**Final Report**

**Team Members:**

Anjum Afrooz Shaik, Vijay Susmitha Bommini

**Title of the project:**

Predictive Modeling of Fetal Health Status Using Machine Learning Algorithms

## **Project Aim:**

This project is aimed at developing a classification model to predict the health status of fetuses based on various fetal health parameters. The dataset used for this task contains features extracted from cardiotocogram (CTG) exams, which are commonly used to monitor fetal health during pregnancy. In this report, we present the implementation details and performance evaluation of the Support Vector Machine (SVM), Random Forest (RF), and AdaBoost classifier for fetal health classification.

**GitHub** – <https://github.com/jumjumshaik/Fetal-classification>

<https://github.com/vbommini/fetal-health>

**Dataset: Campos,D. and Bernardes,J.. (2010). Cardiotocography. UCI Machine Learning Repository.**

[**https://doi.org/10.24432/C51S4N**](https://doi.org/10.24432/C51S4N)

**Support Vector Machine:**

* **Data Loading and Inspection:**

Initially, loaded the fetal health dataset and then Inspect the dataset for missing values and data integrity issues.

* **Data Preprocessing:**

In preprocessing, we separated the dataset into features (X) and target (Y) and performed standardization to scale the input features**.**

* **Model Development:**

The dataset is split into training and testing sets and initialized an SVM classifier with balanced class weights to handle class imbalance.

Then grid search with cross-validation is used to tune hyperparameters such as C, Kernel, and Gamma and identify the best-performing model based on cross-validation results.

* **Model Training and Evaluation:**

In the Next step selected SVM classifier is trained on the training set and the trained model is evaluated on the test set.

Then classification metrics such as accuracy, weighted F1 score, sensitivity, and precision are calculated, and a confusion matrix and ROC Curve for multi-class classification are also visualized to assess model performance.

* **Result Interpretation:**

Training Accuracy and Test Accuracy achieved by the model are **94%** and **89%** respectively.

Finally, the performance metrics are analyzed to determine the effectiveness of the SVM classifier in fetal health classification**.**

**Random Forest Classifier**:

* The dataset was loaded from the provided CSV file and inspected for missing values and infinity values. No missing or infinity values were found.
* Then features were separated into input variables (X) and target variable (Y), where X contains the fetal health parameters and Y contains the corresponding health status labels.
* Standardization was applied to scale the input features using ‘StandardScaler’ from sklearn.preprocessing.
* Then the dataset was split into training and testing sets using an 80:20 ratio, and Hyperparameter tuning was performed using grid search with stratified k-fold cross-validation to find the optimal combination of hyperparameters for the model.
* The best-performing model was selected based on the accuracy score obtained from the cross-validation process.
* The trained RF classifier was evaluated using the testing set to assess its performance on unseen data.
* Classification metrics such as accuracy, weighted F1 score, sensitivity (recall), precision, and confusion matrix were computed to measure the model's effectiveness in classifying fetal health.
* The model achieved a training accuracy of **98%** and a testing accuracy of **94%,** indicating its ability to correctly classify fetal health status.
* The confusion matrix and ROC Curve for multi class classification are generated that give insights into the model's predictive performance for each class.

**Adaboost Classifier:**

First necessary libraries are imported. We loaded the fetal health dataset using the `pd.read\_csv` function and split it into features (X) and the target variable (Y).

We divided the data into training and testing sets using the `train\_test\_split` function and standardized the features using the `StandardScaler`.

We then used an AdaBoost classifier with weighted classes due to class imbalance and a decision tree base estimator (`DecisionTreeClassifier`) and defined a grid of hyperparameters (`param\_grid`) as ‘n\_estimators : [50,100,150] and ‘learning\_rate’ : [0.01, 0.1, 1.0].

We then performed a grid search (`GridSearchCV`) to identify the best combination of hyperparameters based on accuracy and printed the best accuracy and parameters.

We fitted the final AdaBoost classifier with the best hyperparameters on the training data and made predictions on the test data and evaluated the model's performance using accuracy and a classification report (`classification\_report`).

Additionally, we also transformed the categorical test labels into a binary format as it is necessary for multi-class classification evaluation. By binarizing the labels, we converted the multi-class problem into binary classification tasks, which are used to plot ROC curves and calculate the Area Under the Curve (AUC) for each class individually.

We also generated a confusion matrix to assess the model's performance visually.

The model obtained an accuracy of **92%** on the training set and **89%** on the test set.

The classification report also includes detailed metrics such as precision, recall, and F1-score along with accuracy for each class, giving us a complete view of the model's performance.

**CONCLUSION:**

Among the three classifiers, Random Forest achieved the highest accuracy of 94% followed by the Adaboost Classifier (89%) and SVM Classifier (89%).