**ACADEMIC REPORT TM 93**

**Introduction:**

The solution applies machine learning to the cardiotocography (CTG) dataset and aims to classify fetal health into Normal, Suspect, and Pathologic categories. Timely detection of oxygen deprivation in the fetus is vital for avoiding adverse outcomes but because CTG traces are complex, clinicians under pressure may overlook subtle indicators. Accurate and interpretable models are hence vital in clinical decision support, as misclassification may lead to delayed care.

The report summarizes the preprocessing pipeline, exploratory data analysis, and comparative evaluation of multiple algorithms, with random forest (with feature engineering) selected as the final model.

**Preprocessing and EDA:**

The dataset was checked for missing values and a preprocessing function was defined. Features were standardized using StandardScaler, and labels were encoded into numerical categories. A stratified 80/20 train-test split preserved class balance, while SMOTE was used to address imbalance between Normal and minority classes - Suspect and Pathological.  
We used EDA to plot graphs and visualize the data and identify key patterns and trends.  
Patterns - The first target distribution chart shows the class counts for NSP (1=Normal, 2=Suspect, 3=Pathologic). The class imbalance motivated us to use balanced metrics (Balanced Accuracy, Macro-F1) and class-imbalance strategies (SMOTE) in subsequent modeling.  
The Global histograms summarize each numeric feature’s marginal distribution. Several variables (AC, MSTV, ALTV, Nmax and so on) are skewed and/or heavy-tailed, suggesting light monotonic transforms for non-negative features (e.g., `log1p`, `sqrt`) to stabilize variance and make downstream splits more robust.

The heatmap highlights strong feature collinearity (e.g., mode,median and mean), but since Random Forests are robust to redundancy, removing correlated features didn’t improve performance. We therefore prioritize model-based evaluation (OOB/test metrics, permutation importance) over correlation-only pruning.

Boxplots show features like ASTV, ATLV, and AC with clear median shifts and compact IQRs across NSP classes, indicating strong discriminative power; this aligns with later model based feature importance results.

PCA scatter plots show partial class separation but substantial overlap, confirming that no obvious clustering exists and justifying the importance of using classification models that consider multiple features.

Pairplots reveal class separation patterns (e.g., ASTV–ALTV, ALTV–MLTV), motivating feature engineering of ratios, differences. This expanded the feature set from 21 to 38 and improved RF performance (Macro-F1: 0.932 to 0.935).

**Model Design:**  
First, we tested baseline models - linear regression, decision trees, random forests, extra trees, histogram-based gradient boosting, support vector classifier (SVC), and XGBoost. After getting a classification report for all these models, we decided to evaluate the different models on the basis of Balanced Accuracy and Macro-F1 by plotting a bar graph. We chose to go ahead with random forest as it had the highest balanced accuracy and macro-f1 score.  
To further improve predictive performance and interpretability, we applied a feature-engineering pipeline before training the Random Forest model.  
Feature engineering added clinically meaningful pairwise ratios, differences, products, log/sqrt transforms, quantile bins, and LOF/Isolation Forest anomaly scores. A large ensemble Random Forest (n\_estimators=1200, max\_features="sqrt", OOB enabled) was trained and evaluated.  
After feature engineering, we trained a Random Forest as the main model and evaluated its performance. We tested a soft-voting ensemble with XGBoost and trained RF/XGB individually also. We further explored hyperparameter tuning of the Random Forest using randomized search with 5-fold cross-validation, optimizing for macro-F1 and applied per-class threshold calibration to adjust decision boundaries.  
The cross-validated hyperparameter tuning and per-class threshold adjustments caused a slight decline in performance, suggesting overfitting. Therefore, we finalize the baseline Random Forest with feature engineering as our best model. While XGBoost performed relatively well in the plain baseline (without feature engineering), its performance declined after feature engineering. Consequently, both XGBoost alone and the soft-voting ensemble (RF + XGBoost) underperformed compared to the feature-engineered Random Forest, particularly due to low recall on the suspect class. We further applied SHAP and LIME to the finalized Random Forest to interpret feature contributions and understand the model’s decision-making.

In parallel, we systematically experimented with a range of neural network approaches to enhance classification performance on the CTG dataset. These included architectural changes like deeper MLPs and residual connections, training strategies such as focal loss, label smoothing, mixup augmentation, and data preprocessing techniques like power transformation and feature engineering. While some methods offered small gains, many degraded performance compared to the baseline. The most consistent improvements came from neural network ensembles, where averaging predictions from up to 20 independently trained models modestly improved macro-F1 and balanced accuracy. Feature engineering further boosted ensemble performance, showing that richer representations helped NN models generalize better.  
However, despite these efforts, the best-performing neural network ensemble still fell short of the feature-engineered Random Forest in terms of balanced accuracy, macro-F1, and interpretability. Thus, even after exploring both advanced neural architectures and ensembles, we conclude that the Random Forest with feature engineering remains the most effective and reliable model for this dataset.