Course duration

- 3 days

Course Benefits

- Data engineering practice
- High-octane introduction to Python
- Technical reviews of NumPy, pandas, and other Python libraries and data processing systems
- Data visualization and exploratory data analysis
- Data repairing and normalization
- Understanding the data needs and requirements of Machine Learning and Data Science projects
- Python in the Cloud
- Python on Hadoop (PySpark)

Course Outline

1. Defining Data Engineering
    1. Data is King
    2. Translating Data into Operational and Business Insights
    3. What is Data Engineering
    4. The Data-Related Roles
    5. The Data Science Skill Sets
    6. The Data Engineer Role
    7. Core Skills and Competencies
    8. An Example of a Data Product
    9. What is Data Wrangling (Munging)?
    10. The Data Exchange Interoperability Options
    11. Summary
2. Distributed Computing Concepts for Data Engineers
    1. The Traditional Client–Server Processing Pattern
    2. Enter Distributed Computing
    3. Data Physics
    4. Data Locality (Distributed Computing Economics)
    5. The CAP Theorem
    6. Mechanisms to Guarantee a Single CAP Property
    7. Eventual Consistency
    8. Summary
3. Data Processing Phases
    1. Typical Data Processing Pipeline

2. Data Discovery Phase
3. Data Harvesting Phase
4. Data Priming Phase
5. Exploratory Data Analysis
6. Model Planning Phase
7. Model Building Phase
8. Communicating the Results
9. Production Roll-out
10. Data Logistics and Data Governance
11. Data Processing Workflow Engines
12. Apache Airflow
13. Data Lineage and Provenance
14. Apache NiFi
15. Summary

4. Quick Introduction to Python for Data Engineers
    1. What is Python?
    2. Additional Documentation
    3. Which version of Python am I running?
    4. Python Dev Tools and REPLs
    5. IPython
    6. Jupyter
    7. Jupyter Operation Modes
    8. Jupyter Common Commands
    9. Anaconda
    10. Python Variables and Basic Syntax
    11. Variable Scopes
    12. PEP8
    13. The Python Programs
    14. Getting Help
    15. Variable Types
    16. Assigning Multiple Values to Multiple Variables
    17. Null (None)
    18. Strings
    19. Finding Index of a Substring
    20. String Splitting
    21. Triple-Delimited String Literals
    22. Raw String Literals
    23. String Formatting and Interpolation
    24. Boolean
    25. Boolean Operators
    26. Numbers
    27. Looking Up the Runtime Type of a Variable
    28. Divisions
    29. Assignment-with-Operation
    30. Dates and Times
    31. Comments:
    32. Relational Operators

33. The if-elif-else Triad
34. An if-elif-else Example
35. Conditional Expressions (a.k.a. Ternary Operator)
36. The While-Break-Continue Triad
37. The for Loop
38. try-except-finally
39. Lists
40. Main List Methods
41. Dictionaries
42. Working with Dictionaries
43. Sets
44. Common Set Operations
45. Set Operations Examples
46. Finding Unique Elements in a List
47. Enumerate
48. Tuples
49. Unpacking Tuples
50. Functions
51. Dealing with Arbitrary Number of Parameters
52. Keyword Function Parameters
53. The range Object
54. Random Numbers
55. Python Modules
56. Importing Modules
57. Installing Modules
58. Listing Methods in a Module
59. Creating Your Own Modules
60. Creating a Runnable Application
61. List Comprehension
62. Zipping Lists
63. Working with Files
64. Reading and Writing Files
65. Reading Command-Line Parameters
66. Accessing Environment Variables
67. What is Functional Programming (FP)?
68. Terminology: Higher-Order Functions
69. Lambda Functions in Python
70. Example: Lambdas in the Sorted Function
71. Other Examples of Using Lambdas
72. Regular Expressions
73. Using Regular Expressions Examples
74. Python Data Science-Centric Libraries
75. Summary

5. Practical Introduction to NumPy
    1. SciPy
    2. NumPy
    3. The First Take on NumPy Arrays

4. Getting Help
5. Understanding Axes
6. Indexing Elements in a NumPy Array
7. NumPy Arrays
8. Understanding Types
9. Re-Shaping
10. Commonly Used Array Metrics
11. Commonly Used Aggregate Functions
12. Sorting Arrays
13. Vectorization
14. Broadcasting
15. Filtering
16. Array Arithmetic Operations
17. Array Slicing
18. 2-D Array Slicing
19. The Linear Algebra Functions
20. Summary

