1. Explore and experiment with various algorithms and hyper-parameters to arrive at and select your best performing model (in Python). You must justify your target performance matrix selection to compare various algorithms and models.

A screenshot of a graph

Description automatically generated

The Random Forest model exhibits strong performance across all metrics when compared to the other two models. It boasts a high recall, along with the highest accuracy, F1 score, and precision. Moreover, it shares the top position for the ROC AUC score. Due to its ensemble nature—averaging multiple decision trees—Random Forest is less prone to overfitting and demonstrates robustness. In our context, a false positive, which would incorrectly predict that a patient will show up, carries significant repercussions. Given its superior precision, the Random Forest should be the model of choice.

1. Select with justification TOP predictive features out of all included features. How many and which features you should include in your model? What is your methodology to select those features?

We selected six features with the highest feature weights for our model. These were chosen based on their predictive power related to our model's objectives. Initially, all features were used in the first phase of model building to determine the most effective model.

1. Test your deployment thoroughly (both modes). How would you ensure your testing is adequate? Would you test the boundary conditions? Provide evidence.

Locally, the path can be adjusted according to the folder location. However, in a web service or cloud deployment, we need to ensure that the paths are absolute, or that all files are in a single folder. We also add a logging process (please see streamlit\_app.py) to the deployment Python file so that we can access information on the performance of the deployment processes.

1. What is your PUBLIC URL for the Streamlit Server deployment?

<https://chla-prediction-bsan-mlops.streamlit.app/>