

Category	Module/framework/package	Name and a brief description of the 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 (in the stats library)	Glm() function from base R.	The function is called glm(). The default method " glm.fit " uses iteratively reweighted least squares (IRLS), which is an iterative algorithm based on the Newton-Raphson optimization.	N/A
Big data version of R	Bigglm() function from biglm package.	The function is called bigglm(). It handles large data by processing in chunks and uses an iterative updating algorithm.	Working with a large clinical dataset with millions of rows. Using base R's glm() might result in memory overload.
Dask ML	LinearRegression(), LogisticRegression(), and PoissonRegression() from dask_ml.linear_model	Dask_ml has a variety of algorithms: These include admm (Alternating Direction Method of Multipliers), gradient_descent (Michael Grant's implementation of Gradient Descent), lbfgs (L-BFGS solver using scipy.optimize implementation), newton (Newton's Method for Logistic Regression), and proximal_grad (Proximal Gradient Method).	For example, if you are working with large data in Python, and your data is already present in a distributed dask dataframe, it makes sense to use the estimators in dask_ml package. Python provides Generalized Linear Modeling (GLMs) mainly through statsmodels . Compared to the base Python approach using statsmodels, if we are working with a Dask Dataframe rather than a NumPy array, it makes sense to use dask_ml package.
Spark R	Spark.glm() function	It uses the IRLS, which is an iterative algorithm based on the Newton-Raphson optimization.	For example, if you are working on a Spark dataframe or using Spark as your analytics engine to process large insurance data,

			spark.glm() is useful to model the loss ratio. Compare this to base R's glm(), we can still use this but not for very large data.
Spark optimization	MLlib in Spark	Spark MLlib supports several optimization techniques based on the type of model and solver: IRLS, L-BFGS, SGD, and ADMM.	For example, if we have a large dataset that's growing in size or may vary in size, and we need scalable regression, spark MLlib is better than the base Python statsmodel approach. Python provides Generalized Linear Modeling (GLMs) mainly through statsmodels .
Scikit-learn	Linear_model from sklearn package	Scikit-learn classes use Ordinary Least Squares (OLS), LBFGS, or stochastic gradient descent (SGD), depending on which family (Linear, Logistic..)	Sci-kit learn is preferred to statsmodels when it comes to machine learning use-cases. For example, in a machine learning project with a large dataset, where I can set the number of cores (n_cores), sci-kit learn will outperform statsmodels (base Python).