Oracle Machine Learning for R

Data Management

DURATION	MODULES	COURSE CODE
3 Days	**11 Lectures**	**—**

Course Overview

In this course, you will learn about Oracle Machine Learning for R, and learn how to use it to develop machine learning solutions.

What You Will Learn

- 1 Overview of Oracle Machine Learning for R
- Machine Learning
- Objectives
- Introduction to Oracle Machine Learning
- Oracle Machine Learning Product Family
- Oracle Machine Learning Notebooks
- Oracle Machine Learning for SQL
- Oracle Machine Learning for Python
- Oracle Machine Learning Services
- OML Services
- Oracle Data Miner User Interface
- Oracle Machine Learning: Key Attributes
- Oracle Machine Learning Algorithms and Analytics
- Why Oracle for Machine Learning?
- R Implementation in Finance
- What Is R?
- R Environment
- Oracle Machine Learning for R: Features
- Oracle Machine Learning for R: Benefits
- Oracle R Distribution
- ROracle Package
- Requirements for ROracle
- Third-Party RStudio IDE
- OML4R Architectural Components
- OML4R: Efficiency All the Way!
- Oracle Machine Learning for R: Installation Steps

- Roadmap to Install OML4R
- OML4R: Installation Environment
- OML4R: Installing Oracle R Distribution
- OML4R: Install Oracle Machine Learning for R Server
- OML4R: Install Oracle R Distribution on Client
- Oracle Machine Learning for R Packages in Client
- How to Start OML4R in R Console from Client
- Getting Help on OML4R Classes, Functions, and Methods
- Getting Information on OML4R Classes, Functions, and Methods: Examples
- Oracle Machine Learning for R: Global Options
- Summary
- 2 OML4R Transparency Layer: Introduction
- Objectives
- Transparency Layer: Overview
- Invoking In-Database Aggregation Function
- Overloads Graphics Functions for In-Database Statistics
- Connecting to an Oracle Database Instance
- Using ore.connect Function
- Options for Connecting to Oracle Database
- Using the ore.disconnect Function
- Adding Schema with ore.attach
- ore.frame Class
- Data Types and Classes: Mapping R to Database Types
- Working with Overloaded Functions in R
- Support for Time Series Data Preparation
- Time Series Data Preparation: Using Date and Time Arithmetic
- Time Series Data Preparation: Comparing Dates and Times
- In-Database Sampling
- In-Database Sampling Techniques
- In-Database Sampling: Simple Random Sampling
- In-Database Sampling: Split Data Sampling
- In-Database Sampling: Systematic Sampling
- In-Database Sampling: Stratified Sampling
- In-Database Sampling: Cluster Sampling
- In-Database Sampling: Quota Sampling
- Random Partitioning
- Support for R Naming Conventions
- Coercing R and Oracle Machine Learning for R Class Types
- Ordering Framework: Creating Ordered and Unordered ora.frame Objects
- Ordering Framework: ora.frame Ordered
- Ordering Framework: ora.frame Unordered
- Ordering Framework: Using Keys
- Ordering Framework: Using Row Names
- Ordering Framework: Using Ordered Frames
- Global Options
- Summary

- 3 OML4R Transparency Layer: Create and Manage R Objects in Oracle Database
- Objectives
- R Object Persistence
- R Object Persistence with OML4R
- R Object Persistence: Advantages
- OML4R Data Stores
- ore.save() Function
- ore.load() Function
- ore.delete() Function
- ore.datastore() Function
- ore.datastoreSummary() Function
- Create R Objects for In-Database Data
- Synchronize Data with ore.sync() Function
- Getting Objects with ore.get Function
- Move Data to and from the Database
- Ore.push Function
- Ore.pull Function
- Creating Database Tables
- Deleting Database Tables
- Ore.exists Function
- Summary
- 4 OML4R Transparency Layer: Data Preparation and Data Manipulation
- Data Preparation
- Learning Objectives
- Preparing data in Database
- Select Data
- Selecting Data by Column
- Select Data by Row
- Select Data by Value
- Using row indexing
- Indexing an ore.frame Object
- Combine Data
- Aggregate Data
- Transform Data
- Exploratory Data Analysis Functions
- dplyr
- OREdplyr
- Functionality of OREdplyr
- OREdplyr functions
- Examples: Selecting Columns
- Examples: Programming with select_
- Examples: Selecting Distinct Columns
- Examples: Selecting Rows by Position
- Examples: Arranging Columns
- Examples: Filtering Columns
- Examples: Mutating Columns

- Examples: Joining Rows
- Examples: Group Columns and Rows
- Examples: Aggregate Columns and Rows
- Examples: Sample Rows
- Examples: Rank Rows
- Select and Order Data
- Using Third-Party Packages on the R Client
- Installing and Loading a Third-Party Package on the R Client
- Summary
- 5 OML4R Embedded R Execution – R Interface
- Objectives
- Embedded R Execution
- OML4R Embedded R Execution: R Interface Introduction
- Benefits of Embedded R Execution
- API for Embedded R Execution
- User-Defined R Functions for Embedded R Execution
- Installing Package for a Single Database
- Running User-Defined R Functions Using R Interface
- Functionality of Automatic Connection
- Automatic Connection in Embedded R Scripts
- Example to Check Connection
- Using the ore.doEval Function
- Using the tableApply Function
- Using the groupApply Function
- Using the rowApply Function
- Using the indexApply Function
- OML4R-Defined Graphics Function
- Viewing database server-generated graphics in client
- Using ore.indexApply for simulation
- OML4R-Defined Graphics Function Examples
- Summary
- 6 OML4R Embedded R Execution – SQL Interface
- Objectives
- SQL Interface for Embedded R Execution
- Create a User-Defined R Function Using the SQL Interface
- SQL API for Oracle Machine Learning for R
- rqEval Function
- Embedded R Execution: SQL Interface
- rqTableEval: Singleton/Real-Time Scoring
- “rqGroupEval” Functionality
- “rqGroupEval” Build and rqRowEval Score
- Parameters of the SQL Table Functions
- Returning R Statistical Results as a Database Table
- Wrap in Function, Invoke from ore.tableApply
- Determine What Results We Really Need
- Create Script in Repository Using SQL

