Module/framework/package	**Name and brief description of algorithm**	**An example of a situation where using the provided GLM implementation provides superior performance compared to that of base R or its equivalent in Python (identify the equivalent in Python)**
Base R	The optimization process of Iteratively Reweighted Least Squares (IWLS) consists of multiple weighted least squares regressions which optimize the log-likelihood function. Supports multiple families (e.g., Gaussian, Binomial, Poisson). The optimization process uses successive updates which lead to maximum likelihood estimates. Single-threaded operation of this method makes it inefficient when working with large-dimensional or vast datasets.	The IWLS method works efficiently with datasets containing a small number of items or datasets with simple statistical models that have a medium sample size. Large datasets and high-dimensional data cause a substantial rise in computational costs for this method. The Python equivalents for IWLS implementation exist in statsmodels and scikit-learn through Logistic Regression with L-BFGS.
Big Data version of R	Parallel Processing Tools (biglm, ff, bigmemory, RcppParallel) analyze data sizes larger than memory by distributing them across smaller parts and by storing information on external disks. The packages foreach, parallel, snow and RcppParallel enable parallelization of processes. Large datasets benefit from this framework yet it does not possess the reliability of Spark framework.	Superior handles data that exceeds memory capacity which makes it more scalable than Base R. Dask and PySpark serve as equivalent solutions for Python applications. The biglm package enables users to implement GLM fitting as packages that use Dask-ML offer distributed fitting capabilities across big data arrays.
Dask ML	ADMM with Gradient Descent and L-BFGS and Newton’s Method provide distributed data array scalability through their	The combination of out-of-core computation with parallelization leads to enhanced performance when dealing with extensive

	implementation framework. The package enables users to select from L1, L2 and ElasticNet regularizers. The system uses dask.distributed to execute efficient parallel algorithms that operate across distributed clusters. The system executes out-of-core operations efficiently which makes it highly effective for processing very large datasets.	distributed datasets. Scikit-learn does not handle data beyond memory capacity but Dask-ML extends its capabilities to distribute computations across several workers. Such functionality serves High-dimensional and sparse datasets specifically well. Scikit-learn lacks built-in capabilities to distribute computations across multiple computational systems.
Spark R	IRLS within Spark MLlib functions as a distributed computation framework based on Apache Spark. The algorithm supports fitting distributions from five different families which include Gaussian, Binomial, Poisson, Gamma, and Tweedie. The system delivers strong data processing capabilities through cluster-based distributed computation.	The system delivers exceptional performance when operating on distributed data across Spark clusters. SparkR demonstrates improved scalability against Base R and scikit-learn for processing terabytes of distributed data through its distributed framework. The fitting of complex GLMs proves most beneficial when running across large clusters. The system provides fault-tolerance functionality that Base R does not support.
Spark optimization	The three optimization primitives designed for large-scale machine learning include SGD, L-BFGS and Distributed SGD. The online learning capabilities belong to SGD but L-BFGS demonstrates better performance for batch processing because of its fast convergence property. The framework implements mini-batch gradient descent operations which enable quick training of huge dataset quantities.	This framework works perfectly for executing distributed model training on massive systems. L-BFGS achieves faster convergence speeds than SGD when dealing with problems that have good condition numbers. Mini-batch processing enables efficient optimization of big data collections and delivers better results than standard Base R implementations and offers built-in features for gradient aggregation and regularization.

Scikit-Learn	The package provides Gradient Descent, L-BFGS, Newton-CG, SAG, SAGA solvers that handle both convex and non-convex problems in GLMs. The framework enables users to apply L1, L2, ElasticNet regularization methods and perform GridSearchCV and RandomizedSearchCV hyperparameter tuning. Nonlinear data processing via joblib helps speed up the system.	The system works best for datasets of medium size while offering solid implementation capabilities alongside hyperparameter adjustment features. The system falls short of Dask and Spark when dealing with extremely large datasets. The LogisticRegression algorithm from Scikit-learn uses L-BFGS or SAGA for efficient convergence yet it does not support out-of-core computation which makes it inefficient for processing very large datasets.