Boot Camp - Data Science using Cloudera

Duration

10 days

Description

Data scientists build information platforms to ask and answer previously unimaginable questions. Learn how data science helps organizations reduce costs, increase efficiency, improve product delivery, improve customers and users experience, and identify new opportunities. Our bootcamp helps participants understand what data scientists do and the problems they solve, and to become a data scientist. Through in-class simulations, participants apply data science methods to real-world challenges in different scenarios and, ultimately, prepare for data scientist roles in the field.

This bootcamp is oriented to the different roles on the data science landscape, Administrators, Developers and Data Analysts.

This bootcamp delivers the key concepts and expertise participants need to ingest and process data on a Hadoop cluster using the most up-to-date tools and techniques. Employing Hadoop ecosystem projects such as Spark, Hive, Flume, Sqoop, and Impala, this training course is the best preparation for the real-world challenges faced by Hadoop developers. Participants learn to identify which tool is the right one to use in a given situation, and will gain hands-on experience in developing using those tools.

Participants will also learn Apache Pig and Hive and Cloudera Impala will teach you to apply traditional data analytics and business intelligence skills to big data. Cloudera presents the tools data professionals need to access, manipulate, transform, and analyze complex data sets using SQL and familiar scripting languages.

Data visualisation is vital in bridging the gap between data and decisions. Discover the methods, tools and processes involved. Data visualisation is an important visual method for effective communication and analysing large datasets. Through data visualisations we are able to draw conclusions from data that are sometimes not immediately obvious and interact with the data in an entirely different way.

This course will provide you with an informative introduction to the methods, tools and processes involved in visualising big data.

Audience

- This course is suitable for system administrators, developers, data analysts, and statisticians;
- In general to all interested *big data* and *data science*;

Prerequisites

- Knowledge on operating systems like Unix/Linux are preferable but non essential;
- Knowledge in a programming language is preferable but non essential.

Objectives

After conclusions participants will learn:

- How to identify potential business use cases where data science can provide impactful results;
- How to obtain, clean and combine disparate data sources to create a coherent picture for analysis;
- What statistical methods to leverage for data exploration that will provide critical insight into your data;
- Where and when to leverage Hadoop streaming and Apache Spark for data science pipelines;
- What machine learning technique to use for a particular data science project;
- How to implement and manage recommenders using Spark's MLlib, and how to set up and evaluate data experiments;
- What are the pitfalls of deploying new analytics projects to production, at scale;
- How data is distributed, stored, and processed in a Hadoop cluster;
- How to use Sqoop and Flume to ingest data;
- How to process distributed data with Apache Spark;
- How to model structured data as tables in Impala and Hive;
- How to choose the best data storage format for different data usage patterns;
- Best practices for data storage;

- The features that Pig, Hive, and Impala offer for data acquisition, storage, and analysis;
- The fundamentals of Apache Hadoop and data ETL (extract, transform, load), ingestion, and processing with Hadoop tools
- How Pig, Hive, and Impala improve productivity for typical analysis tasks
- Joining diverse datasets to gain valuable business insight
- Performing real-time, complex queries on datasets
- Use big data and data science visualization tools

Boot Camp - Data Science using Cloudera

Course Outline:

Introduction

- About This Course
- About Cloudera
- Course Logistics
- Introductions

Data Science Overview

- What Is Data Science?
- The Growing Need for Data Science
- The Role of a Data Scientist

Introduction to Hadoop and the Hadoop Ecosystem

- Problems with Traditional Large-scale Systems
- Hadoop!
- The Hadoop EcoSystem

Hadoop Architecture and HDFS

- Distributed Processing on a Cluster
- Storage: HDFS Architecture
- Storage: Using HDFS
- Resource Management: YARN Architecture
- Resource Management: Working with YARN

Importing Relational Data with Apache Sqoop

- Sqoop Overview
- Basic Imports and Exports
- Limiting Results
- Improving Sqoop's Performance
- Sqoop 2

Introduction to Impala and Hive

- Introduction to Impala and Hive
- Why Use Impala and Hive?
- Comparing Hive to Traditional Databases
- Hive Use Cases

Modeling and Managing Data with Impala and Hive

- Data Storage Overview
- Creating Databases and Tables
- Loading Data into Tables
- HCatalog
- Impala Metadata Caching

