dataset "Fetal Health Classification" [2] to identify CTG output. This dataset was collected in Porto, PT by a group interested in creating a model to better predict fetal health. Currently, CTGs are highly sensitive but not good in specificity, which makes them excellent predictors of good fetal health but more challenging to identify problems. This study will be reported in the following chapters: Literature Review, Data, Hypothesis/Goals, Results, and Conclusion.

II. LITERATURE REVIEW

Significant work has been done in the past to classify fetuses as suspect or pathological using cardiotocography. With the abundant information that the exam returns on fetal heart rate, variability, contractions, and other movements, researchers have proven that this exam is a strong classifier for fetal health [17]. Further, this classification should be conducted using approximately thirty indicators [18]. In terms of what form of classifier works best, there is mixed literature. According to F. Akhtar and others, SVM is the best classifying technique [12]. There is also evidence from Rajasekhar Nannapenini that a decision tree is not the route to take [15]. The prevailing theory is that a combination of ML models will be a better predictor than any singular model, however, the best model is unknown [2]. This is where our research picks up, in an attempt to construct a layered model for predicting fetal health.

III. DATA

A.Descriptive Analysis

The dataset contains **2126** records with **21** unique features associated with CTG exams. The classes for "fetal_health" associated after the obstetrics exam are **Normal**, **Suspect**, and **Pathological**. Fig 1. shows the pie chart and the share of types of fetal health. There is class imbalance in the dataset. 78 % of the fetal health is classified as normal, %14 as suspect and % 8 as pathological. There are more normal class fetuses than pathological or suspect classes.

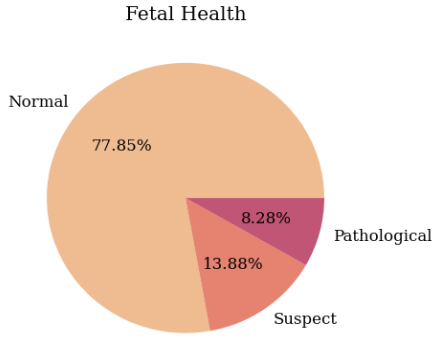


Fig 1. Pie chart showing the share of types of fetal health

Fig 2. displays a two-dimensional representation of the fetal health after applying dimensionality reduction techniques PCA and t-SNE to the features of the dataset.

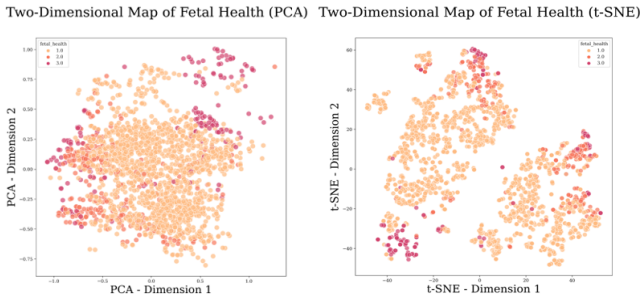


Fig 2. Two-dimensional representation of fetal health after dimensionality reduction

In both graphs, the normal health fetuses (orange) are in the middle part, whereas pathological and suspect fetuses are clustered in the upper right or lower left parts.

B. Exploratory Analysis

Before constructing the model, it is important to inspect the available variables and look for any evident patterns or complications. We inspect the correlation heat map of the feature variables. Any very strong correlations might indicate significant factors or really weak correlations with fetal health might be insignificant variables. It looks like Figure 2 below shows a mix of relationships. Some of the more interesting ones appear to be between the average value of short term variability and decelerations in fetal health rate, and between the time with abnormal long term variability and abnormal short term variability. While many of these variables are implicitly overlapped, they all seem worth investigating with the models we will build.

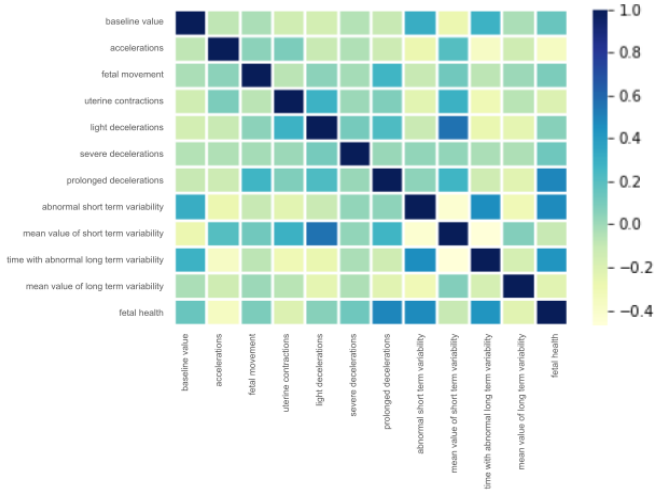


Fig 2. Correlation Heat Map between All Variables

In terms of relation to fetal health, prolonged decelerations and abnormal short term variability as well as time with abnormal long term variability seem to be correlated the strongest. Using pairwise plots, we can investigate the relationship between the inputs before continuing to the preprocessing of data.