6. Practical Introduction to Pandas
    1. What is pandas?
    2. The Series Object
    3. Accessing Values and Indexes in Series
    4. Setting Up Your Own Index
    5. Using the Series Index as a Lookup Key
    6. Can I Pack a Python Dictionary into a Series?
    7. The DataFrame Object
    8. The DataFrame's Value Proposition
    9. Creating a pandas DataFrame
    10. Getting DataFrame Metrics
    11. Accessing DataFrame Columns
    12. Accessing DataFrame Rows
    13. Accessing DataFrame Cells
    14. Using iloc
    15. Using loc
    16. Examples of Using loc
    17. DataFrames are Mutable via Object Reference!
    18. Deleting Rows and Columns
    19. Adding a New Column to a DataFrame
    20. Appending / Concatenating DataFrame and Series Objects
    21. Example of Appending / Concatenating DataFrames
    22. Re-indexing Series and DataFrames
    23. Getting Descriptive Statistics of DataFrame Columns
    24. Getting Descriptive Statistics of DataFrames
    25. Applying a Function
    26. Sorting DataFrames
    27. Reading From CSV Files
    28. Writing to the System Clipboard
    29. Writing to a CSV File

30. Fine-Tuning the Column Data Types
31. Changing the Type of a Column
32. What May Go Wrong with Type Conversion
33. Summary

7. Descriptive Statistics Computing Features in Python
    1. Descriptive Statistics
    2. Non-uniformity of a Probability Distribution
    3. Using NumPy for Calculating Descriptive Statistics Measures
    4. Finding Min and Max in NumPy
    5. Using pandas for Calculating Descriptive Statistics Measures
    6. Correlation
    7. Regression and Correlation
    8. Covariance
    9. Getting Pairwise Correlation and Covariance Measures
    10. Finding Min and Max in pandas DataFrame
    11. Summary
8. Data Grouping and Aggregation with pandas
    1. Data Aggregation and Grouping
    2. Sample Data Set
    3. The pandas.core.groupby.SeriesGroupBy Object
    4. Grouping by Two or More Columns
    5. Emulating SQL's WHERE Clause
    6. The Pivot Tables
    7. Cross-Tabulation
    8. Summary
9. Repairing and Normalizing Data
    1. Repairing and Normalizing Data
    2. Dealing with the Missing Data
    3. Sample Data Set
    4. Getting Info on Null Data
    5. Dropping a Column
    6. Interpolating Missing Data in pandas
    7. Replacing the Missing Values with the Mean Value
    8. Scaling (Normalizing) the Data
    9. Data Preprocessing with scikit-learn
    10. Scaling with the scale() Function
    11. The MinMaxScaler Object
    12. Summary
10. Data Visualization in Python using matplotlib
    1. Data Visualization
    2. What is matplotlib?
    3. Getting Started with matplotlib
    4. The matplotlib.pyplot.plot() Function
    5. The matplotlib.pyplot.scatter() Function
    6. Labels and Titles
    7. Styles
    8. The matplotlib.pyplot.bar() Function

9. The matplotlib.pyplot.hist () Function
10. The matplotlib.pyplot.pie () Function
11. The Figure Object
12. The matplotlib.pyplot.subplot() Function
13. Selecting a Grid Cell
14. Saving Figures to a File
15. Summary

11. Parallel Data Processing with PySpark
    1. What is Apache Spark
    2. The Spark Platform
    3. Languages Supported by Spark
    4. Running Spark on a Cluster
    5. The Spark Shell
    6. The High-Level Execution Flow in Stand-alone Spark Cluster
    7. The Spark Application Architecture
    8. The Resilient Distributed Dataset (RDD)
    9. The Lineage Concept
    10. Datasets and DataFrames
    11. Data Partitioning
    12. Data Partitioning Diagram
    13. Finding the Most Frequently Used Words in PySpark
    14. Summary
12. Python as a Cloud Scripting Language
    1. Python's Value
    2. Python on AWS
    3. AWS SDK For Python (boto3)
    4. What is Serverless Computing?
    5. How Functions Work
    6. The AWS Lambda Event Handler
    7. What is AWS Glue?
    8. PySpark on Glue - Sample Script
    9. Summary

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Python class:

- Practical experience coding in one or more modern programming languages.
- Ability to quickly learn the new material, reinforce the knowledge of a learned topic by doing programming exercises (labs), and then apply knowledge in data engineering mini projects.

Experience in the following *would be useful* for this Python class:

- Knowledge of Python is desirable but not necessary.