1.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 0.8667 | 0.05s |
|  | 1000 | 0.9467 | 0.19s |
|  | 10000 | 0.969 | 1.47s |
|  | 100000 | 0.9818 | 12.14s |
|  | 1000000 | 0.9873 | 118.23s |
|  | 10000000 | 0.9913 | 239.79s |
| XGBoost in R – via caret, with 5-fold CV simple cross-validation | 100 | 0.8621 | 0.23s |
|  | 1000 | 0.9532 | 0.35s |
|  | 10000 | 0.967 | 1.31s |
|  | 100000 | 0.9833 | 10.78s |
|  | 1000000 | 0.9861 | 103.52s |
|  | 10000000 | 0.9895 | 300.67s |

2.

XGBoost in R should be used through direct application of xgboost() while performing simple cross-validation according to the results. The direct approach generates superior predictive results than caret implementation at every dataset size point from 0.8667 to 0.9913 while maintaining better performance than caret implementation at 0.8621 to 0.9895. When working with large datasets containing 10 million observations XGBoost through direct implementation takes 239.79 seconds while caret requires 300.67 seconds to finish.

The performance benefits of the direct approach surpass the convenience advantage of using caret implementation for cross-validation. The direct method achieves enhanced prediction accuracy at the same time it decreases computation time by about 20% when applied to large datasets which would be beneficial for production systems where model training occurs frequently on extensive datasets.