Fetal Health Risk Prediction and Patient Clustering for Personalized Maternal Care: Machine Learning Final Project Report

Devadarshini Pazhanivel Thenmozhi, Sharon Jennifer Justin Devaraj
Khoury College of Computer Sciences
Northeastern University, Boston, USA
pazhanivelthenmozh.d@northeastern.edu, justindevaraj.s@northeastern.edu
GITHUB

***Abstract*—Fetal health monitoring plays a crucial role in preventing complications during pregnancy through early diagnosis. This project presents a comprehensive machine learning framework to predict fetal health risk using cardiotocography data. A continuous risk score is generated through Principal Component Analysis and predicted using Ridge Regression, Random Forest, and XGBoost. SHAP is used to interpret model predictions and highlight the most influential features contributing to fetal risk. Unsupervised clustering techniques, including KMeans and Hierarchical Clustering, are applied to group patients into low, moderate, and high-risk categories, supported by UMAP and PCA visualizations. The models achieved high accuracy with interpretable outputs, showing that this approach could help doctors make better decisions in assessing fetal health. This report outlines the objectives of the project, explains the implementation of various machine learning methods, and reflects on the outcomes and insights gained through the analysis.**

***Index Terms*—Fetal health, Machine learning, KMeans, Ridge regression, Random Forest, XGBoost, SHAP, PCA, Clustering, Hierarchical Clustering, Risk prediction.**

I. Introduction

Fetal health assessment is an important part of prenatal care, as it helps detect signs of fetal distress at an early stage. Cardiotocography (CTG) is a widely used, non-invasive method that tracks fetal heart rate and uterine contractions to assess well-being. However, interpreting CTG results can be subjective and often depends on a doctor's experience, which may lead to inconsistent diagnoses.

To overcome this challenge, machine learning techniques offer a reliable and data-driven way to analyze CTG data. ML models can detect patterns across multiple clinical features and find patterns that may not be obvious by just looking at the data. While traditional classification models assign fetal states into fixed risk categories, this can sometimes miss the full range of possible conditions.

This project takes a more detailed approach by turning the original class labels into a continuous fetal risk score using Principal Component Analysis (PCA). The risk score is then predicted using regression models such as Ridge Regression, Random Forest, and XGBoost. Each model is built in both standard and custom forms to better understand how the algorithms work and compare their performance.

To make the predictions more explainable, SHAP (SHapley Additive exPlanations) is used to show which features contribute most to each prediction. Unsupervised clustering methods like KMeans and Hierarchical Clustering are also applied to group patients based on their risk scores. Together, this combination of regression and clustering forms a strong and explainable system for fetal health risk prediction that could support clinical decision-making.

II. Data

The dataset used in this project is the Fetal Health Dataset, publicly available on Kaggle. It contains 2,126 clinical records obtained from cardiotocographic (CTG) exams, each labeled with a fetal health class. There are 22 features such as baseline heart rate, accelerations, decelerations, variability measures, and histogram-derived metrics. The target variable fetal health indicates the class label (1 - Normal, 2 - Suspect, 3 - Pathological).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   baseline value                                          2126 non-null   float64
 1   accelerations                                           2126 non-null   float64
 2   fetal_movement                                          2126 non-null   float64
 3   uterine_contractions                                    2126 non-null   float64
 4   light_decelerations                                     2126 non-null   float64
 5   severe_decelerations                                    2126 non-null   float64
 6   prolongued_decelerations                                2126 non-null   float64
 7   abnormal_short_term_variability                         2126 non-null   float64
 8   mean_value_of_short_term_variability                    2126 non-null   float64
 9   percentage_of_time_with_abnormal_long_term_variability  2126 non-null   float64
 10  mean_value_of_long_term_variability                     2126 non-null   float64
 11  histogram_width                                         2126 non-null   float64
 12  histogram_min                                           2126 non-null   float64
 13  histogram_max                                           2126 non-null   float64
 14  histogram_number_of_peaks                               2126 non-null   float64
 15  histogram_number_of_zeroes                              2126 non-null   float64
 16  histogram_mode                                          2126 non-null   float64
 17  histogram_mean                                          2126 non-null   float64
 18  histogram_median                                        2126 non-null   float64
 19  histogram_variance                                      2126 non-null   float64
 20  histogram_tendency                                      2126 non-null   float64
 21  fetal_health                                            2126 non-null   float64
dtypes: float64(22)
memory usage: 365.5 KB
```

Fig. 1. Overview of the dataset

The classification of fetal health depends critically on the baseline value. Deviations from the baseline value could be a sign of fetal discomfort or impairment, necessitating more

testing or medical attention. It is crucial to monitor the baseline value accurately and consistently in order to control and assess fetal health.

