

Method	Size	Accuracy	CV Time (s)
XGBoost (Python 5-fold CV)	100	0.890000	3.647653
XGBoost (Python 5-fold CV)	1000	0.944000	0.430796
XGBoost (Python 5-fold CV)	10000	0.975400	1.418593
XGBoost (Python 5-fold CV)	100000	0.986940	6.675767
XGBoost (Python 5-fold CV)	1000000	0.991862	26.669050
XGBoost (Python 5-fold CV)	10000000	0.993118	288.356472

XGBoost in Python via scikit-learn and 5-fold Cross-Validation

The XGBoost model implemented in Python using scikit-learn and evaluated with 5-fold cross-validation consistently demonstrated strong performance across all dataset sizes. As the dataset size increased, testing-set predictive performance improved from 0.89 accuracy at 100 records to 0.993 at 10 million records. The computational time required for model fitting scaled appropriately with dataset size, ranging from approximately 3.6 seconds for 100 rows to about 288 seconds for 10 million rows.

This method offers an excellent balance between speed and predictive accuracy. Even at very large scales, the model fitting remained efficient, showing that XGBoost in Python is highly scalable and well-suited for production-grade machine learning workflows. Compared to the R implementations, the Python approach achieved comparable or slightly better predictive performance at larger dataset sizes, with faster execution times than caret and on par with direct `xgboost()` usage in R.

Thus, for many real-world applications where both rapid model development and high accuracy are critical, XGBoost via scikit-learn in Python emerges as a highly practical and efficient choice.