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	0.95	3.04
	1000	0.9650	2.38
	10000	0.9555	4.77
	100000	0.9547	16.01
	1000000	0.9654	29.36
	10000000	0.9785	180.36
XGBoost in R – via caret, with 5-fold CV simple cross-validation	100	0.95	3.04
	1000	0.9650	2.38
	10000	0.9555	4.77
	100000	0.9547	16.01
	1000000	0.9652	196.5
	10000000	0.9526	369.25

Most scenarios would benefit from implementing XGBoost directly instead of using the caret-based approach according to the provided data. The direct XGBoost approach demonstrated consistent or superior predictive performance especially with 10M observations where it reached an accuracy of 0.9785 while needing substantially less computation time. The direct implementation demonstrated enhanced speed with increased dataset scale since it performed the process twice faster than the caret-based approach when analyzing 10M observations (180.36 seconds against 369.25 seconds). Moreover it delivered better predictive success rates (0.9785 against 0.9526). The efficiency advantage of direct XGBoost implementation enables rapid model development along with better suitability for production settings that face resource constraints.