This session will be recorded

Oracle Machine Learning

AskTOM Office Hours – Product Overview

Introducing Oracle Machine Learning for R for Oracle Autonomous Database

Mark Hornick and Sherry LaMonica
Product Management, Oracle Machine Learning

Copyright © 2022, Oracle and/or its affiliates.

Introducing Oracle Machine Learning for R for Oracle Autonomous Database

Speakers: Mark Hornick and Sherry LaMonica, OML Product Management, Oracle

Join us to learn about Oracle Machine Learning for R (OML4R) for Oracle Autonomous Database. OML4R leverages the database as a high-performance computing environment to explore, transform, and analyze data faster and at scale, while allowing the use of familiar R syntax and semantics. The in-database parallelized machine learning algorithms are exposed through a well-integrated R interface. Further, data scientists and other R users can store and manage user-defined R functions as well as R objects directly in the database – as opposed to being managed in flat files. These features facilitate collaboration across the data science team by enabling convenient hand-off of data science work products from data scientists to application developers for immediate deployment. Run user-defined R functions in database-spawned and managed R engines, with system-supported data-parallel and task-parallel options. This session includes a demonstration through OML Notebooks.

Poll #1: Objective

What is your primary objective for today's session?

- Learn about cool new features in ADB
- Learn how to use R for in-database machine learning
- Get a deeper understanding of OML4R on Autonomous Database
- Other, please specify in Zoom chat

ORACLE

Oracle Machine Learning for R

Introduction

Mark Hornick

Senior Director, Oracle Machine Learning Product Management

August 2022

Copyright © 2022, Oracle and/or its affiliates

New!

OML4R is now available on Oracle Autonomous Database

Use Oracle Autonomous Database as a high-performance computing environment

- Explore, transform, and analyze data faster and at scale, even with ADB auto-scale

Use in-database parallelized and distributed machine learning algorithms

- Build more models on more data, and score large volume data – faster
- Use in-database algorithms from OML4SQL via well-integrated R API
- Now includes in-database algorithms for neural networks, random forest, exponential smooth, and xgboost from OML4SQL
- Increase productivity from automatic data preparation, partitioned models, and integrated text mining capabilities

Run user-defined R functions in database-spawned and managed R engines and manage R objects in the database

- Supports ML team collaboration – easily hand-off data science work products from data scientists to developers
- Run user-defined functions with system-supported data-parallel and task-parallel features
- Use third-party packages supplied with ADB today, and coming soon…user-specified third-party packages
- Return structured and image results in R, SQL, and REST APIs

OML Notebooks supports the Oracle R Distribution 4.0.5 interpreter

- Use R, SQL, and Python paragraphs in the same notebook

Copyright © 2022, Oracle and/or its affiliates

Sample of common enterprise machine learning pain points

“It takes too long to get my data or to get the ‘right’ data”

“I can’t analyze or mine all of my data – it has to be sampled”

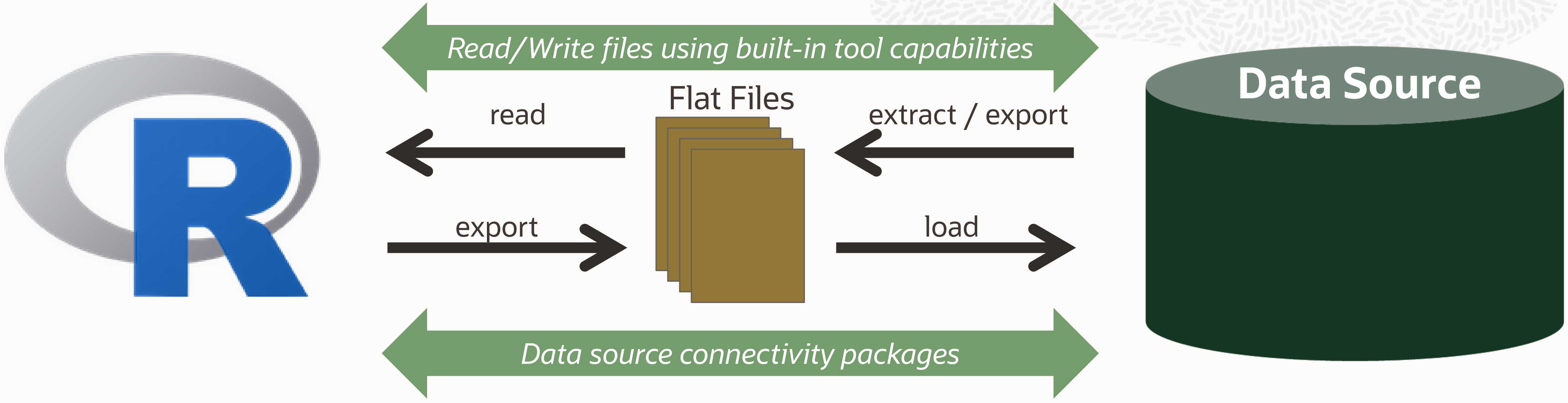
“Putting open source models and results into production takes too long and is ad hoc and complex”

“Our company is concerned about data security, backup and recovery”

“We need to build and score with 100s or 1000s of models fast to meet business objectives”

 Copyright © 2022 Oracle and/or its affiliates.