The existence or absence of accelerations or decelerations, as well as their frequency and length, may all be seen in histograms of fetal heart rate data. Histograms can also be used to spot patterns and trends in the data. Healthcare professionals can learn more about fetal health status by examining the histogram's features.

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations
count	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000
mean	133.303857	0.003178	0.009481	0.004366	0.001889	0.000003	0.000159
std	9.840844	0.003866	0.046666	0.002946	0.002960	0.000057	0.000590
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000
75%	140.000000	0.006000	0.003000	0.007000	0.003000	0.000000	0.000000
max	160.000000	0.019000	0.481000	0.015000	0.015000	0.001000	0.005000

Fig. 2. Dataset Description

The dataset takes into account a lot of features regarding the fetal health.

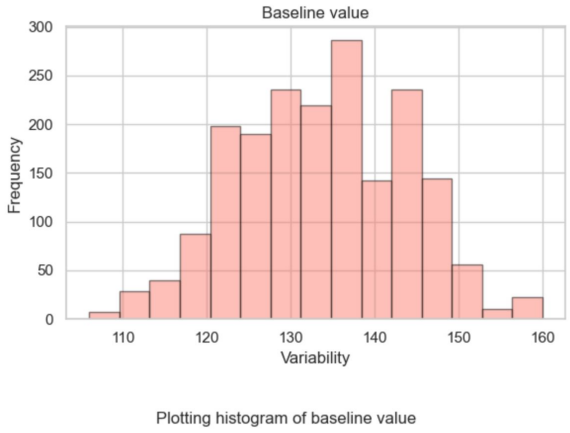

Fig. 3. Baseline Value Plot

The heart rate shows normal variation mostly centered around 130 - 140 bpm.

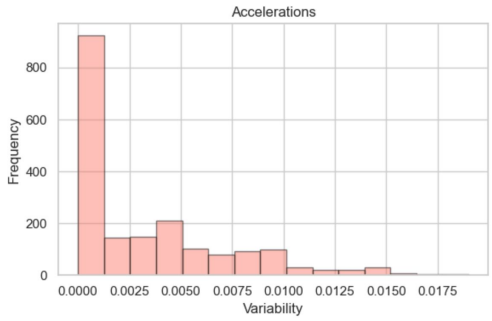

Fig. 4. Histogram of Accelarations

Fetal accelerations are very rare or low in most cases.

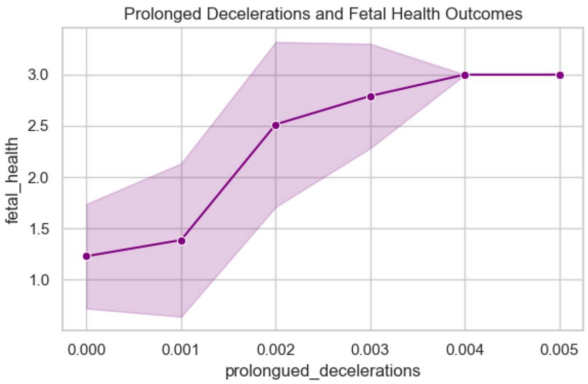

Fig. 5. Prolongued deceleration vs Fetal Health

The fetus is most likely to be at risk when it has high extended decelerations.

III. Related Work

A. Classification–Supervised Learning

J. Li and X. Liu (2021), "Fetal Health Classification Based on Machine Learning"
This study proposed a machine learning framework for fetal health classification using CTG data. Multiple classifiers were used including Gradient Boosting, CatBoost, LightGBM, Cascade Forest, and AdaBoost. Their ensemble model based on soft voting achieved the highest accuracy of 95.9%, outperforming individual classifiers and showcasing the effectiveness of ensemble learning in fetal health prediction.