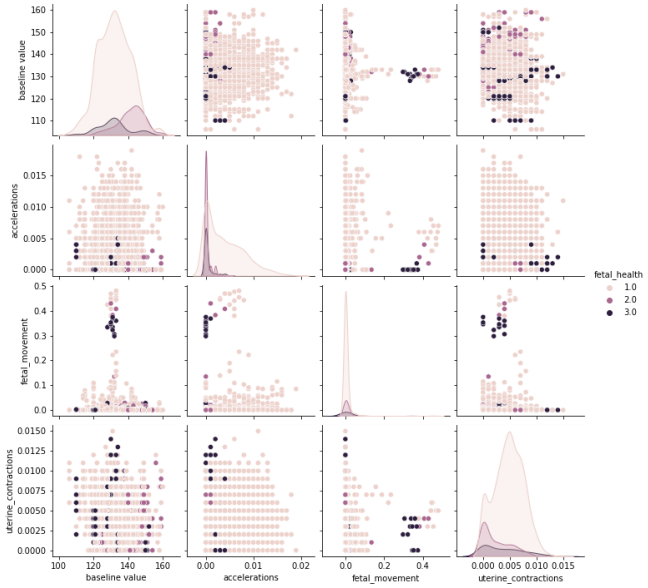


Fig 3. Pairwise Plot between Variables Excluding Decelerations and Abnormality Measurements

In Figure 3, the three classes of fetal health are 1 (pink) for normal, 2 (purple) for suspect, and 3 (black), for abnormal. What the above pairwise plots indicate is that, as is expected, most of the data is normal fetuses. Within that, the graphs showing clusters are fetal movement as compared to uterine contractions, as well as fetal movement against accelerations. The remainder of the plots shown seem to have a generally random distribution. We can now look at the deceleration metrics, including frequency of light, severe, and prolonged decelerations.

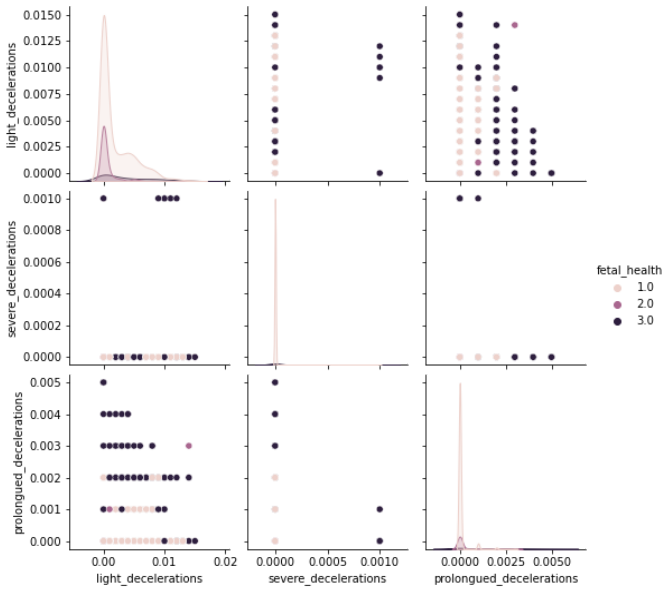


Fig 4. Pairwise Plot between Decelerations Variables

There is not a lot to examine in these graphs in figure 4. No clustering is apparent, so we have limited ability to predict the influences these might have on fetal health.

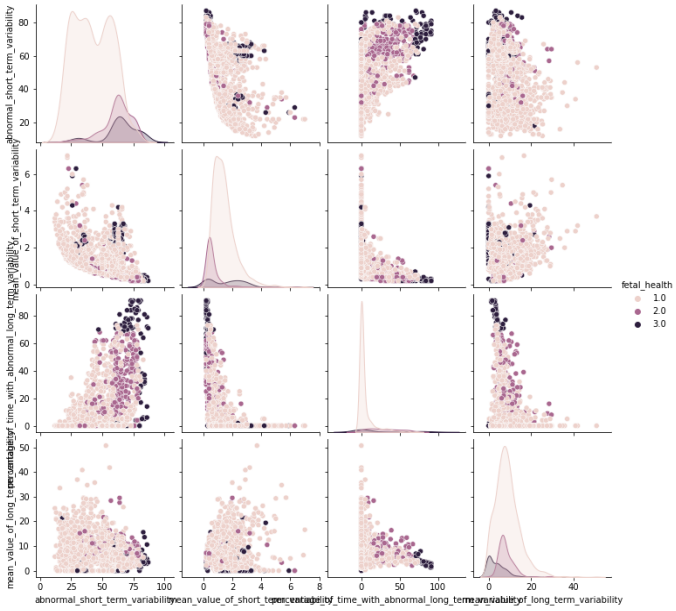


Fig 5. Pairwise Plots between Metrics of Fetal Heart Rate Variability

The pairwise plots represented in Figure 5 do a better job than in Figure 4 showing some clusters in the data. It seems that in comparing long and short-term abnormalities, abnormal and suspect fetuses fall at the extremities of the plots. Given all the patterns and clustering evident, it seems worthwhile to evaluate all of the feature variables in initial models, and potentially eliminate some of the lesser impact variables in subsequent models if needed.

C. Data Preprocessing

Due to the wide range of numeric values in the CTG dataset, all 21 features were scaled using the sklearn package standardscaler function. The features observed during labor that was scaled included contractions, fetal movement, heart rate accelerations/decelerations, and cardiotocography peak details such as min/max or variance. The target feature to predict is fetal health status. The three statuses are: 1- healthy, 2- suspect, and 3- pathological.

IV. HYPOTHESIS/GOALS

We hypothesize that a computational model trained on CTG data can predict fetus health status. The models used in this project are SVM, Logistic Regression, XG-Boost, and dimensionality reduction techniques such as t-SNE and PCA. We classify observations in the dataset with accuracy. Overall, this project may help obstetricians identify fetal health better and may reduce child and maternal mortality.

