

Punto #3.

Consulte en qué consiste y el funcionamiento principal de la librería RAPIDS. Posteriormente, elabore una tabla que compare los métodos e hiperparámetros más relevantes de los regresores mencionados en el punto 2, indicando su implementación o el algoritmo equivalente disponible en RAPIDS.

¿En qué consiste RAPIDS?

NVIDIA RAPIDS es un conjunto de bibliotecas de software de código abierto y API diseñadas para acelerar los procesos de ciencia de datos y análisis dependiendo únicamente de una GPU. Proporciona una interfaz similar a librerías de Python, como pandas, scikit-learn y NetworkX, lo que permite a los científicos de datos aprovechar la inmensa potencia de procesamiento paralelo de las GPU sin tener que aprender a utilizar herramientas completamente nuevas.

¿Cómo es su funcionamiento?

Su objetivo principal es llevar la aceleración por GPU no solo al Deep Learning (lo que ya hacen marcos como TensorFlow o PyTorch), sino también a todo el flujo de trabajo de la ciencia de datos, la limpieza, la ingeniería de características y el aprendizaje automático clásico.

¿Cómo se compara con los regresores ya vistos?

REGRESOR	HIPERPARÁMETROS CLAVE	EQUIVALENTE RAPIDS	NOTAS DE IMPLEMENTACIÓN
Linear Regression	fit_intercept, normalize, copy_X	CumLinear Regression	Totalmente compatible; solucionador DLS generado por GPU.
Lasso	alpha, max_iter, tol, selection	CumLasso	Utiliza descenso coordinado en GPU; equivalente a sklearn.
ElasticNet	alpha, l1_ratio, max_iter, tol	CumElasticNet	Implementación de GPU; combina penalizaciones L1 y L2.
Kernel Ridge	alpha, kernel, gamma, degree, coef0	X No disponible	Aún no existe una versión directa de cuML; se puede aproximar mediante cuML SVR con kernel polinomial / RBF
SGBRegressor	loss, penalty, alpha, learning_rate, max_iter	X No implementado	Actualmente, cuML carece de modelos lineales basados en SGB; se utiliza LinearRegression o Ridge.
Bayesian Ridge	alpha_1, alpha_2, lambda_1, lambda_2	X No disponible	No hay versión cuML; el cálculo requiere muchas matrices, pero se puede acelerar manualmente con GPU mediante Cupy.

REGRESOR	HIPERPARAMETRO CLAVE	EQUIVALENTE RAPIDS	NOTAS DE IMPLEMENTACIÓN
Gaussian Process Regressor	kernel, alpha, optimizer, n-restarts-optimizer	X No disponible	No hay versión GPU en cuML, se utiliza pyTorch o GPyTorch para procesos basados en GPU.
Support Vector Regressor (SVR)	C, kernel, gamma, epsilon, degree	cuML SVR	Implementación completa de GPU; admite kernels lineales, polinómicos y RBF.
Random Forest Regressor	n-estimators, max-depth, max-features, min-samples-split	cuML Random Forest Regressor	Totalmente acelerado por GPU; muy rápido para grandes conjuntos de datos.
Gradient Boosting Regressor / XGBoost	n-estimators, learning-rate, max-depth, subsample, colsample-bytree	cuML Gradient Boosting Regressor	RAPIDS ofrece el GBM nativo de cuML y también es compatible con el backend XGBoost GPU.