Fetal Health Prediction

Soham Ghosh
Roll No. 2022504
B.Tech CSAI 2nd Year

Varin Kala
Roll No. 2022561
B.Tech CSAI 2nd Year

I. Motivation

Child and maternal mortality remain significant global challenges, particularly in low-resource settings. Addressing these issues requires accessible and effective methods to monitor fetal health during pregnancy and intervene promptly to prevent adverse outcomes. By leveraging innovative technologies and data-driven approaches, we aim to save lives by empowering healthcare providers to intervene swiftly and decisively, potentially averting countless tragedies and ensuring that more mothers and babies thrive.

II. Literature Review

As traditional methods for interpreting and classifying Cardiotocography (CTG) data yielded unsatisfactory results, research focus shifted over the last decade toward applying computational intelligent techniques such as Neural Networks for CTG classification. Comert et al. used the CTU-UHB dataset with 552 raw CTG data points to predict the fetal state with 87.9% accuracy. In another work with the same dataset, they reported an accuracy of 91.8% and 93.4%, respectively, with an artificial neural network (ANN) and an extreme learning machine (ELM) Comert (2019). These models have shown high accuracy for normal and pathological states; however, the accuracy dropped to 59% for suspicious CTGs. Sundar et al. [2] generated a model using XGBoost with an accuracy of 96% for the pathological state but only 73% for suspicious state. Santosh et al. [1] released an article on soft-computing based approach on Fetal Health Classification.

III. Dataset

We used the Fetal Health Classification Dataset [3] available on Kaggle. This dataset contains 2126 records of 21 features extracted from Cardiotocogram exams, which have been classified by three expert obstetricians into 3 classes: Normal, Suspect, and Pathological Cases.

After checking that no null values were present for any of the features, we noticed that there were drastically many entries for one class with respect to the others (Figure 2).

Figure 3 shows the correlation matrix of all the features.

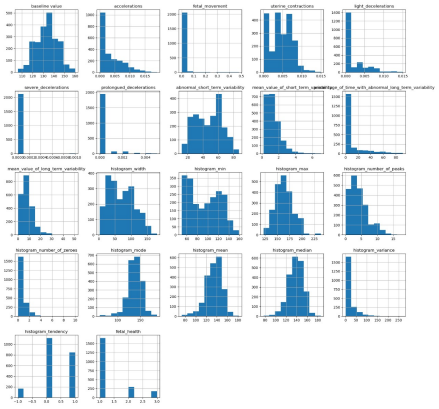

Fig. 1. Feature-wise Distribution

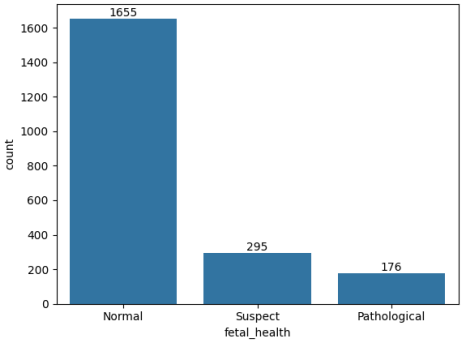

Fig. 2. Count according to Fetal Health

IV. Proposed Architecture

Logistic Regression Models: This is a statistical method used for classification. It models the relationship between the dependent variable and one or more independent variables by estimating probabilities using the logistic function. While it's simple and interpretable, Logistic Regression assumes linear relationships between features and may not capture complex patterns well.

Gradient Boosting Classifiers: Gradient Boosting Classifier (GBC) sequentially builds decision trees, correcting errors iteratively, offering high accuracy in complex datasets. It's versatile, handling various tasks but can be computationally intensive and prone to overfitting. However with cross validation we managed to prevent overfitting.

Random Forest Models: Random Forests (RF) are one of the best choices due to their ability to train each tree independently, utilizing a random subset of the data. This randomness mitigates the risk of overfitting. Our dataset benefited from RF's performance. While RF is relatively faster, increasing the number of estimators can significantly extend the model's training time. Moreover, RF may favor selecting smaller groups when dealing with correlated features