IV. EVALUATION METRICS

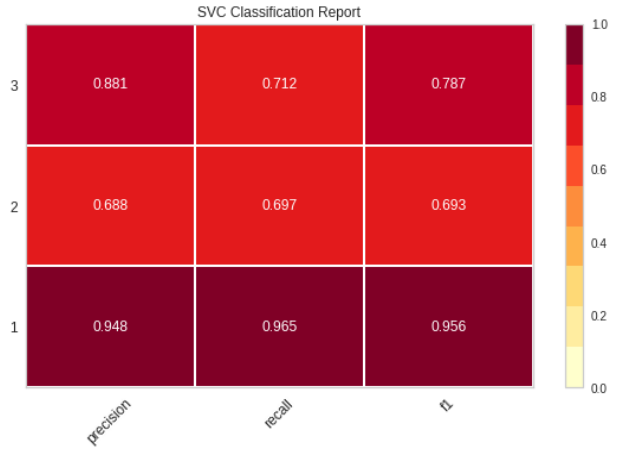
We use some essential metrics that are commonly prescribed to assess the performance of our model. The Regression and Classification approaches are assessed using Accuracy and F1-Score. Now since we are dealing with a problem that is imbalanced in the minority "pathological" state class, we use the "macro" form of the F1 score that compensates for the performance in predicting the majority class correctly. The Clustering approaches are evaluated based on calculating the `adjusted_rand_score` and `adjusted_mutual_info_score` metrics. The `adjusted_rand_score` from sklearn.metric package "evaluates all pairs of samples and counting pairings that are allocated in the same or different clusters in the anticipated and true clusterings, the Rand Index computes a measure of similarity across two clusterings" [19]. The `adjusted_mutual_info_score` "compensates for the fact that, regardless of whether more information is provided, the MI is often greater for two clusterings with a larger number of clusters" [20]. We also compute the percentage of points allocated to clusters based on their original data-set size by a metric known as `data_assigned_to clusters`.

V. REGRESSION & CLASSIFICATION APPROACHES

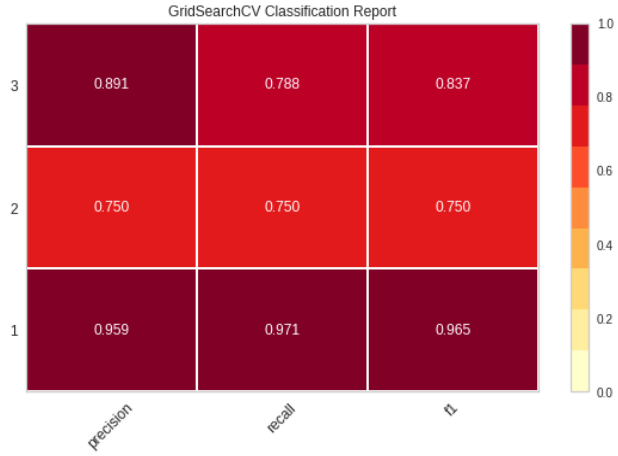
A.Support Vector Machine (SVM)

An SVM model without optimized hyperparameters was able to predict the fetal health status 1 (healthy) with an f1 score of over 95%. (Matrix 1) While the f1 score can be considered more difficult to interpret because it is a harmonic mean of precision and recall, in this case, it was a useful measure of how good the model is. F1 score is useful for this study because when determining fetal health status it is very important to not overdiagnose and do unnecessary procedures. False negatives and false positives are important. The precision and recall for fetal health status group one is also at 95% or over. As is seen in the descriptive analysis, the dataset is not distributed evenly across the three fetal health statuses. This is reflected in the success of the SVM model. Fetal health statuses two and three make up less than 25% of the observations in the dataset. The f1 score for status two and three is 0.69 and 0.79 respectively. It is good that status three particularly has higher precision, 0.88, because this indicates the model does not generate a false positive which can lead to false interventions during labor. The next step was to do hyperparameter tuning on the SVM model. Hyperparameter tuning was done using a grid-search

cross-validation function from the sklearn package. (Matrix 2)



Matrix 1. A matrix of precision, recall, and f1 score for the first SVM model in this study for classifying CTG data into three fetal health statuses.



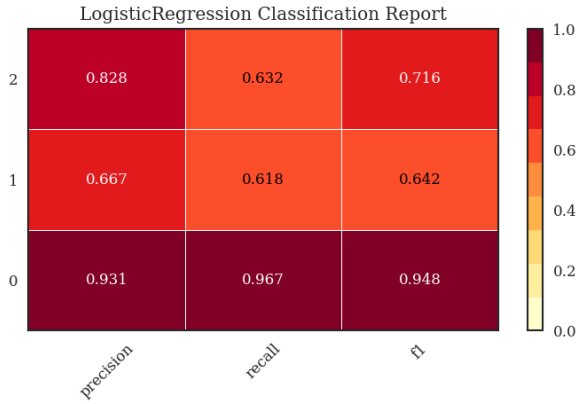
Matrix 2. A matrix of precision, recall, and f1 score for the outcome of the grid search cross-validation for the SVM model for classifying CTG data into three fetal health statuses.

The grid search cross-validation was used to optimize parameters C and gamma. The C parameter determines how much penalty there is on observations classified incorrectly, or error. When C is low, you may have more points incorrectly classified, but you are less likely to overfit the data. The Gamma parameter is the range for which points are considered in creating the dividing vector between classifications. A high gamma considers points close to the border of classification. Grid search increased precision, recall, and the f1 score for all the fetal health statuses. This technique was very successful because none of the scores had to decrease to increase other scores.

