Week 2 Overview Report

Focus: Implementation of the TabNet Model for Structured Data

---

Tasks Completed This Week

- Implemented and trained a TabNet model using the pytorch-tabnet library
- Performed data normalization and preprocessing tailored to TabNet's requirements
- Conducted model evaluation using accuracy, precision, recall, and ROC-AUC metrics
- Compared TabNet performance against baseline deep learning models
- Generated feature importance plots to interpret TabNet's decision-making process

---

Progress Towards Final Goal

- Project is approximately **35-40%** complete
- TabNet provided better interpretability and handled tabular structure efficiently
- Prepared ground for upcoming ensemble model benchmarking and tuning

---

Challenges Faced

- TabNet model training was sensitive to learning rate and batch size
- Longer training times compared to simpler models
- Feature scaling impacted performance unexpectedly
- Managing GPU resources for larger data folds

Support Required

- Best strategies for tuning TabNet hyperparameters
- Advice on balancing TabNet training speed vs. accuracy
- Benchmarks or use cases comparing TabNet with ensemble models like XGBoost or CatBoost

---

Time Spent This Week

Approx. 13 hours

- Data preparation: 2h
- TabNet implementation: 4h
- Evaluation and comparison: 4h

- Feature analysis and reporting: 3h

Plan for the Coming Week

- Benchmark TabNet against Random Forest and XGBoost
- Start ensemble modeling and stacking experiments
- Improve interpretability using SHAP values
- Begin planning dashboard layout for model performance visualization