Course Outline

Effective Machine Learning with Dask

Course Summary

Description

This class focuses on leveraging Dask for Machine Learning in several different ways: Dask implements a number of distributed algorithms; interoperates with popular Python libraries, and integrates with several external projects (e.g., PyTorch). This module looks at each of the options, as well as the full ML lifecycle, from ingesting data to performing inference.

Objectives

At the end of this course, students will understand:

- What does Dask offer – and not offer – for machine learning workflows
- Leveraging Dask for proper out-of-core and/or parallel training
- Implementing an end-to-end workflow with Dask and other tools

Topics

- Introduction
- Dask and scikit-learn
- Data Preparation and Dask's Algorithms and Integration with XGBoost
- Performing Inference at Scale
- Custom Algorithms
- Review Q & A

Audience

This course is intended for those who manage machine learning workflows.

Prerequisites

The following prerequisites are required for this course:

- Python, basic level
- Understanding of ML concepts and workflow, basic level
- Dask programming, basic level

Duration

One day

Course Outline

Effective Machine Learning with Dask

Course Outline

I. Introduction
- A. What makes scale-out machine learning different and challenging
- B. How Dask flexibility approaches distributed ML challenges
- C. Using Dask with – not instead of – other tools
- D. Dask's model for enabling custom algorithms

II. Dask and scikit-learn
- A. Hyperparameter Search
- B. Out-of-core non-parallel training (incremental)
- C. In-memory parallel training
- D. Combining incremental (out-of-core and parallel training
- E. Review of scikit-learn + Dask helper APIs
- F. How to match scikit-learn algorithms to Dask options

III. Data Preparation and Dask's Algorithms and Integration with XGBoost
- A. Ingesting data
- B. Feature engineering transformations
- C. Model training (GLM, clustering)
- D. Pipelines
- E. Dask + xgboost
- F. Dask + GPU (overview)

IV. Performing Inference at Scale
- A. General patterns for inference
- B. Predicting with Dask Futures
- C. About ParallelPostFit
- D. Low-latency vs. batch vs. resource-intensive inference patterns

V. Custom Algorithms
- A. Dask's task scheduler
- B. Mechanisms for distributing, sharing, aggregating, and collecting data
- C. Example: implementing a simulation model

VI. Review and Q & A
- A. Gotchas and best practices
- B. Architecture options for integrating Dask