	Module/framework /package	**Name and a brief description of the algorithm**	**An example of a situation**
a.	Base R (stats::glm)	IRLS represents the default method for maximum likelihood estimation through iterative solutions of weighted least square problems. The IRLS method solves weighted least squares consecutively to estimate maximum likelihood parameters through adjustments of response and weights based on link and variance functions at every iteration	This tool or lib performs detailed statistical modeling and hypothesis testing operations on datasets able to fit within random access memory capacity. The output objects from XGBoost offer deep diagnostic capabilities although they deliver slower performance in terms of speed on large datasets compared to Scikit-learn or distributed frameworks.
b.	Big data version of R (biglm)	biglm updates parameters through iterative processing of chunks for dealing with datasets that exceed memory capacity. The algorithm makes parameter modifications based on successive chunks to conduct parameter approximations without having to load all data.	This solution performs well with data that cannot fit into the memory of a single machine so users can process large datasets without cluster infrastructure. The performance exceeds base R when data swapping takes place but becomes slower than the in-memory implementation of Scikit-learn if the data fits within RAM. Python equivalent for out-of-core: Dask.
c.	Dask ML	SGD performs	Superior Works best

	(dask_ml.glm)	out-of-core/parallel optimization by processing data through small portions.	for streaming data as well as datasets that exceed memory capacity (such as online advertising click-through prediction).
d.	Spark R (SparkR::spark.glm)	The implementation of Distributed IRLS primarily utilizes distributed optimizations of the IRLS method. Spark implements data distribution together with computation processing during weighted least squares operations.	The massive data processing system (potentially handling terabytes of information) operates on Spark clouds (HDFS, S3) effectively. The program utilizes Spark's distributed execution system to perform fault-resistant large-scale parallel processing. Outperforms single-node (R/Scikit-learn) and potentially Dask for very large scale. Python equivalent: PySpark MLlib (pyspark.ml.regression.GeneralizedLinearRegression).
e.	Spark optimization	Stochastic Average Gradient (SAG) - A fast gradient method that averages past gradients, efficient for big data.	A 500GB dataset with sparse features yields optimal GLM results when evaluated through IRLS from base R (too slow) and the non-distributed Scikit-learn.
f.	Scikit-learn (linear_model)	Stochastic Average Gradient for fast convergence on medium datasets.	Due to optimized solvers the dataset containing 50,000 rows could be fitted with logistic regression

			efficiently compared to base R's IRLS routine