Tabular Classification on TabMini

Objective:

This assignment aims to provide hands-on experience with various machine learning models for tabular classification tasks, specifically focusing on the TabMini dataset. You will be evaluating the performance of different models, analyzing their strengths and weaknesses, and gaining insights into the challenges of working with limited data.

Dataset:

The TabMini dataset (available at github RicardoKnauer/TabMini) is a collection of 44 binary classification datasets designed for evaluating machine learning models in low-data regimes.

Models:

You will be working with the following models:

1. **XGBoost:** A gradient boosting algorithm that is widely used for its high accuracy and efficiency.
2. **LightGBM:** Another gradient boosting algorithm known for its speed and scalability.
3. **CatBoost:** A gradient boosting algorithm that handles categorical features effectively.
4. **Random Forest:** An ensemble learning method that combines multiple decision trees.
5. **ResNet:** A deep learning model originally designed for image classification but can be adapted for tabular data.
6. **FT-Transformer:** A transformer-based model that leverages feature tokens for tabular data.
7. **MLP-PLR:** A multilayer perceptron with piecewise linear regression. On Embeddings for Numerical Features in Tabular Deep Learning
8. **TabR:** A model that combines recurrent neural networks with attention mechanisms for tabular data. (link: https://openreview.net/forum?id=rhgIgTSSxW)
9. **TabTransformer:** A transformer-based model specifically designed for tabular data.
10. **TabNet:** A deep learning model that uses sequential attention to select features.
11. **SAINT:** A self-attention based model for tabular data imputation and classification.

Tasks:

1. **Implementation:** Implement the specified models using a machine learning library of your choice (e.g., scikit-learn, PyTorch, TensorFlow). You can leverage existing implementations or build your own from scratch.
2. **Evaluation:** Train and evaluate each model on the TabMini dataset. Use appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and AUC.
3. **Comparison:** Compare the performance of different models across various datasets in TabMini. Analyze their strengths and weaknesses in the context of low-data regimes.
4. **Analysis:** Investigate the impact of hyperparameter tuning on model performance. Experiment with different hyperparameter settings and analyze their effects.
5. **Report:** Write a comprehensive report summarizing your findings. Include the following sections:
   - Introduction: Briefly describe the task and the dataset.
   - Methods: Explain the implemented models and evaluation metrics.

    - Results: Present the performance of each model and compare them.
    - Discussion: Analyze the results and discuss the strengths and weaknesses of different models.
    - Conclusion: Summarize your findings and draw conclusions.

Grading Rubric:

- Implementation (40%): Correct implementation of the specified models.
- Evaluation (30%): Proper evaluation of models using appropriate metrics.
- Comparison (20%): Thorough comparison of model performance and analysis of their strengths and weaknesses.
- Report (10%): Clear and concise report summarizing the findings.

Submission:

Submit your code and report electronically via the course website.

Good luck!