J. Piri and P. Mohapatra (2019), "Exploring Fetal Health Status Using an Association Based Classification Approach"
This work explored associative classification methods like CBA-M1 and CBA-M2 alongside Random Forest and XGBoost. The pruned version of CBA-M2 obtained the best accuracy of 83.54%. The study focused on combining rule-based learning and feature selection to improve efficiency without compromising accuracy.

Z. Cömert, A. Şengür, Ü. Budak, and A. F. Kocamaz (2019), "Prediction of Intrapartum Fetal Hypoxia Considering Feature Selection Algorithms and Machine Learning Models"
This research evaluated classification models such as ANN, KNN, Decision Tree, and SVM for fetal hypoxia detection. SVM performed the best with an accuracy of 88.58%. The study emphasized the role of feature selection in enhancing model accuracy and reducing computational complexity in clinical decision-making scenarios.

B. Regression–Continuous Risk Scoring

H. J. Chang et al. (2022), "Machine Learning Model for Classifying the Results of Fetal Cardiotocography"
This study developed a machine learning model to classify CTG results as normal or abnormal, using over 17,000 records

labeled by physicians. The model achieved an AUROC of 0.89, indicating strong performance in fetal health classification. It highlights how ML-based systems can enhance clinical decisions with objective insights from CTG data.

O. C. Olayemi and O. O. Olasehinde (2024), "Machine Learning Prediction of Fetal Health Status from Cardiotocography Examination in Developing Healthcare Contexts"
This work focused on predicting fetal health using CTG data in low-resource healthcare contexts. Models like Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, CatBoost, and XGBoost were tested on features such as fetal heart rate and uterine contractions. Results showed that ML models can successfully aid early identification of high-risk pregnancies, promoting timely intervention.

C. Hybrid & Clustering Risk Stratification

J. Feng, J. Liang, Z. Qiang, et al. (2023), "A Hybrid Stacked Ensemble and Kernel SHAP-Based Model for Intelligent Cardiotocography Classification"
This paper proposed a hybrid model combining a stacked ensemble (SVM, XGBoost, Random Forest) with a neural network meta-learner and Kernel SHAP for interpretability. The approach achieved high accuracy on CTG datasets and identified key features like accelerations and abnormal short-term variability. It demonstrates the effectiveness of integrating ensemble models with explainability for fetal monitoring.

IV. IMPLEMENTATION

A. Data Preprocessing and Feature Engineering

The dataset consisted of 2,126 complete entries with no missing values. All features were confirmed to be numeric. Outliers were detected and removed using the Interquartile Range (IQR) method, with main focus on skewed features such as histogram variance, histogram median, and histogram mean.

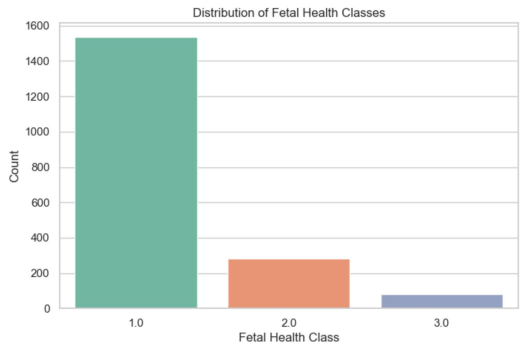

Fig. 6. Distribution of Fetal Health Class

Class imbalance is a significant issue in this dataset. Most samples fall into class 1. This imbalance could lead to biased model training and poor generalization of minority classes. This was addressed by dividing the data into training and testing sets using stratified shuffle splitting, which keeps the class distribution constant between the two subsets. This ensured a more accurate and reliable evaluation of model performance, particularly when detecting rare cases.

All numerical features were normalized using StandardScaler to ensure uniformity. Feature correlation analysis was performed using a correlation matrix and a heat map. This analysis revealed that some features were highly correlated with others and did not provide unique information to the model. Histogram mode and histogram median showed strong correlation with other histogram-based features such as histogram mean. These features were removed from the dataset to prevent overfitting.

B. Risk Score Engineering

Principal Component Analysis (PCA) was used to create a fetal risk score to convert the multiclass fetal health classification into a continuous target suitable for regression. PCA was applied to extract the first principle component, which represents the best combination of features. The scores were scaled to a range of 0 - 100 using MinMax Scaler to create a continuous risk score. This engineered risk score served as the regression target for all models.