Traditional analytics and data source interaction



Deployment
Ad hoc
cron job

Access latency
Paradigm shift: R → *Data Access Language* → R
Memory limitation – data size, in-memory processing
Single threaded
Issues for backup, recovery, security
Ad hoc production deployment

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Oracle Machine Learning

Interfaces for 3 popular data science languages: SQL, R, and Python

- **OML4SQL**
- **OML4R**
- **OML4Py**

OML Notebooks — Collaborative notebook environment based on Apache Zeppelin with Autonomous Database

Oracle Data Miner — SQL Developer extension to create, schedule, and deploy ML solutions through a drag-and-drop interface

OML4Spark — ML for the big data environment from R with scalable algorithms

OML AutoML UI — No-code AutoML interface on Autonomous Database

OML Services — Model Deployment and Management, Cognitive Text on Autonomous Database

Copyright © 2022, Oracle and/or its affiliates . All rights reserved.

Poll #2: Usage

Which of these OML components do you currently use? (select all that apply)

- OML4SQL
- OML4Py
- OML4R
- OML Notebooks
- OML AutoML UI

If something else, please specify in Zoom chat.

Oracle Machine Learning for SQL

Foundation for OML4R in-database machine learning modeling

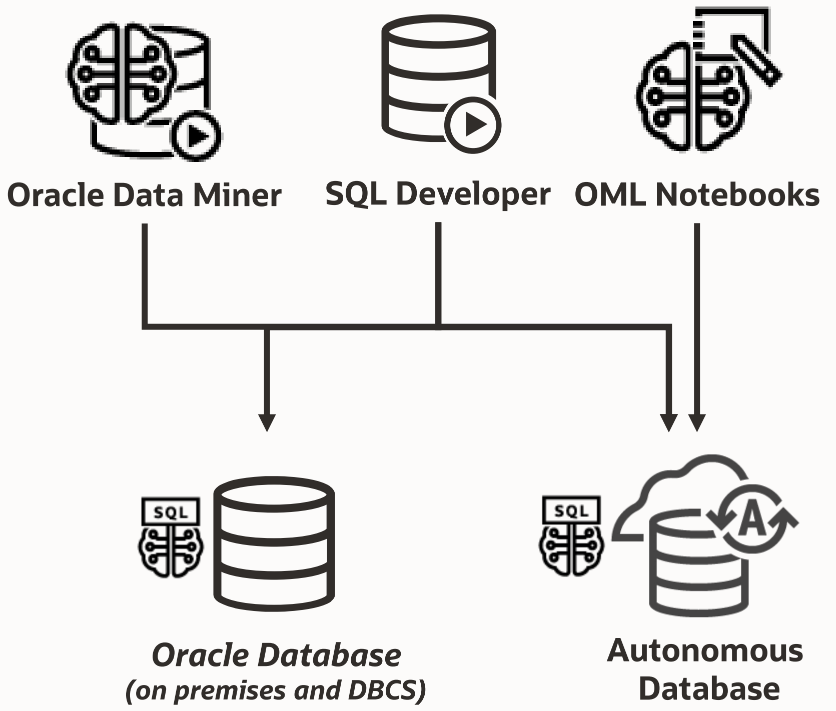
In-database, parallelized, distributed algorithms

- No extracting data to separate ML engine
- Fast and scalable
- Batch and real-time scoring at scale that leverages Exadata storage-tier function pushdown
- Algorithm-specific automatic data preparation
- Explanatory prediction details

ML models as first-class database objects

- Access control per model
- Audit user actions
- Export / import models across databases
- Ease of backup, recovery, and security

Faster time-to-market through immediate solution deployment



 Copyright © 2022, Oracle and/or its affiliates . All rights reserved.