The model was made better overall by grid search. Fetal health status two still had the lowest scores and this would have to be improved for the model to be used by physicians or this group would be the most instruction from a physician.

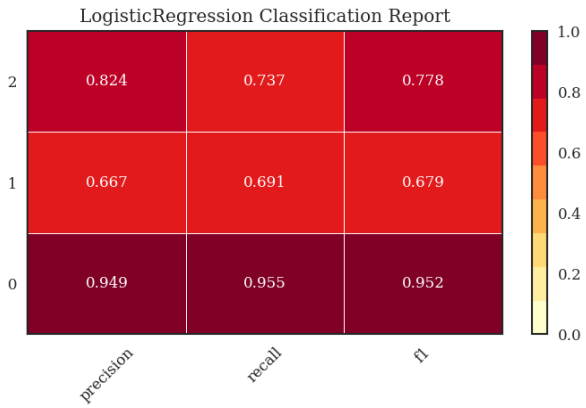
B. Multinomial Logistic Regression

The initial Multinomial Logistic model has 88 % accuracy in total. Because of the class imbalance in the Normal category, the model has higher precision, recall, and f1-score in this category compared to pathological or suspect categories. Matrix 3 shows the initial Multinomial Logistic model's precision, recall, and f1 score. Here, 0 represents normal, 1 is suspect, and 2 is pathological.



Matrix 3. A matrix of precision, recall, and f1 score for the first Multinomial Logistic Regression model in this study for classifying CTG data into three fetal health statuses.

To improve the initial accuracy of the model, we performed a grid search. In grid search, we tried different models with L1 and L2 regularization penalties, different solvers such as newton-cg, lbfgs, and liblinear, and different C parameters, which affect regularization strength. All precision, recall, and f1-score metrics improved for all three categories after the grid search. In the end, we reached 90 % overall accuracy. Matrix 4 displays the results after the grid search.



Matrix 4. A matrix of precision, recall, and f1 score for the outcome of the grid search cross-validation for the Multinomial Logistic Regression model for classifying CTG data into three fetal health statuses.

C.Gradient Boosting Models and efforts towards Model Explainability

We report here the performance of the XG-Boost model however the approach is really model agnostic to other flavors that we tried in this work. This model was trained using 5-fold cross-validation and in this section, we try and explain the importance of the features using the native-feature importance tool within XG-boost. Briefly, ensembles of decision tree methods like gradient boosting can automatically provide estimates of feature importance from the trained predictive model. The importance provides a score that indicates how useful or valuable each feature proved to be in the construction of the boosted decision trees within the model [21]. "The more an attribute is used to make key decisions with decision trees, the higher its relative importance"[21].

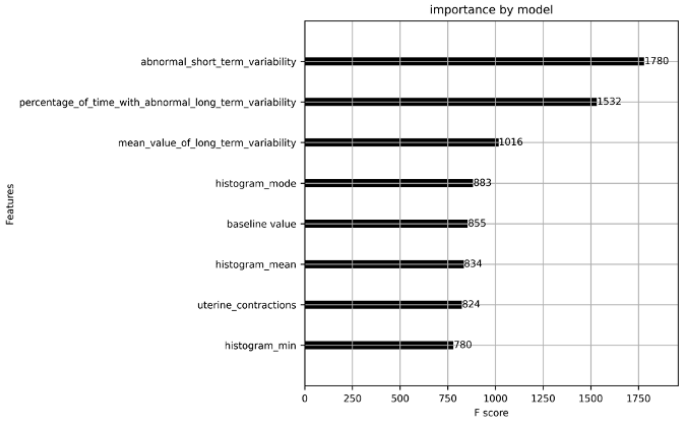


Fig 6. Exploring Feature Space with Classic Feature Attributions based on global model importance.

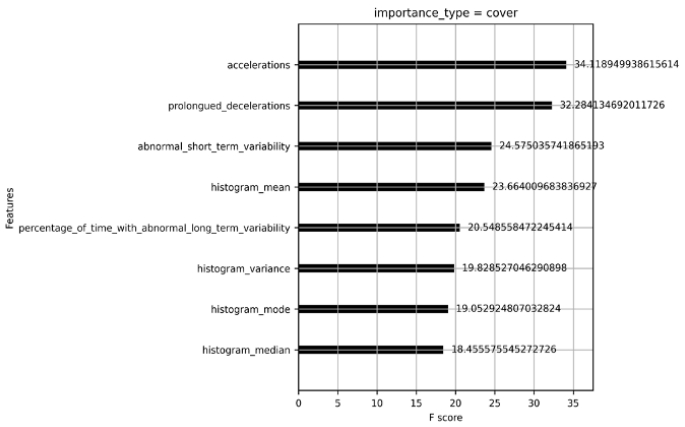


Fig 7. Exploring Feature Space with Classic Feature Attributions based on model importance "cover".

XGBoost reported the best Accuracy of 0.86 and F1-macro score of 0.79 on the test set. The XG-Boostmodel was trained on an 80-20 train-test split with hyper-parameters eta and sub-sample parameter tuned on a further 90-10 train-validation split across 5-fold cross-validation.

We plot various criteria [22] based on which importance of a feature is calculated, Fig. 6 plots the importance based on # of occurrences of features in the trees. It represents "total model importance based on the percentage representing the relative number of times a particular feature occurs in the trees of the model".