Data Formats

- Selecting a File Format
- Hadoop Tool Support for File Formats
- Avro Schemas
- Using Avro with Hive and Sqoop
- Avro Schema Evolution
- Compression

Data Partitioning

- Partitioning Overview

- Partitioning in Impala and Hive

Capturing Data with Apache Flume

- What is Apache Flume?
- Basic Flume Architecture
- Flume Sources
- Flume Sinks
- Flume Channels
- Flume Configuration

Spark Basics

- What is Apache Spark?
- Using the Spark Shell
- RDDs (Resilient Distributed Datasets)
- Functional Programming in Spark

Working with RDDs in Spark

- A Closer Look at RDDs
- Key-Value Pair RDDs
- MapReduce
- Other Pair RDD Operations

Writing and Deploying Spark Applications

- Spark Applications vs. Spark Shell
- Creating the SparkContext
- Building a Spark Application (Scala and Java)
- Running a Spark Application
- The Spark Application Web UI
- Configuring Spark Properties
- Logging

Parallel Programming with Spark

- Review: Spark on a Cluster
- RDD Partitions
- Partitioning of File-based RDDs
- HDFS and Data Locality
- Executing Parallel Operations
- Stages and Tasks

Spark Caching and Persistence

- RDD Lineage
- Caching Overview
- Distributed Persistence

Common Patterns in Spark Data Processing

- Common Spark Use Cases
- Iterative Algorithms in Spark
- Graph Processing and Analysis
- Machine Learning
- Example: k-means

Preview: Spark SQL

- Spark SQL and the SQL Context
- Creating DataFrames
- Transforming and Querying DataFrames
- Saving DataFrames
- Comparing Spark SQL with Impala

Introduction to Pig

- What Is Pig?
- Pig's Features
- Pig Use Cases
- Interacting with Pig

Basic Data Analysis with Pig

- Pig Latin Syntax
- Loading Data
- Simple Data Types
- Field Definitions
- Data Output
- Viewing the Schema
- Filtering and Sorting Data
- Commonly-Used Functions

Processing Complex Data with Pig

- Storage Formats
- Complex/Nested Data Types
- Grouping
- Built-In Functions for Complex Data
- Iterating Grouped Data

Multi-Dataset Operations with Pig

- Techniques for Combining Data Sets
- Joining Data Sets in Pig
- Set Operations
- Splitting Data Sets

Pig Troubleshooting and Optimization

- Troubleshooting Pig
- Logging
- Using Hadoop's Web UI
- Data Sampling and Debugging
- Performance Overview
- Understanding the Execution Plan
- Tips for Improving the Performance of Your Pig Jobs

Introduction to Hive and Impala

- What Is Hive?
- What Is Impala?
- Schema and Data Storage
- Comparing Hive to Traditional Databases
- Hive Use Cases

Querying with Hive and Impala

- Databases and Tables
- Basic Hive and Impala Query Language Syntax
- Data Types
- Differences Between Hive and Impala Query Syntax
- Using Hue to Execute Queries
- Using the Impala Shell

Data Management

- Data Storage
- Creating Databases and Tables
- Loading Data

- Altering Databases and Tables
- Simplifying Queries with Views
- Storing Query Results

Data Storage and Performance

- Partitioning Tables
- Choosing a File Format
- Managing Metadata
- Controlling Access to Data

Relational Data Analysis with Hive and Impala

- Joining Datasets
- Common Built-In Functions
- Aggregation and Windowing

Working with Impala

- How Impala Executes Queries
- Extending Impala with User-Defined Functions
- Improving Impala Performance

Analyzing Text and Complex Data with Hive

- Complex Values in Hive
- Using Regular Expressions in Hive
- Sentiment Analysis and N-Grams
- Conclusion

Hive Optimization

- Understanding Query Performance
- Controlling Job Execution Plan
- Bucketing
- Indexing Data

Extending Hive

- SerDes
- Data Transformation with Custom Scripts
- User-Defined Functions
- Parameterized Queries

Choosing the Best Tool for the Job

- Comparing MapReduce, Pig, Hive, Impala, and Relational Databases
- Which to Choose?

Visualizations Tools

- Try different data visualization tools
- Discover the methods, tools and processes involved.
- Choosing the Best Visualization Tool for the Job

Conclusion