Oracle Machine Learning for R

Empower data scientists with open source environments

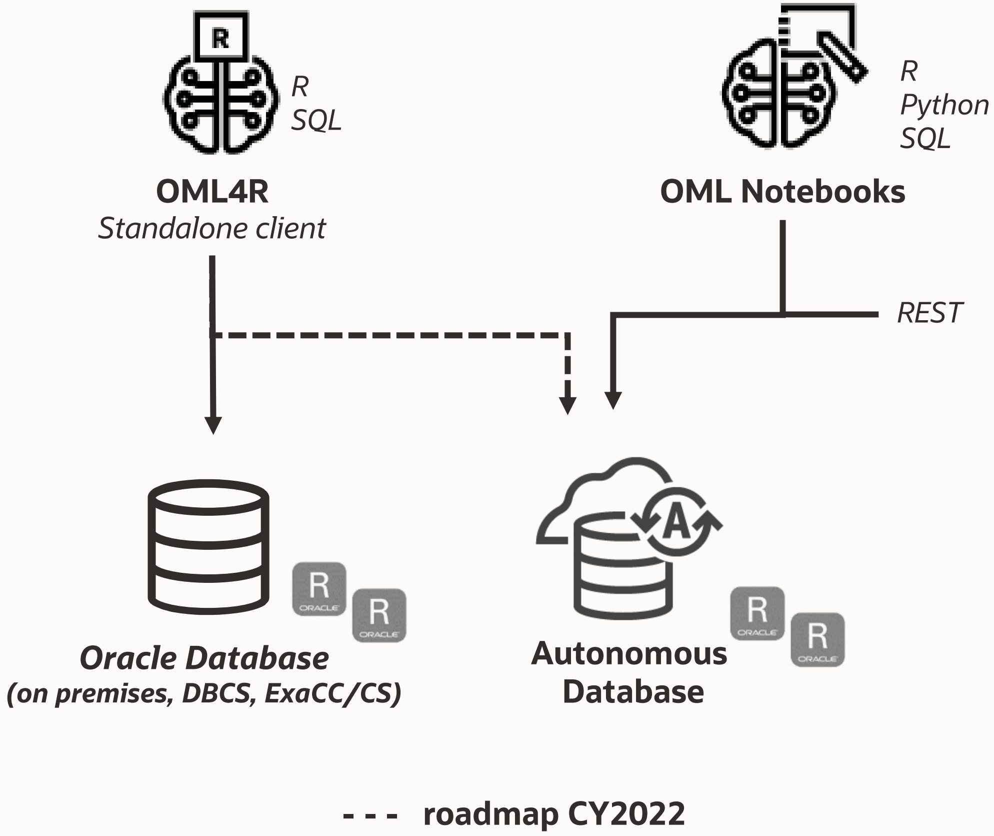
Oracle Database as HPC environment

In-database parallelized and distributed machine learning algorithms

Manage scripts and objects in Oracle Database

Integrate results into applications and dashboards via SQL and REST

No need to provision R engines for solution deployment



 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Oracle Machine Learning for R

Empower data scientists with open source environments

Transparency layer

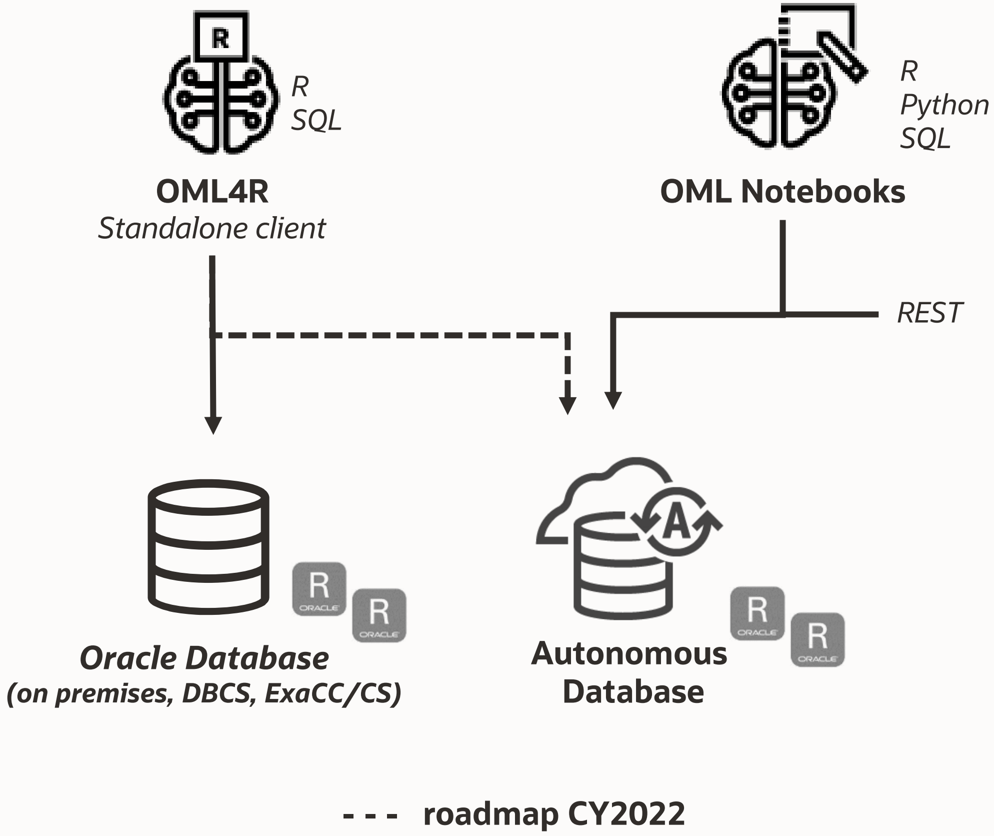
- Leverage proxy objects so data remains in database
- Overload native functions translating functionality to SQL
- Use familiar R syntax on database data