Fig. 7 to the right plots the importances based on the coverage criteria, which is the "relative quantity of observations explained by a particular feature" [22].

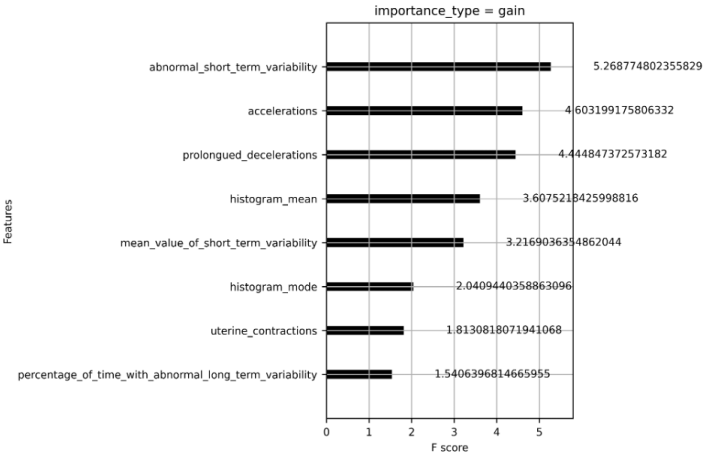


Fig 8. Exploring Feature Space with Classic Feature Attributions based on model importance "gain"

Fig. 8 represents feature importances based on improvement in accuracy brought by a feature to the branches it is on. However, a key failure to all the analyses presented so far is that the relative importance of each feature brought about by the same XG-boost model, in this case, trained and tuned on the training and validation set in fact contradict each other. We can observe the top-3 important feature in each case such as their rank and magnitude are not conserved across the three plots. The most important feature in the plot (a) does not necessarily coincide with the others. This motivates us to further investigate what features have really played a role in having the model achieve an accuracy of 0.897 and F1 of 0.796. Since it is not only important that we successfully predict with reasonable accuracy but explain the predictions of the ML model. All the above-stated facts motivate the use of SHAP values since they come with consistency guarantees (meaning they will order the features correctly and can also explain both local heterogeneity and global convergence).

D. SHapley Additive exPlanations (SHAP) for Explainable AI

We address the feature-importance concern with an approach from an econometric game theory known as SHAP [4-6]. The approach compares the influence of each variable against the other characteristics by executing a large number of predictions. On the whole feature set, including histogram and other features, we ran the SHAP analysis of the XGBoost model inspired by [5].

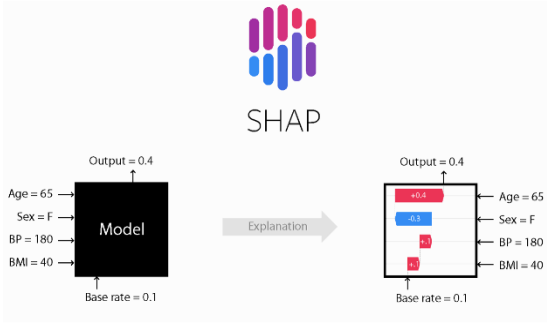
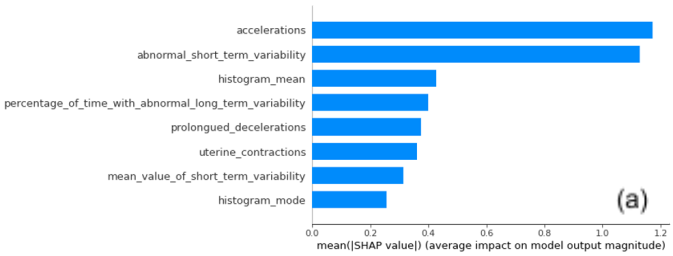


Fig 9. SHAP is a game-theoretic technique to explain any machine learning model's output. It uses the "traditional Shapley values from game theory and their associated expansions to correlate optimum credit allocation with local explanations". Figure courtesy: SHAP (SHapley Additive exPlanations)[5]

In a nutshell, SHAP as shown in this block diagram can explain the "prediction of a sample instance by computing the contribution of each feature to the target using optimal credit allocations"[5]. As an example, let us consider the model in question at a particular prediction instant that takes in the parameters on the left such as {Age, Sex, BP, BMI} and produces an Output labeled above. The SHAP framework computes the contribution of each feature that helps in generating the target shown to the right in Fig. 9. Thus, our XGBoost model can help discover the interaction among the said features to generate a prediction of the target class a fetus belongs. Our motivation to use the SHAP framework to explain predictions of our GBM model stems from the fact that "decision tree models are a powerful tool to discover the interaction among independent variables (features)" [23]. So SHAP explainability can help unravel some of the decisions made by the classifier [7].

E. Exploring Feature Space with SHAP



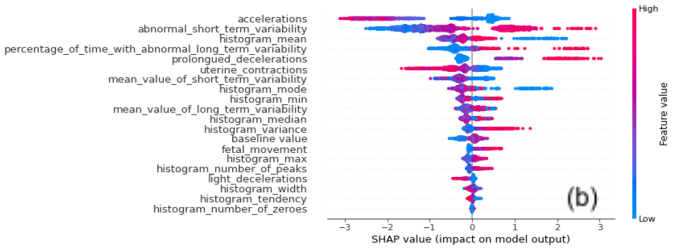


Fig 10. Feature summary bar plot (a) and per-sample importance plot (b).

