Data Engineering Bootcamp using Python and PySpark

Course Number: WA3020

Duration: 5 days

Course Description

This hands-on Data Engineering Bootcamp teaches attendees the foundations of data engineering using Python and Spark SQL. Students learn how to build production-ready data-driven solutions and gain a comprehensive understanding of data engineering.

- Big Data Concepts and Systems Overview for Data Engineers
- Defining Data Engineering
- Data Processing Phases
- Python 3 Introduction
- Python Variables and Types
- Control Statements and Data Collections
- Functions and Modules
- Working with File I/O and Useful Modules
- Practical Introduction to NumPy
- Practical Introduction to pandas
- Data Grouping and Aggregation with pandas
- Repairing and Normalizing Data
- Data Visualization in Python
- Python as a Cloud Scripting Language

- Introduction to Apache Spark
- The Spark Shell
- Spark RDDs
- Parallel Data Processing with Spark
- Introduction to Spark SQL

Skills Gained

- Data Availability and Consistency
- A/B Testing Data Engineering Tasks Project
- Learning the Databricks Community Cloud Lab Environment
- Python Variables
- Dates and Times
- The if, for, and try Constructs
- Dictionaries
- Sets, Tuples
- Functions, Functional Programming
- Understanding NumPy and pandas
- PySpark

Who Can Benefit

This Data Engineer Bootcamp training is targeted to Data Engineers

Prerequisites

Some working experience in any programming language; the students will be introduced to programming in Python. Basic understanding of SQL and data processing concepts, including data grouping and aggregation.

Course Details

Outline

Chapter 1 - Big Data Concepts and Systems Overview for Data Engineers

- Gartner's Definition of Big Data
- The Big Data Confluence Diagram
- A Practical Definition of Big Data
- Challenges Posed by Big Data
- The Traditional Client–Server Processing Pattern
- Enter Distributed Computing

- Data Physics
- Data Locality (Distributed Computing Economics)
- The CAP Theorem
- Mechanisms to Guarantee a Single CAP Property
- Eventual Consistency
- NoSQL Systems CAP Triangle
- Big Data Sharding
- Sharding Example
- Apache Hadoop
- Hadoop Ecosystem Projects
- Other Hadoop Ecosystem Projects
- Hadoop Design Principles
- Hadoop's Main Components
- Hadoop Simple Definition
- Hadoop Component Diagram
- HDFS
- Storing Raw Data in HDFS and Schema-on-Demand
- MapReduce Defined
- MapReduce Shared-Nothing Architecture
- MapReduce Phases
- The Map Phase
- The Reduce Phase
- Similarity with SQL Aggregation Operations
- Summary

Chapter 2 - Defining Data Engineering

- Data is King
- Translating Data into Operational and Business Insights
- What is Data Engineering
- The Data-Related Roles
- The Data Science Skill Sets
- The Data Engineer Role
- Core Skills and Competencies
- An Example of a Data Product

- What is Data Wrangling (Munging)?
- The Data Exchange Interoperability Options
- Summary

Chapter 3 - Data Processing Phases

- Typical Data Processing Pipeline
- Data Discovery Phase
- Data Harvesting Phase
- Data Priming Phase
- Exploratory Data Analysis
- Model Planning Phase
- Model Building Phase
- Communicating the Results
- Production Roll-out
- Data Logistics and Data Governance
- Data Processing Workflow Engines
- Apache Airflow
- Data Lineage and Provenance
- Apache NiFi
- Summary

Chapter 4 - Python 3 Introduction

- What is Python?
- Python Documentation
- Where Can I Use Python?
- Which version of Python am I running?
- Running Python Programs
- Python Shell
- Dev Tools and REPLs
- IPython
- Jupyter
- Hands-On Exercise
- The Anaconda Python Distribution
- Summary

Chapter 5 - Python Variables and Types

- Variables and Types
- More on Variables
- Assigning Multiple Values to Multiple Variables
- More on Types
- Variable Scopes
- The Layout of Python Programs
- Comments and Triple-Delimited String Literals
- Sample Python Code
- PEP8
- Getting Help on Python Objects
- Null (None)
- Strings
- Finding Index of a Substring
- String Splitting
- Raw String Literals
- String Formatting and Interpolation
- String Public Method Names
- The Boolean Type
- Boolean Operators
- Relational Operators
- Numbers
- "Easy Numbers"
- Looking Up the Runtime Type of a Variable
- Divisions
- Assignment-with-Operation
- Hands-On Exercise
- Dates and Times
- Hands-On Exercise
- Summary