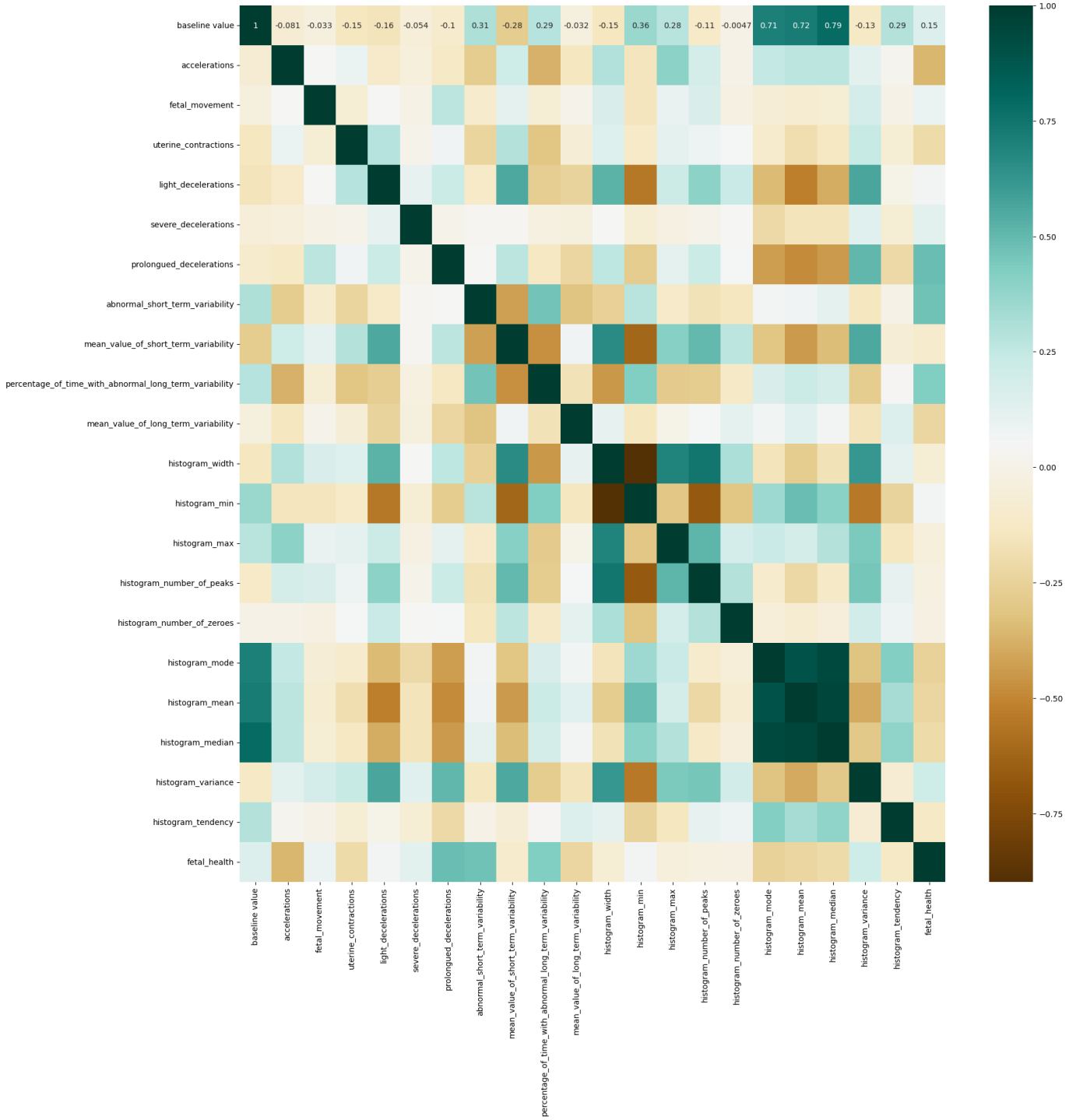

Fig. 3. Correlation Matrix of All Features

of comparable relevance.

K Neighbours Classifier: K-Nearest Neighbors (KNN) is a non-parametric algorithm used for both classification and regression tasks. It classifies data points based on the majority class of their K nearest neighbors in the feature space. After scaling we do not see many outliers in our data so KNN should perform accurately for our dataset.

SVM: Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates classes in the feature space. SVM is effective in high-dimensional spaces, making it suitable for complex data. It can handle non-linear relationships through kernel functions. In our scenario where the separating hyperplane between classes is non linear this model perform robustly.

GridSearch CV Grid Search Cross-Validation (Grid Search CV) is a technique used to tune hyperparameters for machine learning models systematically. It exhaustively searches through a specified grid of hyperparameters, evaluating each combination using cross-validation to find the optimal settings. Thus we modify our best model even more by using the best parameters for our dataset.

V. Methodology

On taking a closer look at the data, we found out that the features are spread out across the 2-D space (Figure 4). So, we scale the data to fit the model better and reduce outliers (Figure 5).

We divided the data into train and test datasets and then applied different classification methods as well as cross validation and found their accuracies using the test data.

Next, we performed Grid Search on various loss functions, different learning rates, and increasing maximum depths of the decision trees in the ensemble to find out the combination which gives us the best accuracy.

We finally trained a Gradient Boosted Decision Tree Model according to the recently found optimal parameters to get the best result.

VI. Results

Initial Methods applied to dataset:

- Logistic Regression: Accuracy = 89.71%
- Gradient Boosting of Decision Trees: Accuracy = 94.53%
- Random Forest: Accuracy = 93.47%
- kNN Classifier: Accuracy = 89.88%
- Support Vector Machine: Accuracy = 91.18%

The Best parameters for the Model turned out as follows:

- Loss Function: Deviance
- Learning Rate = 0.075
- Maximum Depth = 3

After applying these parameters to the final Model, our Accuracy became 95.5%.

