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	0.8500	0.02s
	1000	0.9400	0.05s
	10000	0.9710	0.26s
	100000	0.9838	1.46s
	1000000	0.9884	13.17s
	10000000	0.9906	113.26s
XGBoost in R – via caret, with 5-fold CV simple cross-validation	100	0.9194	2.20s
	1000	0.9640	4.83s
	10000	0.9812	22.63s
	100000	0.9901	210.90s
	1000000	0.9923	400.90s
	10000000	0.9949	1020.60s

XGBoost via caret with 5-fold CV should be the recommended model for most production applications based on performance results. The caret implementation achieves superior predictive performance throughout all dataset sizes ranging from 0.9194 to 0.9949 while the alternative method produces results from 0.8500 to 0.9906. This indicates significant accuracy improvements. The longer computational duration (up to 9x for the biggest dataset) becomes acceptable because XGBoost via caret with 5-fold cross-validation delivers superior prediction results that matter most in critical applications.

The direct XGBoost implementation serves as an acceptable solution for time-sensitive applications that handle big datasets or scenarios with restricted computational resources. The direct XGBoost implementation gives high performance baseline while executing tasks at dramatically accelerated speeds which makes it ideal for rapid development situations. Our standard machine learning projects will benefit significantly from caret implementation because its thorough validation and advanced hyperparameter optimization produce superior results even though the processing time is longer.