Parallel, distributed in-database algorithms

- Scalability and performance
- Exposes in-database algorithms available from OML4SQL

Embedded execution

- Manage and invoke user-defined R functions
- Data-parallel, task-parallel, and non-parallel execution
- Use open source packages to augment functionality



Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Poll #3: Feature Areas

What OML4R feature areas are you most interested in? (select all that apply)

- Transparency layer – data exploration and preparation
- In-database algorithms – scalable model building and data scoring
- Embedded R execution (ERE) – solution deployment from R, SQL, and REST
- Datastore – persist R objects in the database
- Script repository – store user-defined functions in the database for ERE

Oracle Machine Learning optimized for Oracle RAC

Examples

Familiar algorithms redesigned to enable distributed parallelism and scalability across cluster nodes

Scoring takes advantage of storage-tier optimizations with function push-down (Exadata platform)

Optimized memory utilization

- Data brought into memory incrementally as needed
- Models cached in the library cache and can be shared across queries
- Leverages disk-aware structures – relying on DB memory manager for efficient allocation in multi-user environment
- When building/scoring partitioned models, not all partitions need be loaded

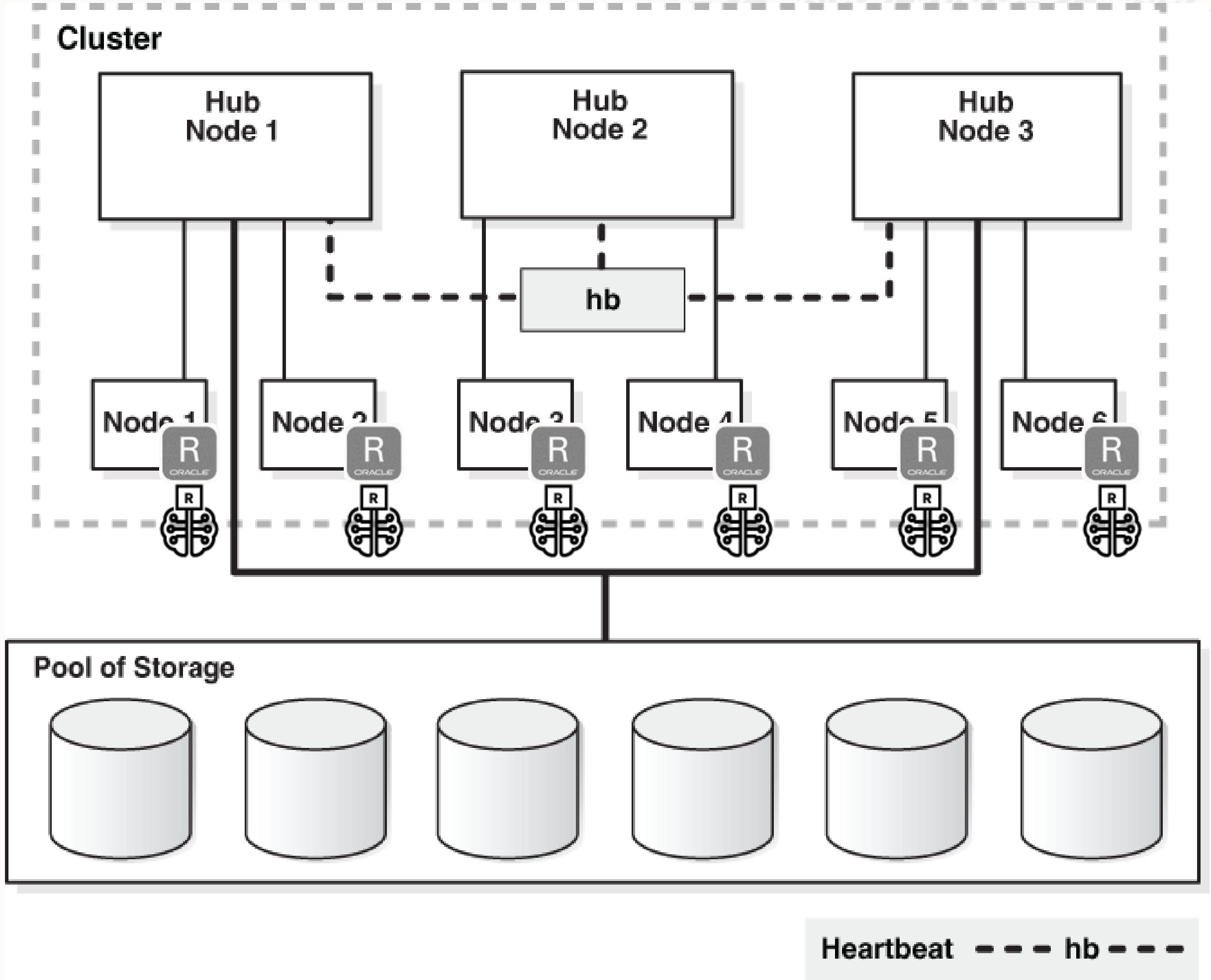
Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

