

LIFELINE: ACADEMIC REPORT

1. Background & Problem

Cardiotocography (CTG) is used to monitor fetal well-being during labour, and plays an important role in reducing pregnancy complications. However, CTG readings can be difficult to interpret, and manual analysis is often time-consuming. This project aims to build a solution that supports clinicians in spotting fetal distress by using real-world data and machine learning to reliably classify cases as Normal, Suspect, or Pathologic.

2. Dataset

Our team used the UCI CTG dataset (<https://archive.ics.uci.edu/dataset/193/cardiotocography>), which has 2126 samples and 21 features. These features are heart rate metrics (LB, AC, FM, UC, DL, DP), variability (ASTV, MSTV, ALTV, MLTV), and statistical features (Mode, Mean, Median, Variance, Tendency). The variables are integer or continuous, and there are no missing values in the dataset. However, the dataset is imbalanced across the three classes Normal (1655), Suspect (295), and Pathological (176). Thus, feature standardisation and a stratified train-test split were performed to address the class imbalance.

The data was also relatively clean, with no missing values, hence little data cleaning was required. Furthermore, since the features are specifically used by doctors for predictions, little feature engineering had to be done to achieve high performing models. In fact, attempts at feature engineering actually hurt the models performance.

3. Models

Different models were tested, including linear models (Logistic Regression), tree models (Decision Tree, Random Forest, LightGBM, XGBoost), clustering (k-Nearest Neighbours), and neural approaches (TabNet, PyTorch, MLP). After, the performance of the models were compared using classification reports of precision, recall, and F1-score.

Tree based ensembles performed the best, likely because the dataset is tabular (with neural networks being too prone to overfitting). 5-fold cross validation was then performed on the models to determine which model performed the best.

4. Results

We did quick experiments where we trained and tested the models on one train test split, and changed the random state a few times to get an idea of the models performance on the task. This way we could try out multiple models. After doing this, we realised that tree based ensembles tended to perform the best, and decided to do 5-fold cross validation on selected ensembles, with some hyper-parameter tuning, to determine the best performing model for submission. Our classification reports, with the average train-test time, are below:


- **Random Forests:**

 Average train-test time: 2.32 seconds.

Classification Report (5-fold CV on training set):

	precision	recall	f1-score	support
1	0.95	0.98	0.97	1655
2	0.87	0.73	0.79	295
3	0.95	0.89	0.92	176
accuracy			0.94	2126
macro avg	0.92	0.87	0.89	2126
weighted avg	0.94	0.94	0.94	2126

- **LightGBM (balanced weights):**

 Classification Report (5-fold CV on training set):

	precision	recall	f1-score	support
1	0.97	0.98	0.98	1655
2	0.87	0.86	0.87	295
3	0.94	0.92	0.93	176
accuracy			0.96	2126
macro avg	0.93	0.92	0.92	2126
weighted avg	0.96	0.96	0.96	2126

- **XGBoost (balanced weights):**

 Average train-test time: 0.51 seconds.

Classification Report (5-fold CV on training set):

	precision	recall	f1-score	support
0	0.96	0.99	0.97	1655
1	0.90	0.80	0.85	295
2	0.96	0.91	0.94	176
accuracy			0.95	2126
macro avg	0.94	0.90	0.92	2126
weighted avg	0.95	0.95	0.95	2126

- **Imbalance-focused Ensembles (SPE):**

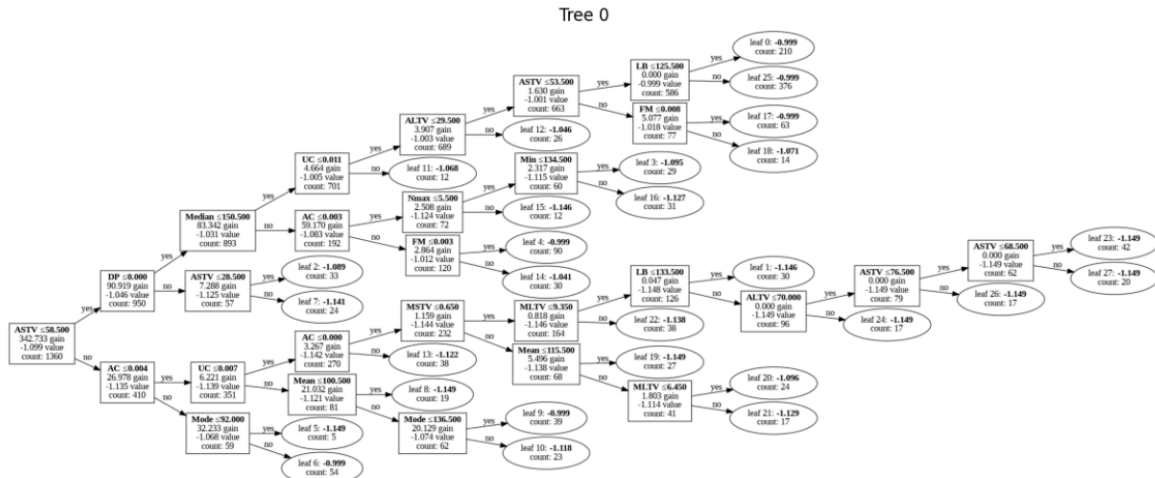
➡ Average train-test time: 0.52 seconds.

Classification Report (5-fold CV on training set):

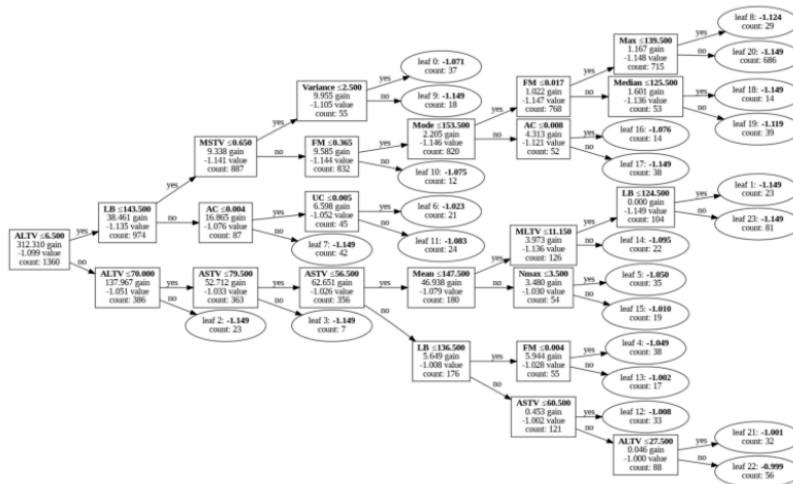
	precision	recall	f1-score	support
1	0.97	0.97	0.97	1655
2	0.82	0.84	0.83	295
3	0.91	0.92	0.92	176
accuracy			0.95	2126
macro avg	0.90	0.91	0.91	2126
weighted avg	0.95	0.95	0.95	2126

5. Interpretability

After comparing the 5-fold cross validation, and also taking into account the average train-test duration, we decided that LightGBM was the best performing model overall. With this in mind, we wanted to visualize the actual decisions being made within the decision tree. However, there are quite a few weak learners, so we decided to just display the first 2.



Tree 1



Conclusion:

After comparing the 5-fold cross validation results , and also taking into account the average train-test duration, we decided that LightGBM was the best performing and most efficient model overall, and thus would be our model for submission.