Method Used	**Dataset Size**	**Testing-set pred performance**	**Time taken for fitting the model**
XGBoost in Python via scikit-learn and 5-fold CV	100	92.0%	1.407 secs
	1000	95.1%	0.766 secs
	10000	97.4%	0.169 secs
	100000	98.8%	0.947 secs
	1000000	99.4%	7.958 secs
	10000000	99.5%	83.37 secs
XGBoost in R – direct use of xgboost() with simple cross-validation	100	84.8%	0.036 secs
	1000	88.2%	0.272 secs
	10000	94.4%	2.542 secs
	100000	96.6%	22.93 secs
	1000000	97.1%	220.6 secs
	10000000	97.1%	2382.0 secs
XGBoost in R – via caret, with 5-fold CV simple cross-validation	100	96.1%	3.197 secs
	1000	96.3%	7.433 secs
	10000	98.3%	38.33 secs
	100000	99.1%	357.7 secs
	1000000	99.2%	3619.3 secs
	10000000		

Although all the methods show great performances especially when using a greater number of samples, I would definitely recommend Python scikit-learn as it clearly outperforms other R methods in time. It can even be improved by simply shifting to GPU. It can also be integrated with variety of high-end technology tools as well.

The last sample is not included because it takes too much time to run, and it will be over the deadline. I did not expect this to take so long.