- Invoke from SQL Select Statement
- Manage User-Defined R Functions Using the SQL Interface
- Using Access Privileges with User-Defined Functions
- Manage User-Defined R Functions and Datastores in SQL
- PL/SQL Procedures for Managing R Scripts and Datastores
- Data Dictionary Views for Datastores
- Manage Datastores in SQL – Example
- Return Value of Embedded Execution SQL Table Functions Using Pre-Defined
- Graphics Function
- Summary
- 7 Modeling in OML4R: Part 1
- Objectives
- Supported ML Techniques
- ORE Package Overview
- OREdm Package
- OREdm Algorithm Input Requirements
- OREdm Features
- OREdm Algorithms
- Association Rules – Apriori Algorithm
- Using the ore.odmAssocRules Function
- Attribute Importance – Minimum Description Length
- Classification – Decision Tree
- Clustering – Expectation Maximization
- Using the ore.odmEM Function: Example
- Using the ore.odmEM Function: Output
- Feature Extraction – Explicit Semantic Analysis (ESA)
- Using the ore.odmESA Function
- Generalized Linear Models
- Building a Linear Regression Model
- Using Ridge Estimation for Coefficients of ore.odmGLM Model
- Building a Logistic Regression GLM
- Specifying a Reference Value in Building a Logistic Regression GLM
- Clustering – K-Means
- Clustering – K-Means Features
- Using the ore.odmKM Function: Example
- Using the ore.odmKM Function: Output
- Classification – Naïve Bayes
- Using the ore.odmNB Function
- Feature Extraction Non-negative Matrix Factorization (NMF)
- Using the ore.odmNMF Function
- Clustering – Orthogonal Partitioning
- Limitation of Orthogonal Partitioning
- Using the ore.odmOC Function
- Feature Extraction – Singular Value Decomposition
- Using the ore.odmSVD Function
- Support Vector Machine

- Using the ore.odmSVM Function
- Partitioned Models
- Summary
- 8
- Modeling in OML4R: Part 2
- Objectives
- OREmodels Package
- Linear Model
- Linear Model Example
- Stepwise Linear Regression
- Stepwise Regression Example
- Generalized Linear Models
- Using the ore.glm Function
- Neural Networks
- Using ore.neural and Specifying Activations
- Random Forest
- Using ore.randomForest
- Singular Value Decomposition
- SVD Example Using ore.frame
- Visualization Function: Example
- Principal Component Analysis
- Using the prcomp and princomp Functions
- Summary
- 9 Working with ROracle
- Objectives
- Overview of ROracle
- Reading Database Table to R data.frame
- Writing Database Table from R data.frame
- ROracle 1.3-2 Enhancements
- Connect to an extproc for Use within OML4R Embedded R Execution
- Unload Driver
- Get Info Methods
- Read/Write Table Methods
- Using Attributes to Map NCHAR, CLOB, BLOB, NCLOB Columns Correctly with
- dbWriteTable
- Insert date and time data into a table
- Insert and update data in a table
- Send Query Methods
- Invoking Stored Procedures
- Summary
- 10 OML4R Statistics Engine
- Objectives
- Statistics Engine Overview
- OML4R Statistics Engine
- Significance Tests
- Distribution Functions

- Additional Functions
- Base SAS Equivalent Functionality
- OML4R Statistical Functions
- ore.summary
- Statistics Supported by ore.summary
- ore.summary Parameters
- ore.summary Examples
- ore.rank
- ore.rank Parameters
- ore.rank Examples
- ore.sort
- ore.sort Parameters
- ore.sort Examples
- Example Output
- ore.corr
- ore.corr Parameters
- ore.corr Examples
- ore.crosstab
- ore.crosstab Parameters
- ore.corr Examples
- ore.crosstab Examples
- ore.freq
- ore.freq Parameters
- ore.freq Examples
- ore.esm
- ore.esm Example
- ore.esm Output
- ore.univariate
- ore.univariate Parameters
- ore.univariate Examples
- Summary
- 11 OML4R Best Practices
- Objectives
- Managing Memory Limits
- Computing and Setting Memory Limits for Embedded R
- Open Source Packages
- Should I use a third-party package on database data?
- Machine Learning Interface: Benefits
- Transparency Layer and Machine Learning: Memory Management
- Considerations
- Explicitly Specifying Oracle Database Parallelism
- Explicitly Specifying OML4R Parallelism
- Embedded R Execution Initial Memory Management Considerations
- Memory Consumed in R
- Datastore: Benefits
- Datastore: R Object Persistence

- Object Migration
- Summary