OML4R on RAC

Supporting Oracle Database and Database Cloud Service

On each node…

- Install R
- Install OML4R server components
- Install desired third-party R packages for use with embedded R execution

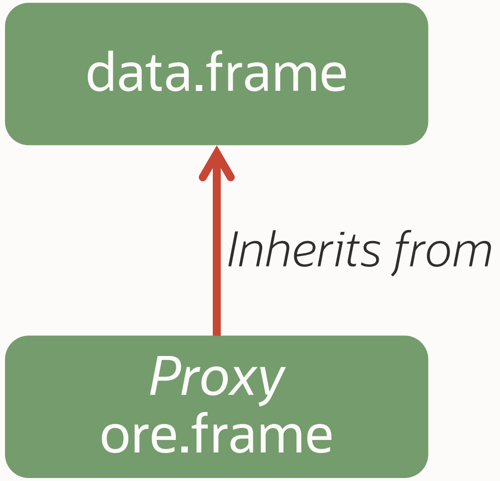


Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

R Proxy objects

Example using *iris* dataset

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa




```
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
> str(IRIS)
'data.frame':   150 obs. of  5 variables:
Formal class 'ore.frame' [package "OREbase"] with 12 slots
  ..@ .Data    : list()
  ..@ dataQry  : Named chr "( select /*+ no_merge(t) */  \"Sepal.Length\" VAL001,\"Sepal.Wid
th\" VAL002,\"Petal.Length\" VAL003,\"Petal.Width\" VAL004,\"Sp"| __truncated__
  .. ..- attr(*, "names")= chr "2539_1"
  ..@ dataObj  : chr "2539_1"
  ..@ desc     :'data.frame':   5 obs. of  2 variables:
  .. ..$ name  : chr  "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" ...
  .. ..$ Sclass: chr  "numeric" "numeric" "numeric" "numeric" ...
  ..@ sqlName  : chr
  ..@ sqlValue : chr  "\"Sepal.Length\"" "\"Sepal.Width\"" "\"Petal.Length\"" "\"Petal.Width
```

```
"( select /*+ no_merge(t) */  \"Sepal.Length\" VAL001,\"Sepal.Width\" VAL
002,\"Petal.Length\" VAL003,\"Petal.Width\" VAL004,\"Species\" VAL005 fro
m \"RQUSER\".\"IRIS\" t  )"
```


Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Transparency Layer

In-database performance – indexes, query optimization, parallelism, partitioning

Leverages proxy objects for database data

Uses familiar Python and R syntax to manipulate database data

Overloads Python and R functions translating functionality to SQL

```
# Create table from R data.frame data
ore.create(iris, table = 'IRIS')

# Create a temporary table from R data.frame
IRIS_TMP <- ore.push(iris)

# Get proxy object to DB table IRIS
ore.sync(table = 'IRIS')
ore.attach()
```

```
dim(IRIS)
head(IRIS)
summary(IRIS)
std(IRIS$age)
scale(IRIS$age)
```