VII. Inference

- Scaling the features to fit the model better and reduce outliers is a crucial preprocessing step, especially when dealing with features spread out across the 2-D space. It helps improve the model's performance (kNN Classifier) by ensuring that all features contribute equally to the prediction.
- Trying out different classification methods and comparing their accuracies on the test data helps in selecting the most suitable model for the dataset, considering factors like accuracy, interpretability, and computational complexity.
- Performing Grid Search to find the optimal combination of hyperparameters is essential for improving model performance. It helps fine-tune the model to achieve the best possible accuracy.
- Despite the significantly higher number of entries initially belonging to one class compared to others, the accuracy remained high, indicating that the skewed input distribution did not pose a problem (Figure 6).

VIII. (

Individual Contribution) The work was equally shared by us. Both of us contributed in all aspects of the project, including brainstorming, implementation, debugging, compilation etc.

Soham Ghosh

- Suggesting the topic for the project
- Dataset Research
- Final Methodology
- Implementation
- Compilation of the Project

Varin Kala

- Suggesting the topic for the project
- Literature Review
- Dataset Research
- Implementation
- Compilation of the Project

References

[1] Sahana Das, Himadri Mukherjee, Kaushik Roy and Chanchal Kumar Saha, Fetal Health Classification from Cardiotocograph for Both Stages of Labor—A Soft-Computing-Based Approach, Special Issue **Diagnosis and Management in Prenatal Medicine—2nd Edition**

[2] Sundar, C.; Chitradevi, M.; Geetharamani, G. Classification of cardiotocogram data using neural network based machine learning technique

[3] https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification/data

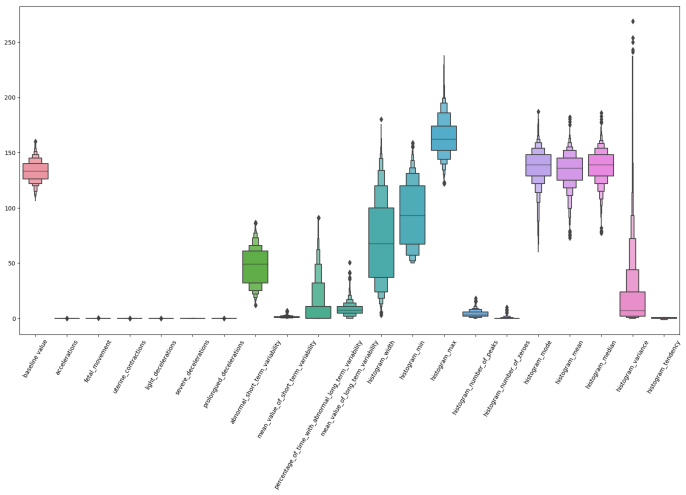

Fig. 4. Distribution before scaling

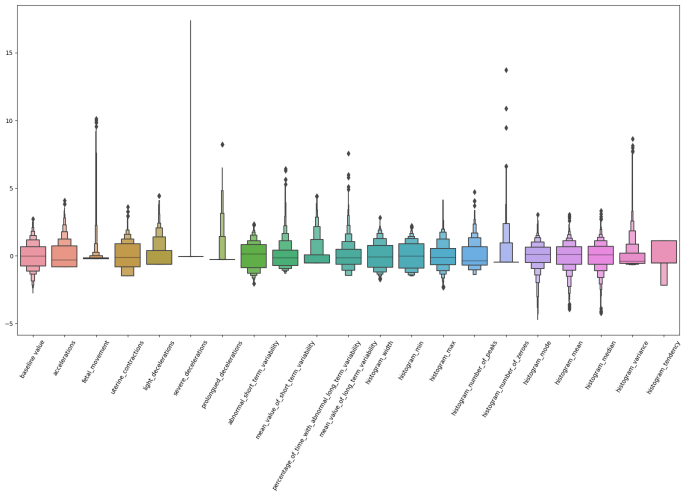

Fig. 5. Distribution after scaling

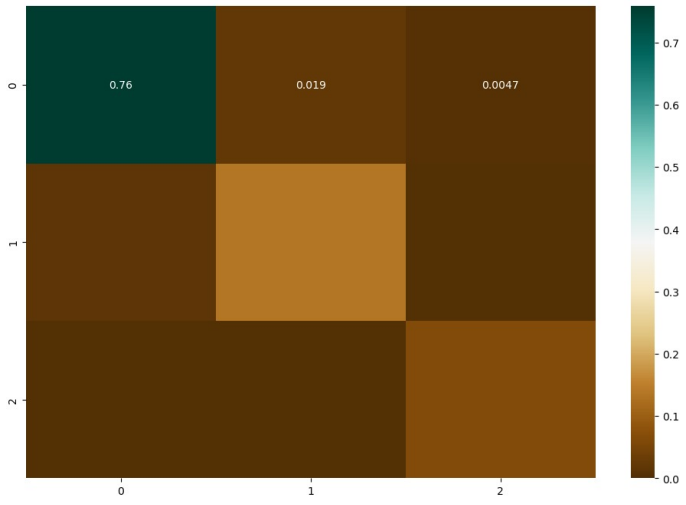

Fig. 6. Confusion Matrix: Distribution of the actual vs the predicted classes. 0 denotes Normal, 1 means suspected, 2 means Pathological. Note that for every class, the highest probability is obtained at the diagonal itself