

Szkolenie: Cloudera
Cloudera Data Scientist Training

CLOUDERA
Training Partner

DOSTĘPNE TERMINY

2022-12-12 | 4 dni | Warszawa / Virtual Classroom
2022-12-12 | 4 dni | Warszawa / Wirtualna sala
2023-02-20 | 4 dni | Kraków / Virtual Classroom
2023-02-20 | 4 dni | Kraków / Wirtualna sala
2023-04-17 | 4 dni | Warszawa / Virtual Classroom
2023-04-17 | 4 dni | Warszawa / Wirtualna sala

Cel szkolenia:

This four-day workshop covers enterprise data science and machine learning using Apache Spark in Cloudera Data Science Workbench (CDSW). Participants use Spark SQL to load, explore, cleanse, join, and analyze data and Spark MLlib to specify, train, evaluate, tune, and deploy machine learning pipelines. They dive into the foundations of the Spark architecture and execution model necessary to effectively configure, monitor, and tune their Spark applications. Participants also learn how Spark integrates with key components of the Cloudera platform such as HDFS, YARN, Hive, Impala, and Hue as well as their favorite Python or R packages.

Through narrated lecture, recorded demonstrations, and hands-on exercises, you will learn how to:

- How to use Apache Spark to run data science and machine learning workflows at scale
- How to use Spark SQL and DataFrames to work with structured data
- How to use MLlib, Spark's machine learning library
- How to use PySpark, Spark's Python API
- How to use sparklyr, a dplyr-compatible R interface to Spark
- How to use Cloudera Data Science Workbench (CDSW)
- How to use other Cloudera platform components including HDFS, Hive, Impala, and Hue

Plan szkolenia:

- **Data Science Overview**
 - What Data Scientists Do

- What Process Data Scientists Use
- What Tools Data Scientists Use
- **Cloudera Data Science Workbench (CDSW)**
 - Introduction to Cloudera Data
- **Science Workbench**
 - How Cloudera Data Science
- **Workbench Works**
 - How to Use Cloudera Data Science
- **Workbench**
 - Entering Code
 - Getting Help
 - Accessing the Linux Command Line
 - Working with Python Packages
 - Formatting Session Output
- **Case Study**
 - DuoCar
 - How DuoCar Works
 - DuoCar Datasets
 - DuoCar Business Goals
 - DuoCar Data Science Platform
 - DuoCar Cloudera EDH Cluster
 - HDFS
 - Apache Spark
 - Apache Hive
 - Apache Impala
 - Hue
 - YARN
 - DuoCar Cluster Architecture
- **Apache Spark**
 - Apache Spark
 - How Spark Works
 - The Spark Stack
 - Spark SQL
 - DataFrames
 - File Formats in Apache Spark
 - Text File Formats

- Parquet File Format
- **Summarizing and Grouping DataFrames**
 - Summarizing Data with Aggregate
 - Functions
 - Grouping Data
 - Pivoting Data
- **Window Functions**
 - Introduction to Window Functions
 - Creating a Window Specification
 - Aggregating over a Window Specification
- **Exploring DataFrames**
 - Possible Workflows for Big Data
 - Exploring a Single Variable
 - Exploring a Categorical Variable
 - Exploring a Continuous Variable
 - Exploring a Pair of Variables
 - Categorical-Categorical Pair
 - Categorical-Continuous Pair
 - Continuous-Continuous Pair
- **Apache Spark Job Execution**
 - DataFrame Operations
 - Input Splits
 - Narrow Operations
 - Wide Operations
 - Stages and Tasks
 - Shuffle
- **Processing Text and Training and Evaluating Topic Models**
 - Introduction to Topic Models
 - Scenario
 - Extracting and Transforming Features
 - Parsing Text Data
 - Removing Common (Stop) Words
 - Counting the Frequency of Words
 - Specifying a Topic Model
 - Training a topic model using Latent Dirichlet Allocation (LDA)
 - Assessing the Topic Model Fit

- Examining a Topic Model
- Applying a Topic Model
- **Training and Evaluating Recommender Models**
 - Introduction to Recommender Models
 - Scenario
 - Preparing Data for a Recommender Model
 - Specifying a Recommender Model
 - Spark Interface Languages
 - PySpark
 - Data Science with PySpark
 - sparklyr
 - dplyr and sparklyr
 - Comparison of PySpark and sparklyr
 - How sparklyr Works with dplyr
 - sparklyr DataFrame and MLib Functions
 - When to Use PySpark and sparklyr
- **Running a Spark Application from (CDSW)**
 - Overview
 - Starting a Spark Application
 - Reading Data into a Spark SQL Data Frame
 - Examining the Schema of a Data Frame
 - Computing the Number of Rows and
- **Columns of a DataFrame**
 - Examining Rows of a DataFrame
 - Stopping a Spark Application
- **Inspecting a Spark SQL DataFrame**
 - Overview
 - Inspecting a DataFrame
 - Inspecting a DataFrame Column
 - Inspecting a Primary Key Variable
 - Inspecting a Categorical Variable
 - Inspecting a Numerical Variable
 - Inspecting a Date and Time Variable
- **Transforming DataFrames**
 - Spark SQL DataFrames
 - Working with Columns

