s

Week 08 – GLM Investigation Assignment

Asritha Suraparaju

Course Code: 5230-07: High-Performance Computing

March 30, 2025

Module/Framework/Package	**Name and Brief Description of the Algorithm**	**Example Where It Outperforms Base R or Equivalent in Python**
Base R (stats package)	**Iteratively Reweighted Least Squares (IRLS)**: A traditional optimization method for GLMs that iteratively updates weights to improve model estimates. It works well for small to medium dataset	Works fine for small datasets but struggles with large datasets due to memory limitations. If the dataset is too big, the computation becomes slow or may fail due to memory overflow. **Python Equivalent**: statsmodels.api.GLM.
Big Data R (High-Performance Computing in R)	**Optimized memory-efficient methods (biglm, speedglm)**: These methods process data in chunks rather than loading it all into memory, preventing performance issues.	When working with datasets with millions of rows, these methods allow computation to happen without requiring all data to fit in RAM, making it much faster than Base R. **Python Equivalent**: sklearn.linear_model.SGDClassifier, which also processes data in batches.
Dask-ML (dask_ml.glm)	**Stochastic Gradient Descent (SGD) and Newton's Method**: Dask-ML distributes computations across multiple CPU cores or even multiple machines, making it efficient for big data.	If you are working with a dataset that exceeds the available RAM, Dask-ML allows parallel processing across distributed systems, making it much faster than Base R. **Python Equivalent**: sklearn.linear_model.SGDClassifier.
Spark R (spark.glm)	**IRLS and Distributed Optimization**: Similar to Base R's IRLS but optimized for Spark's	When handling datasets too large for a single machine (e.g., billions of rows), Spark R splits computations across a cluster, greatly improving speed and efficiency. **Python Equivalent**: pyspark.ml.classification.LogisticRegression.

	distributed computing framework, allowing for large-scale computations across multiple nodes.	
Spark MLlib Optimization	**L-BFGS and Stochastic Gradient Descent (SGD)**: Uses advanced optimization techniques for large-scale distributed computing. L-BFGS (Limited-memory BFGS) is particularly effective for handling large datasets with complex models.	Best suited for massive datasets stored in a distributed system (e.g., cloud computing environments). It significantly outperforms Base R when working with data that spans multiple machines. **Python Equivalent**: pyspark.mllib.optimization.
Scikit-learn (sklearn.linear_model)	**L-B** Provides multiple solvers for optimization, allowing flexibility depending on dataset size and complexity. L-BFGS is great for large datasets, while liblinear is efficient for high-dimensional sparse data.	Works better than Base R for moderately large datasets by offering better solver selection, faster convergence, and built-in regularization. **R Equivalent**: glmnet, which also provides efficient optimization methods.