|  |  |  |
| --- | --- | --- |
| Module/framework/package | Algorithm used | When it performs better than Base R or Python equivalent |
| Base R (stats library) | IRLS functions as the typical procedure for GLMs to determine the best weight values by running the model multiple times. | Small and medium-sized datasets can benefit from this method yet it fails to handle extensive data effectively. |
| Big Data R (HighPerformanceComputing) | The package employs parallel computing and chunk processing through bigglm and speedglm functions. | This method functions for datasets bigger than memory limits thus surpassing Base R's glm(). |
| Dask ML | The system implements gradient-based optimization together with parallel computing. | Scikit-learn operates at a slower pace than parallel processing of large data by multiple computers. |
| Spark R (spark.glm) | IRLS optimization runs as a distributed system designed especially for Spark cluster environments. | Large datasets exceeding Base R memory capacity can be managed effectively by using this software. Faster than Python's statsmodels or sklearn in big data scenarios. |
| Spark MLlib Optimization | The system executes training operations for large-scale GLM through SGD and L-BFGS methodologies. | The platform suits large machine learning applications which require parallel processing although it differs from standard R or Python models. |
| Scikit-learn | The software utilizes liblinear, lbfgs and saga solvers because these tools optimize GLM efficiency. | The system performs better than Base R for datasets of medium to large sizes particularly in sparse data operations and multi-core CPU applications. |