- Selecting Columns
- Dropping Columns
- Specifying Columns
- Adding Columns
- Changing the Column Name
- Changing the Column Type
- **Monitoring, Tuning, and Configuring Spark Applications**
 - Monitoring Spark Applications
 - Persisting DataFrames
 - Partitioning DataFrames
 - Configuring the Spark Environment
- **Machine Learning Overview**
 - Machine Learning
 - Underfitting and Overfitting
 - Model Validation
 - Hyperparameters
 - Supervised and Unsupervised Learning
 - Machine Learning Algorithms
 - Machine Learning Libraries
 - Apache Spark MLlib
- **Training and Evaluating Regression Models**
 - Introduction to Regression Models
 - Scenario
 - Preparing the Regression Data
 - Assembling the Feature Vector
 - Creating a Train and Test Set
 - Specifying a Linear Regression Model
 - Training a Linear Regression Model
 - Examining the Model Parameters
 - Examining Various Model Performance Measures
 - Examining Various Model Diagnostics
 - Applying the Linear Regression Model to the Test Data
 - Evaluating the Linear Regression Model on the Test Data
 - Plotting the Linear Regression Model
 - Training a Recommender Model using Alternating Least Squares
 - Examining a Recommender Model

- Applying a Recommender Model
- Evaluating a Recommender Model
- Generating Recommendations
- **Working with Machine Learning Pipelines**
 - Specifying Pipeline Stages
 - Specifying a Pipeline
 - Training a Pipeline Model
 - Querying a Pipeline Model
 - Applying a Pipeline Model
- **Deploying Machine Learning Pipelines**
 - Saving and Loading Pipelines and Pipeline Models in Python
 - Loading Pipelines and Pipeline Models in Scala
 - Working with Rows
 - Ordering Rows
 - Selecting a Fixed Number of Rows
 - Selecting Distinct Rows
 - Filtering Rows
 - Sampling Rows
 - Working with Missing Values
- **Transforming DataFrame Columns**
 - Spark SQL Data Types
 - Working with Numerical Columns
 - Working with String Columns
 - Working with Date and Timestamp Columns
 - Working with Boolean Columns
- **Complex Types**
 - Complex Collection Data Types
 - Arrays
 - Maps
 - Structs
- **User-Defined Functions**
 - User-Defined Functions
 - Defining a Python Function
 - Registering a Python Function as a
 - User-Defined Function
 - Applying a User-Defined Function

- **Reading and Writing Data**
 - Reading and Writing Data
 - Working with Delimited Text Files
 - Working with Text Files
 - Working with Parquet Files
 - Working with Hive Tables
 - Working with Object Stores
 - Working with pandas DataFrames
- **Combining and Splitting DataFrames**
 - Joining DataFrames
 - Cross Join
 - Inner Join
 - Left Semi Join
 - Left Anti Join
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join
 - Applying Set Operations to DataFrames
 - Splitting a DataFrame
- **Training and Evaluating Classification Models**
 - Introduction to Classification Models
 - Scenario
 - Preprocessing the Modeling Data
 - Generate a Label
 - Extract, Transform, And Select Features
 - Create Train and Test Sets
 - Specify A Logistic Regression Model
 - Train the Logistic Regression Model
 - Examine the Logistic Regression Model
 - Evaluate Model Performance on the Test Set
- **Tuning Algorithm Hyperparameters Using Grid Search**
 - Requirements for Hyperparameter Tuning
 - Specifying the Estimator
 - Specifying the Hyperparameter Grid
 - Specifying the Evaluator

- Tuning Hyperparameters using Holdout Cross-validation
- Tuning Hyperparameters using K-fold Cross-validation
- **Training and Evaluating Clustering Models**
 - Introduction to Clustering
 - Scenario
 - Preprocessing the Data
 - Extracting, Transforming, and Selecting Features
 - Specifying a Gaussian Mixture Model
 - Training a Gaussian Mixture Model
 - Examining the Gaussian Mixture Model
 - Plotting the Clusters
 - Exploring the Cluster Profiles
 - Saving and Loading the Gaussian Mixture Model
- **Overview of sparklyr**
 - Connecting to Spark
 - Reading Data
 - Inspecting Data
 - Transforming Data Using dplyr Verbs
 - Using SQL Queries
 - Spark DataFrames Functions
 - Visualizing Data from Spark
 - Machine Learning with MLlib
- **Introduction to Additional CDSW Features**
 - Collaboration
 - Jobs
 - Experiments
 - Models
 - Applications

Wymagania:

The workshop is designed for data scientists who use Python or R to work with small datasets on a single machine and who need to scale up their data science and machine learning workflows to large datasets on distributed clusters. Data engineers, data analysts, developers, and solution architects who collaborate with data scientists will also find this workshop valuable. Workshop participants walk through an end-to-end data science and machine learning workflow based on realistic scenarios and

datasets from a fictitious technology company. The material is presented through a sequence of brief lectures, interactive demonstrations, extensive hands-on exercises, and lively discussions. The demonstrations and exercises are conducted in Python (with PySpark) using Cloudera Data Science Workbench (CDSW). Supplemental examples using R (with sparklyr) are provided.

Poziom trudności



Prowadzący:

Certified Cloudera Instructor.