Chapter 6 - Control Statements and Data Collections

- Control Flow with The if-elif-else Triad

- An if-elif-else Example
- Conditional Expressions (a.k.a. Ternary Operator)
- The While-Break-Continue Triad
- The for Loop
- The range() Function
- Examples of Using range()
- The try-except-finally Construct
- Hands-On Exercise
- The assert Expression
- Lists
- Main List Methods
- List Comprehension
- Zipping Lists
- Enumerate
- Hands-On Exercise
- Dictionaries
- Working with Dictionaries
- Other Dictionary Methods
- Sets
- Set Methods
- Set Operations
- Set Operations Examples
- Finding Unique Elements in a List
- Common Collection Functions and Operators
- Hands-On Exercise
- Tuples
- Unpacking Tuples
- Hands-On Exercise
- Summary

Chapter 7 - Functions and Modules

- Built-in Functions
- Functions
- The "Call by Sharing" Parameter Passing

- Global and Local Variable Scopes
- Default Parameters
- Named Parameters
- Dealing with Arbitrary Number of Parameters
- Keyword Function Parameters
- Hands-On Exercise
- What is Functional Programming (FP)?
- Concept: Pure Functions
- Concept: Recursion
- Concept: Higher-Order Functions
- Lambda Functions in Python
- Examples of Using Lambdas
- Lambdas in the Sorted Function
- Hands-On Exercise
- Python Modules
- Importing Modules
- Installing Modules
- Listing Methods in a Module
- Creating Your Own Modules
- Creating a Module's Entry Point
- Summary

Chapter 8 - File I/O and Useful Modules

- Reading Command-Line Parameters
- Hands-On Exercise (N/A in DCC)
- Working with Files
- Reading and Writing Files
- Hands-On Exercise
- Hands-On Exercise
- Random Numbers
- Hands-On Exercise
- Regular Expressions
- The re Object Methods
- Using Regular Expressions Examples

- Hands-On Exercise
- Summary

Chapter 9 - Practical Introduction to NumPy

- NumPy
- The First Take on NumPy Arrays
- The ndarray Data Structure
- Getting Help
- Understanding Axes
- Indexing Elements in a NumPy Array
- Understanding Types
- Re-Shaping
- Commonly Used Array Metrics
- Commonly Used Aggregate Functions
- Sorting Arrays
- Vectorization
- Vectorization Visually
- Broadcasting
- Broadcasting Visually
- Filtering
- Array Arithmetic Operations
- Reductions: Finding the Sum of Elements by Axis
- Array Slicing
- 2-D Array Slicing
- Slicing and Stepping Through
- The Linear Algebra Functions
- Summary

Chapter 10 - Practical Introduction to pandas

- What is pandas?
- The Series Object
- Accessing Values and Indexes in Series
- Setting Up Your Own Index
- Using the Series Index as a Lookup Key

- Can I Pack a Python Dictionary into a Series?
- The DataFrame Object
- The DataFrame's Value Proposition
- Creating a pandas DataFrame
- Getting DataFrame Metrics
- Accessing DataFrame Columns
- Accessing DataFrame Rows
- Accessing DataFrame Cells
- Using iloc
- Using loc
- Examples of Using loc
- DataFrames are Mutable via Object Reference!
- The Axes
- Deleting Rows and Columns
- Adding a New Column to a DataFrame
- Appending / Concatenating DataFrame and Series Objects
- Example of Appending / Concatenating DataFrames
- Re-indexing Series and DataFrames
- Getting Descriptive Statistics of DataFrame Columns
- Navigating Rows and Columns For Data Reduction
- Getting Descriptive Statistics of DataFrames
- Applying a Function
- Sorting DataFrames
- Reading From CSV Files
- Writing to the System Clipboard
- Writing to a CSV File
- Fine-Tuning the Column Data Types
- Changing the Type of a Column
- What May Go Wrong with Type Conversion
- Summary

Chapter 11 - Data Grouping and Aggregation with pandas

- Data Aggregation and Grouping
- Sample Data Set

- The pandas.core.groupby.SeriesGroupBy Object
- Grouping by Two or More Columns
- Emulating SQL's WHERE Clause
- The Pivot Tables
- Cross-Tabulation
- Summary

Chapter 12 - Repairing and Normalizing Data

- Repairing and Normalizing Data
- Dealing with the Missing Data
- Sample Data Set
- Getting Info on Null Data
- Dropping a Column
- Interpolating Missing Data in pandas
- Replacing the Missing Values with the Mean Value
- Scaling (Normalizing) the Data
- Data Preprocessing with scikit-learn
- Scaling with the scale() Function
- The MinMaxScaler Object
- Summary

Chapter 13 - Data Visualization in Python

- Data Visualization
- Data Visualization in Python
- Matplotlib
- Getting Started with matplotlib
- The matplotlib.pyplot.plot() Function
- The matplotlib.pyplot.bar() Function
- The matplotlib.pyplot.pie () Function
- The matplotlib.pyplot.subplot() Function
- A Subplot Example
- Figures
- Saving Figures to a File
- Seaborn