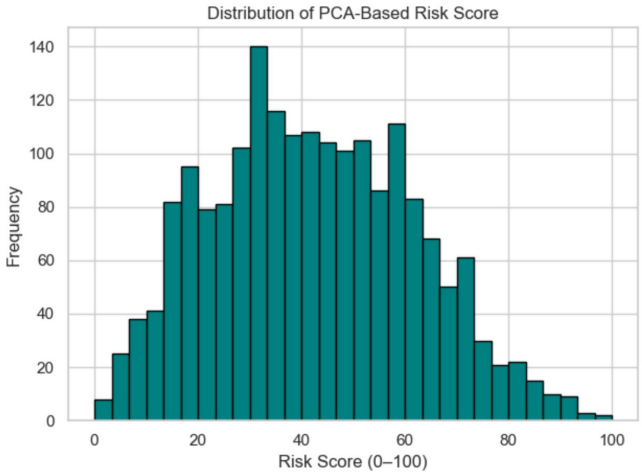

Fig. 7. PCA-based fetal risk scores

C. Regression Models

All of the regression and clustering models used in this project were implemented from scratch in Python. This gave us a deeper understanding of how each algorithm works internally, including regularization, clustering, and gradient boosting mechanisms.

To validate the performance of these custom implementations, standard models from scikit-learn libraries were also used. Comparing them helped check the accuracy of the custom models and ensured they gave meaningful results. This way we ensure that our models produce reliable predictions for real-world use.

1) Standard Ridge Regression: The standard Ridge Regression model was implemented using the Ridge class from scikit-learn with L2 regularization to address multicollinearity. The model was trained on scaled features and evaluated using MAE, RMSE, and R^2. SHAP explainability was used to assess feature influence.

2) Ridge Regression: Ridge Regression was implemented by modifying the basic linear regression formula to include L2 regularization. This helps prevent overfitting by shrinking the weights of less important features. A bias term was added, and the final weights were calculated using matrix operations. The model was trained to predict continuous fetal risk scores, and outputs were clipped between 0 and 100 to keep them within the expected range. The model's performance was nearly equivalent to the standard implementation.

3) Standard Random Forest Regressor: The standard Random Forest model was implemented using scikit-learn's RandomForestRegressor in combination with GridSearchCV to perform hyperparameter tuning. Grid search was run with 5-fold cross-validation to identify the best combination of parameters based on mean squared error. The best-performing model from this search was used for final predictions.

4) Random Forest Regressor: The Random Forest model was built by combining multiple decision trees trained on randomly selected subsets of the data using bootstrapped sampling. Each tree was trained independently, and they all saw slightly different versions of the dataset. Thus, the ensemble became more diverse and less likely to overfit. Predictions were made by averaging the outputs of all individual trees. This model demonstrated how ensembling reduces variance while achieving comparable results to the standard model.

5) Standard XGBoost Regressor: The model was implemented using the XGBRegressor from the XGBoost library. Hyperparameter tuning was done using GridSearchCV, testing multiple combinations of parameters. Once the best hyperparameters were identified, the final model was trained on the full training set and used to predict fetal risk scores.

6) XGBoost Regressor: XGBoost started with an initial prediction, and then sequentially added new trees that learned to predict the residuals. Each tree tried to correct the mistakes of the previous one, and a learning rate was used to control how much each new tree could influence the final prediction. Over time, this iterative process built a strong, accurate model that was able to learn subtle patterns in the data.

D. Model Interpretation using SHAP

To make the predictions from both standard and custom models easier to understand, SHAP (SHapley Additive exPlanations) was used. SHAP helped to show how much each feature contributed to a model's prediction for a given case. For every regression model it was used to explain both overall feature importance and individual predictions. Visual tools like summary plots, bar charts, and waterfall plots made it clear which features had the biggest impact on the risk scores. SHAP was also applied to clustering, helping explain how patients were grouped into low, moderate, and high-risk categories. This made the entire prediction and clustering process more easier to interpret from a clinical perspective.

