**Assignment Reflection and Experience**

**Title: XGBoost Model Performance Evaluation - Simple Reflection**

**Introduction**

The purpose of this assignment involved evaluating the performance of XGBoost models operating on varying dataset volumes when implemented in both Python and R. The training process for XGBoost models was evaluated through two approaches: direct XGBoost implementation and XGBoost implementation through the caret package. The project required expanding small data into larger datasets while training models and saving results and final report generation.

**What I Did**

I expanded the PimaIndiansDiabetes2 dataset through multiple sample copies.

The datasets contained 100, 1,000, 10,000, 100,000 and 1,000,000 records.

The XGBoost model training using Python scikit-learn took place with 5-fold cross-validation methodology.

The training of models occurred in R through direct xgboost() implementation and caret::xgbTree execution.

The process included recording both accuracy results and execution times for each dataset size.

The final table incorporated Python and R results together.

A Word document containing all results and explanations served as the final product.

**Problems I Faced**

My laptop experienced overheating and crashing issues when I ran caret model training on large datasets particularly 1 million records.

Training caret models on very large datasets caused errors related to memory allocation.

Training on large datasets would cause Google Colab to disconnect from the system.

The merged CSV files contained corrupt or incomplete data which produced NaN values.

The accuracy value of -inf appeared when some caret models experienced failure during execution.

**How I Solved the Problems**

The training of caret models occurred on small datasets with maximum records reaching 100,000 to prevent laptop crashes.

I limited the boosting rounds to 2 to speed up the training process and decrease its weight.

The process included proper matching of column names and re-uploading clean CSV files.

The rows containing NaN values were removed while -inf values received "N/A" substitution.

Only direct XGBoost handled large datasets while caret worked exclusively on smaller datasets.

I constructed the complete Word report manually from the beginning.

**Lessons Learned**

Big Data requires planning since unprepared execution of large models might result in system failure.

Simple models reduce both memory problems and overheating issues.

Users should check all files after merging data results.

The practice of respecting hardware limits demands that users work with smaller datasets instead of larger ones.

Regular saving of work is essential to prevent work loss.

**Final Thoughts**

Through this assignment I learned substantial knowledge about machine learning and big data management while resolving practical system crashes and memory-related issues. My knowledge about work adjustment according to my laptop's performance limitations enables me to better organize tasks for large projects.