Big Data and (Py)Spark - Introduction

(https://github.com/learn-co-curriculum/dsc-big-data-pyspark-intro) (https://github.com/learn-co-curriculum/dsc-big-data-pyspark-intro/issues/new)

Introduction

In this section, you will be introduced to the idea of big data and the tools data scientists use to manage it.

Big Data

Big data is undoubtedly one of the most hyped terms in data science these days. Big data analytics involves dealing with data that is large in volume and high in variety and velocity, making it challenging for data scientists to run their routine analysis activities. In this section, you'll learn the basics of dealing with big data through parallel and distributed computing.

Parallel and Distributed Computing with MapReduce

We start this section by providing more context on the ideas of parallel and distributed computing and **MapReduce**. When talking about distributed and parallel computing, we refer to the fact that complex (and big) data science tasks can be executed over a cluster of interconnected computers instead of on just one machine. You'll learn that MapReduce allows us to convert these big datasets into sets of tuples as key:value pairs, as we'll cover in more detail in this section.

Apache Spark and PySpark

Apache Spark is an open-source distributed cluster-computing framework that makes it easier (and feasible) to use huge amounts of data! It was developed in response to limitations of MapReduce and written using the Scala programming language. Fortunately for Python developers, there is also a Python interface for Spark called **PySpark**. Throughout these lessons we will use the terms "Spark" and "PySpark" fairly interchangeably, though technically "Spark" is the underlying framework and "PySpark" is the Python library we'll be using.

Installing and Configuring PySpark with Docker

PySpark was not part of the original environment setup you completed. While the interface is in Python, Spark relies on an underlying Java virtual machine (JVM) that can be challenging to install.

Therefore we will provide instructions for installing PySpark with and without **Docker**, a container system that uses an "image" to handle a lot of the configuration for you.

Spark Unstructured API

First we'll look at the fundamental low-level data structures used by Spark: SparkContext and RDDs.

RDDs (Resilient Distributed Datasets)

Resilient Distributed Datasets (RDDs) are the core concept in PySpark. RDDs are immutable distributed collections of data objects. Each dataset in RDD is divided into logical partitions, which may be computed on different computers (so-called "nodes") in the Spark cluster. In this section, you'll learn how RDDs in Spark work. Additionally, you'll learn that RDD operations can be split into actions and transformations.

Word Count with MapReduce

You'll use MapReduce with Spark RDDs to solve a basic NLP task where you compare the attributes of different authors of various texts.

Spark Structured API

Then we'll look at the modern use of Spark: SparkSession, Spark DataFrames, and MLlib.

Spark DataFrames

Spark DataFrames are built on top of RDDs, but have a more intuitive and performant data structure. They are also what we'll use for machine learning with Spark.

Machine Learning with Spark

After you've solved a basic MapReduce problem, you will learn about employing the machine learning modules of PySpark. You will perform both a regression and classification problem and get the chance to build a full parallelizable data science pipeline that can scale to work with big data.

Summary

In this section, you'll learn the foundations of Big Data and how to manage it with Apache Spark!