E. Clustering for Fetal Risk Grouping

1) KMeans (Standard and Custom): To identify natural groupings in fetal risk levels, KMeans clustering was applied. Both standard KMeans from scikit-learn and a custom implementation were used. UMAP was used to reduce feature dimensionality before clustering. The reduced data was clustered into three groups and each cluster was mapped to a risk category (Low, Moderate, or High) based on the average risk score of that group. A color-coded visualization was created to show how patients were separated in the UMAP space. Additionally, KMeans was also applied to the predicted risk scores, clustering patients solely based on their predicted score distribution.

2) Hierarchical Clustering (Standard and Custom): Hierarchical clustering was applied to group patients into fetal risk categories using both standard and custom implementations. UMAP was first used to reduce feature dimensions, followed by Agglomerative Hierarchical Clustering with Ward's linkage to form three clusters. These were mapped to Low, Moderate, and High risk based on average risk scores. Additionally, hierarchical clustering was applied directly to the predicted risk scores. A dendrogram was generated to visualize the hierarchy and cluster separations. The clusters' separation was demonstrated by dendrograms.

Clustering performance was measured using Silhouette Score, Davies-Bouldin Index, and Dunn Index to evaluate cohesion and separation.

F. Prediction Mapping and Risk Group Assignment

After generating the predicted continuous risk scores from all regression models, the outputs were processed to assign each patient into a risk group—Low, Moderate, or High. This was achieved using a rule-based mapping function based on score thresholds. Similarly, clustering algorithms like KMeans and Hierarchical Clustering were applied to both the UMAP-reduced feature space and to the predicted scores. Each cluster was interpreted and labeled as a risk group based on the mean risk score within that cluster.

V. Results and Discussion

A. Regression Performance Comparison

Table I presents a comparative evaluation of all six regression models based on MAE, RMSE, and R^2 score. These metrics were calculated on the held-out test set using the continuous PCA-based risk score as the target.

Ridge Regression ended up performing the best, because the target we were trying to predict : the continuous risk score was created using PCA which applies a linear projection of features. Since Ridge is also a linear model, it naturally fit

TABLE I
PERFORMANCE METRICS OF REGRESSION MODELS

Model	MAE	RMSE	R^2 Score
Standard Ridge Regression	0.10	0.12	1.0000
Custom Ridge Regression	0.20	0.23	0.9999
Standard Random Forest	1.55	2.20	0.9870
Custom Random Forest	1.82	2.56	0.9824
Standard XGBoost	1.27	1.67	0.9924
Custom XGBoost	1.52	2.04	0.9888

this setup really well. While models like Random Forest and XGBoost are designed to handle more complex, nonlinear relationships, they didn't surpass Ridge in this case due to the relatively linear nature of the target. Still, both models gave excellent results and showed strong predictive power.

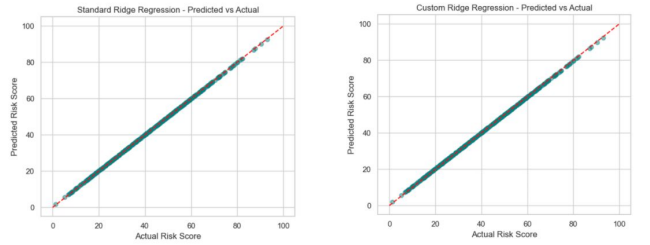

Fig. 8. Predicted vs Actual Scatter Plots - Ridge

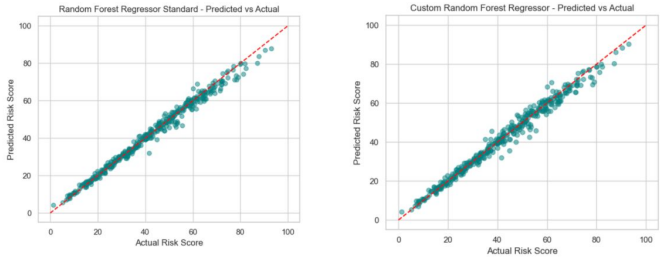

Fig. 9. Predicted vs Actual Scatter Plots - Random Forest

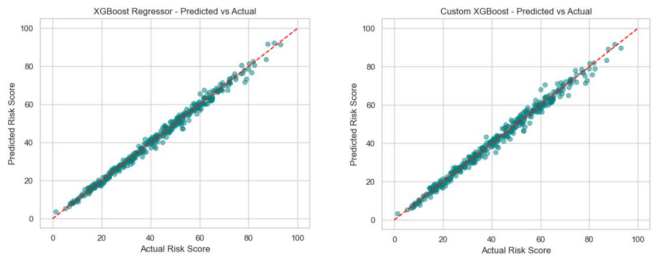

Fig. 10. Predicted vs Actual Scatter Plots - XGBoost

B. Clustering Results

Unsupervised clustering techniques were applied to the UMAP-reduced feature space and predicted risk scores. Table II summarizes the clustering evaluation metrics.

