XGBoost Model Comparison Using Python and R

Introduction

This report presents a comparative analysis of XGBoost implementations across Python (via scikit-learn) and R (using both direct xgboost() and caret packages) using datasets of varying sizes. The goal is to assess and compare predictive performance and computational efficiency of each approach on synthetic data generated through sampling with replacement. Predictions are made using logistic regression models.

Methodology

Tools Used: Python (scikit-learn), R (base xgboost, caret), 5-fold cross-validation.

Dataset: Synthetic data sampled from original dataset; predictions made using logistic regression.

Evaluation Metrics:

- Accuracy on the testing set.

- Time taken for model fitting in seconds.

Results Table

Method	**Dataset Size**	**Accuracy**	**Time (s)**
Python XGBoost	100	0.8500	0.88
Python XGBoost	1,000	0.9380	1.88
Python XGBoost	10,000	0.9737	7.99
Python XGBoost	100,000	0.9885	4.02
Python XGBoost	1,000,000	0.9918	10.46
Python XGBoost	10,000,000	0.9932	94.28
R xgboost() direct CV	100	0.9009	0.34
R xgboost() direct CV	1,000	0.9340	0.65
R xgboost() direct CV	10,000	0.9592	1.44

R xgboost() direct CV	100,000	0.9720	7.96
R xgboost() direct CV	1,000,000	0.9749	72.60
R xgboost() direct CV	10,000,000	0.9756	238.84
R caret xgboost	100	0.9100	1.88
R caret xgboost	1,000	0.9349	0.70
R caret xgboost	10,000	0.9724	1.64
R caret xgboost	100,000	0.9847	12.97
R caret xgboost	1,000,000	0.9887	109.28
R caret xgboost	10,000,000	0.9896	1007.01

Analysis & Recommendation

While all methods show improvement in accuracy with larger datasets, Python's implementation of XGBoost provides consistently high accuracy with better computational efficiency at scale. For instance, Python XGBoost reaches 0.9932 accuracy at 10 million rows in 94 seconds, whereas R caret takes over 1000 seconds for slightly lower accuracy. Therefore, Python XGBoost is recommended for large-scale applications due to its performance and speed.

Conclusion

XGBoost remains a powerful model across platforms. However, Python's implementation is better optimized for speed and large data. R caret provides more interpretability and structured CV, which is ideal for smaller projects. Selection depends on use case and resource constraints.