 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Data Types

Mapping between R and Oracle Database

SQL – ROracle Read	R	SQL – ROracle Write
varchar2, char, clob, rowid	character	varchar2(4000)
number, float, binary_float, binary_double	numeric	if(ora.number==T) number else binary_double
integer	integer	integer
	logical	integer
date, timestamp	POSIXct	timestamp
	Date	timestamp
interval day to second	difftime	interval day to second
raw, blob, bfile	‘list’ of ‘raw’ vectors	raw(2000)
	factor (and other types)	character

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

OML4R packages provided with Autonomous Database

Cairo
DBI
IRdisplay
IRkernel
KernSmooth
MASS
Matrix
ORE
OREbase
OREcommon
OREdm
OREdplyr
OREds
OREeda
OREembed
OREgraphics
OREmodels
OREpredict
OREstats
OREXml
R6

ROracle
arules
assertthat
base
base64enc
boot
class
cli
cluster
codetools
compiler
crayon
datasets
digest
dplyr
ellipsis
evaluate
fansi
fastmap
Foreign
generics
glue

grDevices
graphics
grid
highr
htmltools
jsonlite
knitr
lattice
lazyeval
lifecycle
magrittr
markdown
methods
mgcv
mime
nlme
nnet
Parallel
pbdZMQ
Pillar
pkgconfig
png

purrr
randomForest
repr
rlang
rpart
spatial
splines
statmod
stats
stats4
stringi
stringr
survival
tcltk
tibble
tidyselect
tools
utf8
utils
uuid

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

OML4R Algorithms on ADB

Classification

- Decision Tree
- Logistic Regression
- Naïve Bayes
- Neural Network
- Support Vector Machine
- Random Forest
- XGBoost (21c)

Regression

- Generalized Linear Model
- Neural Network
- Support Vector Machine
- XGBoost (21c)

Clustering

- Hierarchical k-Means
- Orthogonal Partitioning
- Expectation Maximization

Attribute Importance

- Minimum Description Length
- Random Forest

Anomaly Detection

- 1 Class Support Vector Machine

Market Basket Analysis

- Apriori – Association Rules

Feature Extraction

- Nonnegative Matrix Factorization
- Principal Component Analysis
- Singular Value Decomposition
- Explicit Semantic Analysis

Time Series

- Single Exponential Smoothing
- Double Exponential Smoothing

Supports automatic data preparation, partitioned model ensembles, integrated text mining

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

R interface for Embedded R Execution

Build an ML model on the iris data set

```
buildModel <- function(dat,dsname) {
  mod <- lm(Petal.Length~Petal.Width, dat)
  ore.save(mod,dsname)
  TRUE
}

ore.scriptCreate('buildModel',buildModel)

ore.sync(table='IRIS')  # get ore.frame proxy object

ore.tableApply(IRIS, FUN.NAME='buildModel',
               dsname= 'LM-model-iris-species',
               ore.connect=TRUE)
```

Save resulting model in database datastore

Store the UDF in the script repository

Provide proxy object IRIS
Invoke UDF by name
Specify the datastore name

Copyright © 2022, Oracle and/or its affiliates

SQL interface for Embedded R Execution – Autonomous Database

Data from IRIS table

Specify function arguments

SQL

```
select *
from table(rqTableEval(inp_nam  => 'IRIS',
                       par_lst  => '{"dsname":"LM-model-iris-species",
                                     "ore_service_level":"MEDIUM"}',
                       out_fmt  => 'XML',
                       scr_name => 'buildModel')
```

Return result as XML string

Specify name of UDF in repository

 Copyright © 2022, Oracle and/or its affiliates

Store UDF in script repository using the R and SQL APIs

```
scoreData <- function(dat, dsname) {
      ore.load(dsname)
      dat$Prediction <- predict(mod, newdata = dat)
      dat[,c("Petal.Length","Prediction")]
    }

ore.scriptCreate(name = "scoreData", FUN = scoreData, overwrite = TRUE)
```

SQL

```
BEGIN
 sys.rqScriptCreate('scoreData',
   'function(dat, dsname) {
      ore.load(dsname)
      dat$Prediction <- predict(mod, newdata = dat)
      dat[,c("Petal.Length","Prediction")]
    }',FALSE, TRUE); -- not sharing function and enable overwrite
END;
```


Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

R interface for Embedded R Execution

Example of parallel partitioned data flow

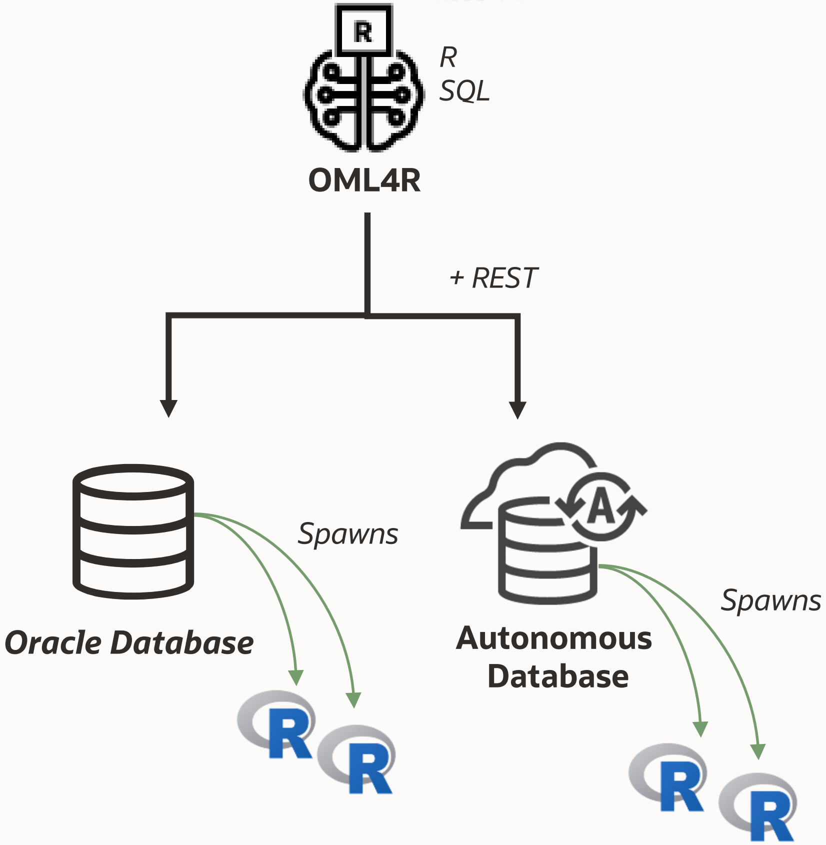
```
ore.sync(table='IRIS') # get ore.frame proxy object

PRED = ore.rowApply(IRIS,
         FUN.NAME = 'scoreData',
         rows = 10,
         parallel = 4,
         FUN.VALUE = data.frame(Petal.Length=numeric(),
                                Prediction=numeric())

class(PRED)    # returns an ore.frame proxy object

ore.create(PRED,table = 'BATCH_SCORES') # persist table

with(BATCH_SCORES, table(Species, Prediction))
```



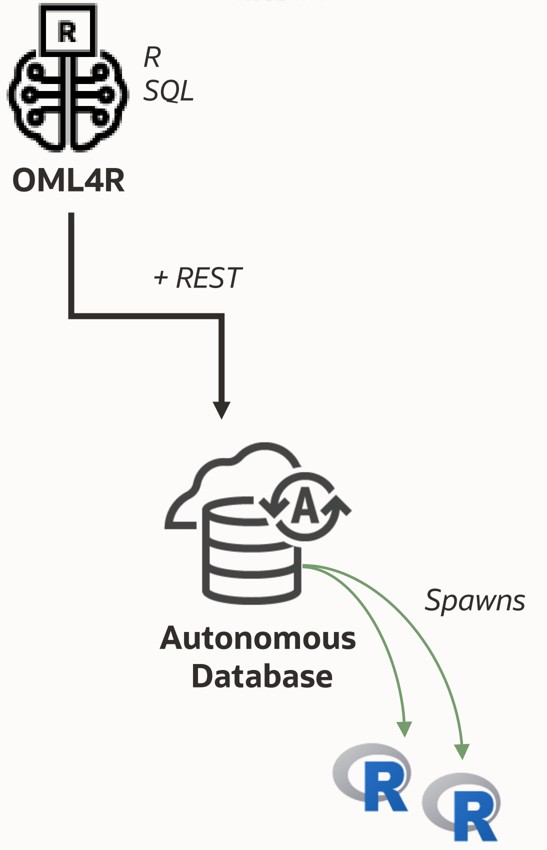
 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

SQL interface for Embedded R Execution – Autonomous Database

Example of parallel partitioned data flow

SQL

```
-- create table BATCH_SCORES as
select * from table(rqRowEval(
  inp_nam  => 'IRIS',
  par_lst  => '{"dsname":"LM-model-iris-species",
                "ore_parallel_flag":true,
                "ore_service_level":"MEDIUM"}',
  out_fmt  => '{"Petal.Length":"number",
                "Prediction":"number"}',
  row_num  => 10,
  scr_name => 'scoreData'));
```