The upper plot in Fig. 10 (a) shows the feature significance plot representing the average contribution of all the samples across the dataset. The corresponding SHAP values for all samples are shown in the density scatter plot in the lower half (b). The color bar in the lower plot represents the relative magnitude of the feature value. Together with the feature magnitude in the color bar and SHAP magnitude in the x-axis, we can allude to whether a higher feature value drives a greater impact on model output and vice versa. It's worth noting that while the acceleration feature has a greater overall model impact than the abnormal-short-term feature in figure (a), abnormal-short-term has a greater influence in those cases where it counts. In other words, the abnormal-short-term has a higher impact on a few predictions, whereas acceleration has a lower impact on all predictions.

F. Discovering Interactions between features with SHAP

The SHAP dependency charts in the next few sub-figures (a-h) in Fig. 11 demonstrate how a single characteristic feature affects the whole dataset. Across all samples, we plot the feature value versus, the SHAP value for a particular feature. SHAP dependence plots account for feature interaction effects and are only defined in data-supported parts of the input space. Interaction effects cause the vertical dispersion of SHAP values at a single feature value [24], and another feature is chosen for coloring to emphasize probable interactions. Plots (a) through (h) in Fig. 11, shown below represent the SHAP magnitudes against the corresponding feature values and we paint the pixels in the plots with normalized feature values of the next important feature from the SHAP feature importance. For example in (a) we plot the feature value for accelerations on the x-axis with the SHAP value for accelerations on the y-axis and the data tips are painted by the feature value next-important in SHAP plots in this case histogram_median. This is repeated for the top-6 features. Note that feature magnitudes are normalized when used for SHAP analysis, so the color bar in all the plots is bound within [0-1].

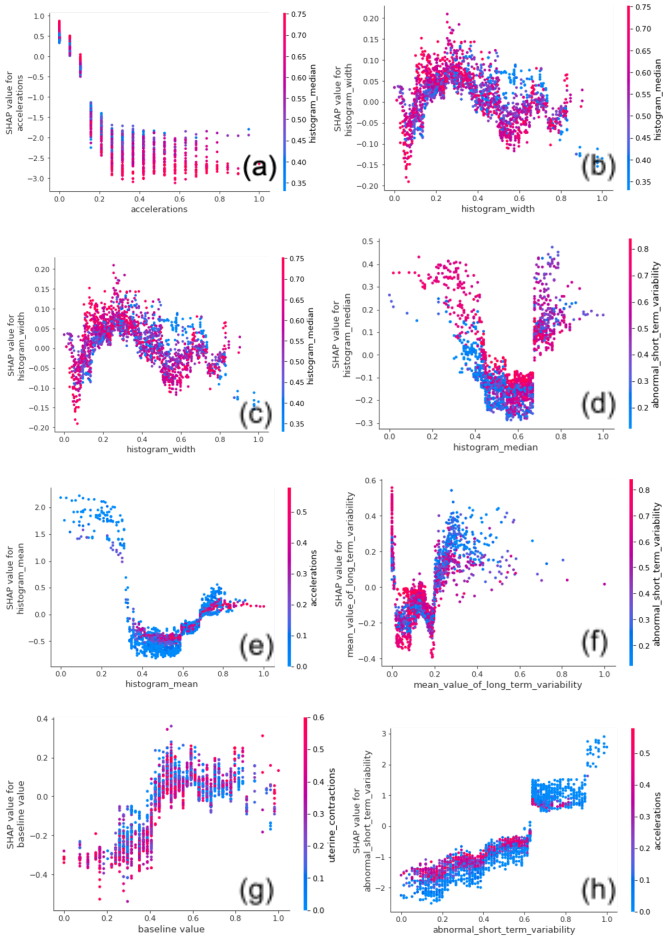


Fig 11. Interaction plot of top-6 SHAP (a-h) features painted with its corresponding top SHAP attribute.

G. Calculating Bayesian Posterior Inference from SHAP Analysis

Next, we turn our attention to a supervised clustering approach and note that our SHAP feature importance magnitudes basically represent relative weights or probabilities of the features. Thus, the cumulative sum should represent the cumulative probability of favoring or rejecting a particular hypothesis (predicted state). Here we show the performance of 2 clustering algorithms using PCA top (a) and t-SNE to the bottom (b), which is performed on the SHAP magnitudes of each sample, the color bar represents the log-odds ratio of favoring a prediction of the fetal health being diagnosed as pathological. We observe that t-SNE embedding on SHAP shows distinctively the presence of 3 clusters a largely dominant blue representing normal, a mild pink showing that predictions are suspect, and a small cluster of high log probability of being in the pathological state. The results are consistent with the distribution of our classes that we saw in Fig. 1.

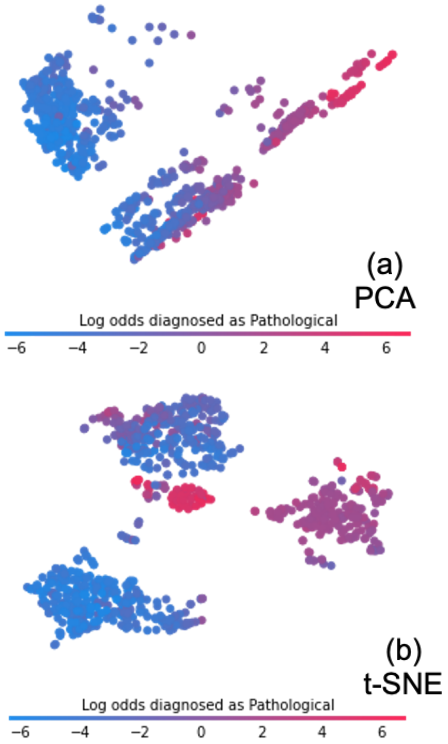


Fig 12. PCA and t-SNE on the SHAP values to extract PCA and t-SNE SHAP embedding are shown in (a) and (b) respectively. We select the top-2 embeddings that represent > 80% variance in SHAP magnitudes. Clusters are represented in the embedding space colored by SHAP magnitude values at each sample. Colorbar represents the log-odds ratio of a person being staged as being diagnosed in a pathological state.

