XGBoost Model Performance Summary

# Objective

Testing-set predictive results and model execution durations for XGBoost models receive evaluation based on different dataset scales through three different implementation approaches.- Python XGBoost (5-fold cross-validation)

- R xgboost() direct method

- R caret::xgbTree method (for small datasets only)

# Methods

# A bootstrapping method produced synthetic datasets which stemmed from PimaIndiansDiabetes2 database.Dataset sizes tested: 100, 1,000, 10,000, 100,000, and 1,000,000 records.The analysis of large datasets (>=100,000) involved using a limited boosting round number (nrounds=2) because of time and processing limitations.A record of model fitting duration in seconds and testing-set predictive accuracy percentage was obtained for each tested method.

# Results

|  |  |  |  |
| --- | --- | --- | --- |
| Method Used | Dataset Size | Testing-set Predictive Performance (%) | Time Taken to Fit (seconds) |
| Python XGBoost 5-fold CV | 100 | 80.0 | 2.73 |
| R caret::xgbTree | 100 | N/A | 2.12 |
| R xgboost() direct | 100 | 84.28 | 0.01 |
| R xgboost() direct | 1000 | 87.8 | 0.01 |
| R caret::xgbTree | 1000 | N/A | 4.71 |
| Python XGBoost 5-fold CV | 1000 | 90.6 | 0.37 |
| Python XGBoost 5-fold CV | 10000 | 100.0 | 0.78 |
| R xgboost() direct | 10000 | 89.33 | 0.06 |
| R caret::xgbTree | 10000 | N/A | 24.99 |
| Python XGBoost 5-fold CV | 100000 | 86.69 | 1.6 |
| R xgboost() direct | 100000 | 81.53 | 0.26 |
| R xgboost() direct | 100000 | 88.59 | 0.62 |
| R caret::xgbTree | 100000 | N/A | 234.12 |
| R xgboost() direct | 1000000 | 88.06 | 6.33 |
| R caret::xgbTree | 1000000 | N/A | 2355.55 |
| Python XGBoost 5-fold CV | 1000000 | 86.59 | 6.63 |
| R xgboost() direct | 1000000 | 81.66 | 2.32 |

# Recommendation

Based on the experimental results:  
  
For small datasets (100, 1,000, 10,000):

The predictive results from all tested methods remained comparable.

The xgbTree function in caret provides adaptable model tuning options yet performs slightly slower than native xgboost execution.

- For large datasets (100,000, 1,000,000):

The fitting process completed significantly faster when practitioners used xgboost() directly in R and Python.

Training with caret proved impractical for big datasets because of limited available time.

The model team decided against conducting training for 10,000,000 records because of hardware restrictions.

The observed trends show that additional accuracy improvements would take an enormous increase in training duration.

**Final Recommendation:**

Xgboost() directly applied to large datasets should be used because it provides the fastest fitting times and highest efficiency. When model tuning matters for datasets of medium and small sizes then caret::xgbTree provides an appropriate solution.