 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Compute resources

OCPUs, memory, auto-scale

When the ADB instance auto-scale is disabled, total VM usage is limited to 1 OCPU

When auto-scale is enabled, total VM usage is limited to 2 x base OCPUs,
up to a max of 5 VMs per tenant (8 OCPU per VM)

Example:
ADB instance provisioned with 8 OCPUs will allow 2 VMs allocated to run containers for a total of 16 OCPU

Service level	OCPU limit with auto-scale	OCPU limit without auto-scale	Memory (GB)	Storage (GB)	Max concurrent runs with auto-scale	Max concurrent runs without auto-scale
High	8	1	8	2	Up to 3	Up to 3
Medium	4	1	4	2	Up to 20	Up to 3
Low	1	1	2	2	Up to 100	Up to 5
TP	1	1	2	2	Up to 100	Up to 5
TP Urgent	1	1	2	2	Up to 100	Up to 5

 Copyright © 2022, Oracle and/or its affiliates

ORACLE

Oracle Machine Learning for R

Demonstration

Sherry LaMonica
Consulting MTS, Oracle Machine Learning Product Management
August 2022

Copyright © 2022, Oracle and/or its affiliates

Why Oracle for Machine Learning with R?

Oracle integrates ML across the Oracle stack and the enterprise

Empower data scientists and R users with powerful in-database ML from an R API

Eliminate costly data movement and latency to client R engines

Scale R for data exploration, data preparation, and ML algorithms

Use in-database algorithms for regression, classification, time series, association rules, attribute importance, clustering, feature extraction, and anomaly detection

Benefit from automatic data preparation, partition models, integrated text mining

Deploy ML models and R UDFs easily with data-parallel and task-parallel support

Leverage existing database backup, recovery, and security

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Poll #4: Expectations and Satisfaction

How many stars would you give this session?

- *****
- ****
- ***
- **
- *

R consortium Mission and Vision

Promote the R language and lead initiatives in support of the R community

The R Consortium works with and provides support to the R Foundation and key organizations developing, maintaining, distributing, and using R software.

Oracle is a founding member of and contributor to the R Consortium.

https://r-consortium.org

For more information…

OML Webpage
https://oracle.com/machine-learning

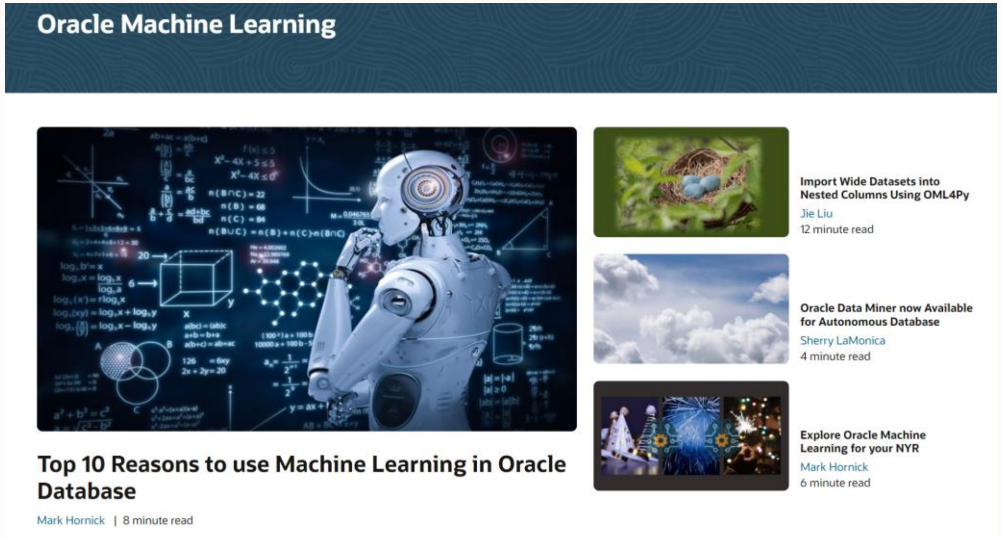
Machine Learning Blog
https://bit.ly/omlblogs

GitHub Repository
https://bit.ly/omlgithub

OML Office Hours
https://bit.ly/omlofficehours

Oracle Live Labs
For Oracle Database: Introduction to Oracle Machine Learning for R
For Oracle Autonomous Database: coming soon

OML4R Documentation
https://docs.oracle.com/en/database/oracle/machine-learning/oml4r



 Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Q & A

Thank you

Mark Hornick
mark.hornick@oracle.com

Sherry LaMonica
sherry.lamonica@oracle.com

Group: Oracle Machine Learning

 Copyright © 2022, Oracle and/or its affiliates