VI.CLUSTERING APPROACHES

A.Supervised Clustering with SHAP weights

Continuing the clustering approach that we saw earlier, here we paint the clusters using SHAP-2D t-SNE embedding based on the top-3 SHAP features to pictorially discover how each of which might have played a role in the prediction. We observe that based on just a single feature, it is not possible to discern any clear clusters within the embedding space, which reinforces the fact that predictions using the SHAP offer a more effective structure than just a means to select important features.

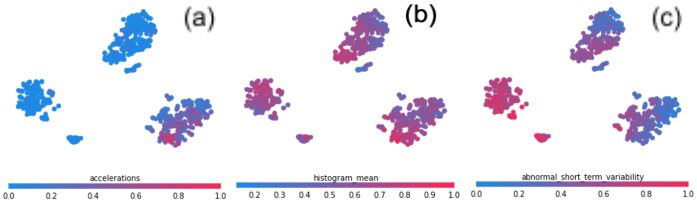


Fig 13. Clusters based in the embedding space based on the top-3 important SHAP feature. (a)2D-TSNE SHAP Embedding, colormap based on: “accelerations”. (b) 2D-TSNE SHAP Embedding: colormap based on: “histogram-mean”. (c) 2D-TSNE SHAP Embedding: colormap based on: “abnormal-short-term-variability”.

B. Clustering with Uniform Manifold Approximation and Projection (UMAP)

Uniform Manifold Approximation and Projection (UMAP) [8] is a variant of the dimension reduction approach that operates similar to t-SNE and is used in this section for visualization as well as non-linear dimension reduction.

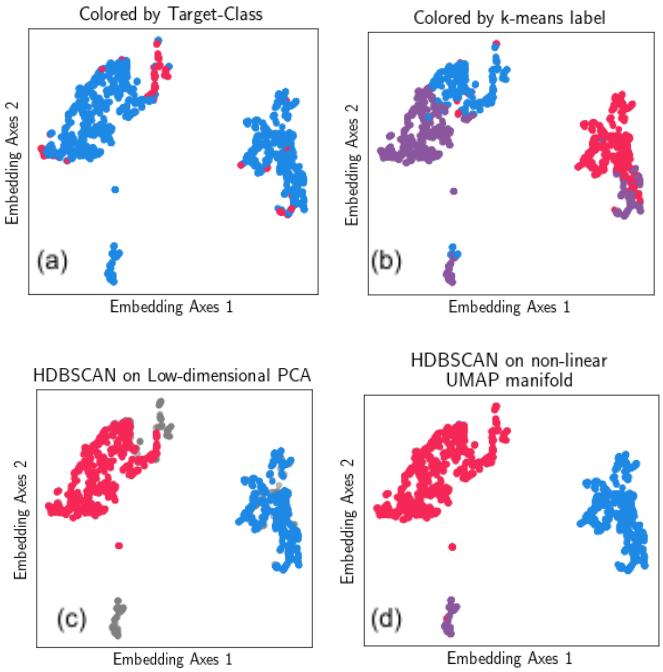


Fig 14. Clustering results with U-MAP embedding with various configurations. Embeddings are represented by (a) colored points by target class (b) Naive K-Means embedding with painted pixels by K-Means class membership. Advanced density-based clustering (c) on PCA and (d) tuned UMAP non-linear manifold.

U-MAP assumes a key factor with regard to the distribution of the data predominant of which is the fact that the data that we are trying to cluster is distributed on a locally connected Riemannian manifold with the Riemann metric being either locally constant or some form of its approximation [9]. In this work, we use U-MAP in conjunction with traditional clustering-based approaches [10] like K-Nearest Neighbors Fig. 14 (a) and (b) alongside a lower-dimensional embedding with PCA (c) followed by finally a locally connected non-linear U-MAP manifold with a density-based clustering approach called Density-Based Clustering Based on Hierarchical Density Estimates (HDBSCAN)[12] to boost the performance of the density clusters. There are many similarities as well as differences that separate UMAP and other techniques like t-SNE while clustering the data.

The most apparent difference is that, unlike t-SNE, UMAP does not fully retain density. However much like t-SNE, UMAP can generate false in cluster partitions, thereby producing much finer grouping than that is present in the original data. Hence there are compelling reasons to utilize UMAP as a clustering preprocessing stage [10]. Like in any clustering method, we explore the tunable hyperparameters and evaluate the clusters that emerge to see if they can be validated by the original label distribution in the data. Using UMAP, we reduce the data to two dimensions shown in Fig 14. We first look at the data (a), which is colored by the target class represented. Now we use a traditional clustering with K-means and paint UMAP embedded data by cluster membership (b). We explore the HDBSCAN method, which is known to extend beyond the spherical cluster assumption. For an advanced HDBSCAN method, we reduce the dimensionality of the data down to 12 dimensions shown in Fig 14 (c) with a PCA reduction since it was seen to cover a majority of the variance. Since HDBSCAN is known to scale poorly with increasing dimensionality of the data [10]. Subsequently, we adjust the clustering settings in order to eliminate noise patterns in the data and bring out the original clusters, we do so by adjusting the `min_dist` and `n_neighbors` and again perform HDBSCAN with tuned parameters. We observe the general global structure is still retained in (d), however, the now that parameters are tuned to secure clusters that provide larger gaps between the clusters, we observe that some of the noise in the upper-left corner in Fig. 14 (c) is removed in Fig.14 (d). Additionally, we also observe that the final tuned UMAP embedding with HDBSCAN clustering performs best in terms of metrics discussed in the next section listed in TABLE 1: since we are not forced to tolerate the relative lack of density in 21-dimensional space which helps the HDBSCAN reveal cleaner clusters.

