|  |  |  |  |
| --- | --- | --- | --- |
| Method Used | Dataset Size | Testing-set predictive performance (Test Accuracy) | Time taken for the model to be fit |
| XGBoost in Python via scikit-learn and 5-fold CV | 100 | 1.000 | 0.180s |
|  | 1000 | 0.975 | 0.898s |
|  | 10000 | 0.9815 | 0.539s |
|  | 100000 | 0.9881 | 2.6393s |
|  | 1000000 | 0.9916 | 23.3994s |
| XGBoost in R – direct use of xgboost() with simple cross-validation | 100 | 0.80 | 0.06 |
|  | 1000 | 0.945 | 0.10 |
|  | 10000 | 0.950 | 0.12 |
|  | 100000 | 0.9506 | 0.87 |
|  | 1000000 | 0.9504 | 3.72 |
| XGBoost in R – via caret, with 5-fold CV simple cross-validation | 100 | 0.90 | 17.959 |
|  | 1000 | 0.98 | 25.17782 |
|  | 10000 | 0.984 | 41.71721 |
|  | 100000 | 0.99085 | 194.25172 |
|  | 1000000 | 0.99257 | 2009.44074 |

2.

The XGBoost in Python via scikit-learn implementation with 5-fold cross-validation should be adopted according to performance metrics. The approach strikes the best prediction precision and computational performance rate across full data ranges. The XGBoost in Python via scikit-learn implementation reaches 99.16% accuracy when processing 1,000,000 observations within 23.4 seconds which represents 86 times faster performance than R caret implementation with similar accuracy levels.

The R implementation of XGBoost delivers the fastest processing times by finishing the largest dataset in 3.72 seconds although this speed comes with reduced prediction accuracy. The lower prediction accuracy of 4-5 percentage points compared to other methods on large datasets makes this approach unfeasible since it offers minimal time benefits compared to the Python implementation.

The R caret implementation with 5-fold CV demonstrates slightly higher accuracy on big datasets (99.26% vs 99.16%) yet takes more than 33 minutes to analyze 1 million observations. Most practical applications would find this accuracy-to-time ratio unacceptably poor. For most practical business applications the Python implementation stands out as the better option because it delivers performance that is both equivalent to the R caret version and multiple times faster when applied to large datasets or needs repeated model testing.