|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 0.9500 | 0.01008391s |
|  | 1000 | 0.935000 | 0.01435709s |
|  | 10000 | 0.949000 | 0.06828213s |
|  | 100000 | 0.949500 | 0.63202405s |
|  | 1000000 | 0.950865 | 6.03221297s |
|  | 10000000 | 0.960156 | 40.03255266s |
| XGBoost in R – via caret, with 5-fold CV simple cross-validation | 100 | 0.8792256 | 2.288597s |
|  | 1000 | 0.9639696 | 4.361512s |
|  | 10000 | 0.9840007 | 23.639893s |
|  | 100000 | 0.9882155 | 270.635956s |
|  | 1000000 | 0.9892152 | 2400.659126s |
|  | 10000000 | 0.995353 | 7900.362254s |

Most practical applications should adopt direct XGBoost over the caret implementation based on performance metrics. The testing accuracy of caret reaches 0.995 but requires 200 times longer training duration of 7900 seconds compared to 40 seconds for the XGBoost implementation. The direct implementation of XGBoost enables both fast execution times along with accurate prediction results at 0.96+ accuracy making it an optimal choice for production applications and iterative model development since it offers better computational capacity. Larger datasets reveal an improved efficiency advantage for direct XGBoost implementation because it demonstrates better scalability for real-world applications that rarely find the increased accuracy from caret justified when considering the extra processing time.