VI.CONCLUSION

The XG-Boost models performed the best Accuracy of 0.897 and an F1-macro score of 0.796 on the test set. Metric scores are marginalized based on class imbalance with “balanced-weighting” which means that weights are inversely proportional to sample distribution. We explored native XG-Boost feature importances, however, they were shown to lack congruity within the same model and its parameters. In order to address the issue, the model predictions are explored with a game-theoretic framework known as SHAP which helps understand interactions, latent effects, and cluster formations.

SVM has an overall accuracy of 0.93, Multinomial Logistic Regression has 0.90 accuracies and 0.79 F1-Score. Overall for our classification models, SVM performed better than the

multinomial logistic regression and the XG-Boost by the metrics we discussed in Table 1.

TABLE 1: Performance Metrics of deployed models used in this study.

Model	Accuracy	f1-score
SVM	0.93	1-0.96 2-0.75 3-0.84
Multinomial Logistic Regression	0.90	0.79
XG-Boost	0.897	0.796
Traditional UMAP + HDBSCAN	adjusted_rand_score: 0.0392 adjusted_mutual_info_score: 0.0294 data_assigned_to clusters: 0.83	
Enhanced UMAP + HDBSCAN	adjusted_rand_score: -0.0024 adjusted_mutual_info_score: 0.0136 data_assigned_to clusters: 1.0	

In a second approach aside from classification and regression models, we also demonstrated the way how U-MAP learns the manifold structure within the CTG data and is able to represent a low-dimensional embedding. The lower-dimensional embedding is capable of preserving the topological structure of the computed manifold [18]. We report the best performance of our clustering method by evaluating on the basis of metrics of adjusted_rand_score and mutual_info_score and observe that the last two rows of classifiers in Table I, namely traditional U-MAP+HDBSCAN and Enhanced U-MAP+HDBSCAN allocates 83% and 100% data to their respective clusters respectively.

The results from the study indicate that the pipelines such as these could be applied to biophysical datasets for the classification of pathologies and potentially be useful to medical researchers who can improve the dataset by adding administered drugs.

ACKNOWLEDGMENT

The authors acknowledge the reviews from the instructor Prof. Cantay Caliskan and the peer-reviewers comments that helped address the modeling approaches.

REFERENCES

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

[2] Deressa, Tadele Debisa, and Kalyani Kadam. "Prediction of fetal health state during pregnancy: a survey." International Journal of Computer Science Trends and Technology (IJCST) 6.1 (2018)

[3] "Fetal Health Classification" Retrieved on April 16, 2022 from,https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification?select=fetal_health.csv.

[4] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Advances in Neural Information Processing Systems (2017).

[5] Explainability of Tree-based models using SHAP

[6] **SHAP** (SHapley Additive exPlanations)

[7] Interpretable Machine Learning "A Guide for Making Black Box Models Explainable" by Christoph Molnar.

[8] McInnes, L, Healy, J, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, ArXiv e-prints 1802.03426, 2018.

[9] UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

[10] Using UMAP for Clustering

[11] Akhtar, F., Li, J., Azeem, M., Chen, S., Pan, H., Wang, Q., & Yang, J. J. (2019). Effective large for gestational age prediction using machine learning techniques with monitoring biochemical indicators. The Journal of Supercomputing, 76, 1–9.

[12] HDBSCAN clustering algorithm

[13] Mehbodniya, A., Lazar, A. J. P., Webber, J., Sharma, D. K., Jayagopalan, S., K, K., Singh, P., Rajan, R., Pandya, S., & Sengan, S. (2021). Fetal health classification from cardiotocographic data using machine learning. Expert Systems, e12899. https://doi.org/10.1111/exsy.12899

[14] Nannapaneni, Rajasekhar. "Human Fetus Health Prediction Using Decision Tree." (Feb 2020) https://education.dellemc.com/content/dam/dell-emc/documents/en-us/2019KS_Nannapaneni-Human_fetus_health_prediction_using_decision_tree.pdf

[15] 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

[16] Kermany, D. S., Goldbaum, M., & Cai, W. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5), 1122–1131.

[17] Stoean, R., & Stoean, C. (2013). Modeling medical decision-making by support vector machines, explaining by rules of evolutionary algorithms with feature selection. Expert Systems with Applications, 40(7), 2677–2686.

[18] How Exactly UMAP Works: And why exactly it is better than tSNE, Nikolay Oskolkov, Towards Data Science.

[19] Sklearn Rand index adjusted for chance

[20] Sklearn: Adjusted Mutual Information between two clusterings.

[21] Feature Importance and Feature Selection With XGBoost in Python, by Jason Brownlee on August 31, 2016 in XGBoost

[22] The Multiple faces of 'Feature importance' in XGBoost

[23] Feature Interaction Constraints

[24] shap.dependence_plot