- Getting Started with seaborn
- Histograms and KDE
- Plotting Bivariate Distributions
- Scatter plots in seaborn
- Pair plots in seaborn
- Heatmaps
- A Seaborn Scatterplot with Varying Point Sizes and Hues
- ggplot
- Summary

Chapter 14 - Python as a Cloud Scripting Language

- Python's Value
- Python on AWS
- AWS SDK For Python (boto3)
- What is Serverless Computing?
- How Functions Work
- The AWS Lambda Event Handler
- What is AWS Glue?
- PySpark on Glue - Sample Script
- Summary

Chapter 15 - Introduction to Apache Spark

- What is Apache Spark
- The Spark Platform
- Spark vs Hadoop's MapReduce (MR)
- Common Spark Use Cases
- Languages Supported by Spark
- Running Spark on a Cluster
- The Spark Application Architecture
- The Driver Process
- The Executor and Worker Processes
- Spark Shell
- Jupyter Notebook Shell Environment
- Spark Applications

- The spark-submit Tool
- The spark-submit Tool Configuration
- Interfaces with Data Storage Systems
- The Resilient Distributed Dataset (RDD)
- Datasets and DataFrames
- Spark SQL, DataFrames, and Catalyst Optimizer
- Project Tungsten
- Spark Machine Learning Library
- Spark (Structured) Streaming
- GraphX
- Extending Spark Environment with Custom Modules and Files
- Spark 3
- Spark 3 Updates at a Glance
- Summary

Chapter 16 - The Spark Shell

- The Spark Shell
- The Spark v.2 + Command-Line Shells
- The Spark Shell UI
- Spark Shell Options
- Getting Help
- Jupyter Notebook Shell Environment
- Example of a Jupyter Notebook Web UI (Databricks Cloud)
- The Spark Context (sc) and Spark Session (spark)
- Creating a Spark Session Object in Spark Applications
- The Shell Spark Context Object (sc)
- The Shell Spark Session Object (spark)
- Loading Files
- Saving Files
- Summary

Chapter 17 - Spark RDDs

- The Resilient Distributed Dataset (RDD)
- Ways to Create an RDD

- Supported Data Types
- RDD Operations
- RDDs are Immutable
- Spark Actions
- RDD Transformations
- Other RDD Operations
- Chaining RDD Operations
- RDD Lineage
- The Big Picture
- What May Go Wrong
- Miscellaneous Pair RDD Operations
- RDD Caching
- Summary

Chapter 18 - Parallel Data Processing with Spark

- Running Spark on a Cluster
- Data Partitioning
- Data Partitioning Diagram
- Single Local File System RDD Partitioning
- Multiple File RDD Partitioning
- Special Cases for Small-sized Files
- Parallel Data Processing of Partitions
- Spark Application, Jobs, and Tasks
- Stages and Shuffles
- The "Big Picture"
- Summary

Chapter 19 - Introduction to Spark SQL

- What is Spark SQL?
- Uniform Data Access with Spark SQL
- Using JDBC Sources
- Hive Integration
- What is a DataFrame?
- Creating a DataFrame in PySpark

- Creating a DataFrame in PySpark (Cont'd)
- Commonly Used DataFrame Methods and Properties in PySpark
- Commonly Used DataFrame Methods and Properties in PySpark (Cont'd)
- Grouping and Aggregation in PySpark
- The "DataFrame to RDD" Bridge in PySpark
- The SQLContext Object
- Examples of Spark SQL / DataFrame (PySpark Example)
- Converting an RDD to a DataFrame Example
- Example of Reading / Writing a JSON File
- Performance, Scalability, and Fault-tolerance of Spark SQL
- Summary

Lab Exercises

- Lab 1. Data Availability and Consistency
- Lab 2. A/B Testing Data Engineering Tasks Project
- Lab 3. Learning the Databricks Community Cloud Lab Environment
- Lab 4. Python Variables
- Lab 5. Dates and Times
- Lab 6. The if, for, and try Constructs
- Lab 7. Understanding Lists
- Lab 8. Dictionaries
- Lab 9. Sets
- Lab 10. Tuples
- Lab 11. Functions
- Lab 12. Functional Programming
- Lab 13. File I/O
- Lab 14. Using HTTP and JSON
- Lab 15. Random Numbers
- Lab 16. Regular Expressions
- Lab 17. Understanding NumPy
- Lab 18. A NumPy Project
- Lab 19. Understanding pandas
- Lab 20. Data Grouping and Aggregation
- Lab 21. Repairing and Normalizing Data

- Lab 22. Data Visualization and EDA with pandas and seaborn
- Lab 23. Correlating Cause and Effect
- Lab 24. Learning PySpark Shell Environment
- Lab 25. Understanding Spark DataFrames
- Lab 26. Learning the PySpark DataFrame API
- Lab 27. Data Repair and Normalization in PySpark