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 | 0.85 | 0.03s |
|  | 1000 | 0.945 | 0.19s |
|  | 10000 | 0.9775 | 1.4s |
|  | 100000 | 0.9833 | 12.49s |
|  | 1000000 | 0.9867 | 118.44s |
|  | 10000000 | 0.9913 | 240.59s |
| XGBoost in R – via caret, with 5-fold CV simple cross-validation | 100 | 0.95 | 0.23s |
|  | 1000 | 0.95 | 0.26s |
|  | 10000 | 0.973 | 1.68s |
|  | 100000 | 0.9835 | 14.03s |
|  | 1000000 | 0.9853 | 133.47s |
|  | 10000000 | 0.9943 | 300.69s |

2.

XGBoost implemented through the caret package with 5-fold cross-validation should be used according to the provided data. The chosen approach produces better predictive results on all dataset sizes especially when analyzing the 10M dataset where it reaches 0.9943 compared to direct XGBoost at 0.9913. The extended computational time of caret implementation at 25% longer duration is offset by superior model performance. Production environments will benefit from the caret package by using its standardized interface for hyperparameter tuning and model comparison and feature selection capabilities even though its processing time ends up being slightly longer.