Shreya Samir Kurani
HDS_5230-07
Assignment Week_08

GLM INVESTIGATION

Number	Module/framework/package	Name & Description of Optimization algorithm	An example of a situation where it provides superior performance
a.	Base R (stats package)	It is a form of Fishers Scoring known as Iteratively Reweighted Least Squares (IRLS). It provides an iterative solution to weighted least squares problems to estimate parameters.	It is best suited for small to medium sized memory data sets. The equivalent in Python is: `statsmodels.api.GLM`.
b.	Big Data R (glmnet package)	In case of regularized models, `glmnet` relies on cyclical coordinate descent; `bigstatsr` employs memory mapping as well as parallel computing for its support of very large datasets.	For high dimensional or large dataset, it outperforms base R. It uses Python equivalent: `scikit-learn` with `LogisticRegressionCV` or `lightgbm` for the regularized models.
c.	Dask ML (dask-glm)	They support ADMM, L-BFGS, Newton's Method, and Gradient Descent. It is designed to run distributed computation on large datasets.	It is helpful when the data cannot fit into a memory. Better than base R. Python non-distributed equivalent: `scikit-learn`.
d.	SparkR (spark.glm)	IRLS uses Spark's distributed DataFrames and is optimized.	It is ideal for large-scale Spark environments. Scales far better than base R. Python equivalent: PySpark's `glm()`.
e.	Spark MLlib (spark.mllib)	It uses scalable optimization techniques for regularized GLMs are rotational as L-BFGS, OWL-QN, and SGD frame employed.	It can perform quite well on very large datasets within a distributed cluster. Python equivalent: PySpark MLlib.
f.	Scikit-learn (sklearn.linear_model)	It offers liblinear, L-BFGS, SAG, SAGA solvers. Supports various GLMs and regularization types.	It is efficient for medium to large datasets requiring regularization needs. More flexible than base R for solver selection