Method Used	Dataset Size	Testing-set predictive performance	Time taken for the model to be fit
XGBoost in Python via scikit-learn and 5-fold CV	100		
	1000		
	10000		
	100000		
	1000000		
	10000000		
XGBoost in R – direct use of xgboost() with simple cross-validation	100	0.90	0.2
	1000	0.9450	0.6
	10000	0.9735	0.9
	100000	0.9774	1.9
	1000000	0.9786	6.09
	10000000	0.9862	63.29
XGBoost in R – via caret, with 5-fold CV simple cross-validation	100	0.95	1.96
	1000	0.98	3.87
	10000	0.9835	18.77
	100000	0.9863	102.67
	1000000	0.9906	423.67
	10000000	0.9926	1024.3

The XGBoost implementation through caret with 5-fold cross-validation proves superior to direct xgboost() usage with simple cross-validation. The caret implementation delivers superior predictive performance across all dataset sizes by achieving testing-set performance increases from 0.05 to 0.0156. The performance enhancement provides exceptional value for processing large datasets because small accuracy gains lead to substantial practical advantages.

The better performance outcome from this approach demands significant computational resources. The caret implementation needs substantially more time to train models than the direct xgboost() method because it requires 5-17 times longer processing time based on dataset

dimensions. The choice between caret and direct xgboost() implementation depends on whether accuracy or fast development and real-time predictions take priority given available computing power.