TABLE II
CLUSTERING EVALUATION METRICS

Method	Silhouette	Davies-Bouldin	Dunn Index
Standard KMeans	0.4369	0.8582	0.0002
Custom KMeans	0.4408	0.8412	0.0001
Standard Hierarchical	0.5329	0.5457	0.0005
Custom Hierarchical	0.5069	0.5078	0.0005

The evaluation metrics show that hierarchical clustering outperformed KMeans across all three indices. Standard hierarchical clustering achieved the highest silhouette score (0.5329) and the lowest Davies-Bouldin index (0.5457), suggesting well-defined and well-separated clusters. It also matched custom hierarchical clustering in terms of the Dunn index (0.0005), reflecting strong intra-cluster compactness and inter-cluster separation.

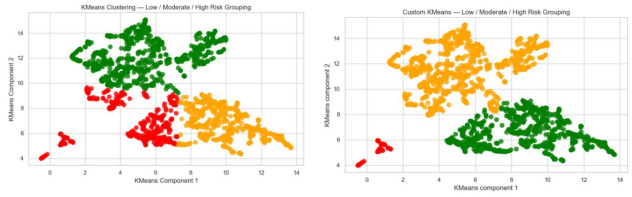

Fig. 11. K Means Cluster

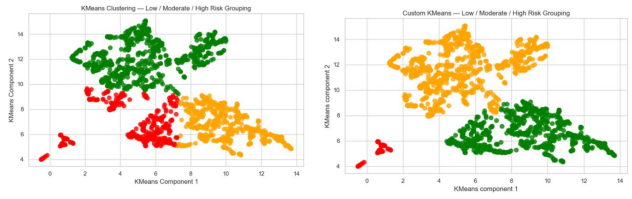

Fig. 12. Hierarchical Clusters

The visualizations in Fig. 10 and Fig. 11 illustrate the clustering results of the predicted fetal risk scores using both KMeans and Hierarchical Clustering. The plots show the separation of data points into three distinct groups—Low, Moderate, and High Risk—mapped using UMAP for dimensionality reduction. While both clustering methods successfully distinguish the groups, hierarchical clustering (Fig. 11) appears to produce more compact and well-separated clusters compared to KMeans (Fig. 10), aligning with the evaluation metrics.

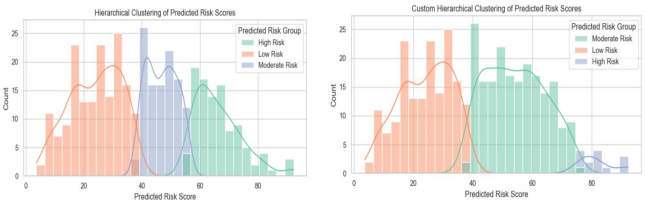

Fig. 13. Analysis of Risk Score Distribution by Clustering Method

The standard hierarchical clustering shows well-separated and evenly distributed risk groups, reflecting clear distinctions in predicted risk scores. The custom model shows some overlap between Moderate and High Risk categories.

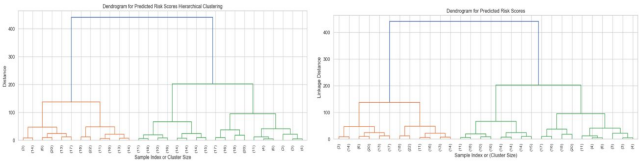

Fig. 14. Dendrograms for Hierarchical Clustering

The dendrogram structure from hierarchical clustering validated the grouping within the input features and the predicted risk scores. It revealed clear separations between clusters at higher linkage distances, indicating strong dissimilarity between patient groups.

C. SHAP-Based Feature Interpretability

SHAP was used to understand how different features contributed to fetal risk predictions across all models. Features such as histogram width, histogram min, and histogram variance were consistently among the most influential in shaping the predictions. Interestingly, while these core features remained important across all models, the influence slightly varied depending on the model's architecture—linear vs. tree-based. For example, XGBoost and Random Forest models was also influenced by features like percentage of time with abnormal long term variability and uterine contractions.

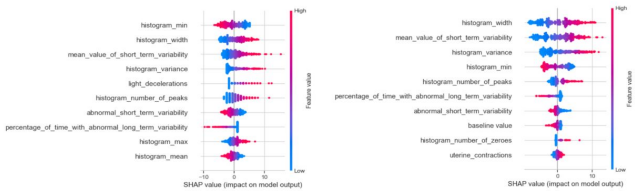

Fig. 15. SHAP For Ridge Regression and XG Boost

VI. Conclusion

In this project, we built a full machine learning framework to predict fetal health risk using cardiotocography (CTG) data. Instead of sticking with rigid class labels, we created a continuous risk score using PCA, which allowed us to approach the problem with regression models. Ridge Regression ended up performing the best, likely because it matched the linear nature of our PCA-based target. Random Forest and XGBoost still delivered strong results, proving their reliability even when the data isn't perfectly suited for them.

To understand how the models were making predictions, we used SHAP values, which gave us clear insights into which features mattered most like histogram width, variability measures, and decelerations. On top of that, we explored clustering techniques like KMeans and Hierarchical Clustering to group patients based on risk scores. These groupings made sense both visually (through UMAP plots) and numerically (with strong clustering metrics).

In the end, the combination of predictive models, feature explainability, and clustering gives us a well-rounded and interpretable system that could support real-world decisions in fetal health care.

VII. Future Work

In the future, we could implement:

- Nonlinear Risk Score Generation: Instead of PCA, future work can explore nonlinear dimensionality reduction methods like autoencoders or t-SNE to generate more realistic risk scores that capture complex interactions among features.
- Temporal Modeling with Raw Signals: The current dataset consists of 21 numerical features that summarize CTG readings rather than raw time-series signals. If full time-series data becomes available, we could explore models like LSTMs or temporal CNNs to allow the model to learn temporal dependencies and patterns directly.

Acknowledgment

Thank you to Professor Ahmad for a great semester! This course has been a valuable learning experience, and we truly appreciate the opportunity to apply what we've learned to a meaningful healthcare problem.

References

[1] J. Li and X. Liu, "Fetal Health Classification Based on Machine Learning," 2021.

[2] J. Piri and P. Mohapatra, "Exploring Fetal Health Status Using an Association Based Classification Approach," 2019.

[3] Z. Cömert, A. Şengür, Ü. Budak, and A. F. Kocamaz, "Prediction of Intrapartum Fetal Hypoxia Considering Feature Selection Algorithms and Machine Learning Models," *Computers in Biology and Medicine*, vol. 113, 2019.

[4] H. J. Chang, Y. J. Kim, J. W. Park, S. J. Kwon, and J. K. Lee, "Machine Learning Model for Classifying the Results of Fetal Cardiotocography," *Yonsei Medical Journal*, vol. 63, no. 7, pp. 672–678, 2022.

[5] O. C. Olayemi and O. O. Olasehinde, "Machine Learning Prediction of Fetal Health Status from Cardiotocography Examination in Developing Healthcare Contexts," *Journal of Computer Science Research*, vol. 6, no. 1, 2024.

[6] J. Feng, J. Liang, Z. Qiang, et al., "A Hybrid Stacked Ensemble and Kernel SHAP-Based Model for Intelligent Cardiotocography Classification," *BMC Medical Informatics and Decision Making*, vol. 23, 2023.

[7] A. M. Vilalta, "Fetal Health Classification Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification

[8] S. Lundberg, "SHAP: A Unified Approach to Explain the Output of Machine Learning Models," GitHub Repository. [Online]. Available: https://github.com/slundberg/shap

[9] L. McInnes and J. Healy, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," [Online]. Available: https://umap-learn.readthedocs.io/en/latest/

[10] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[11] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. Available: https://scikit-learn.org/stable/