ORACLE

Bringing R to the Enterprise

Using R for Enterprise-level Performance, Scalability, Ease of Production Deployment, and Security

July 10, 2020 | Version 2.0

PURPOSE STATEMENT

This document provides an overview of features and enhancements included in release Oracle Machine Learning for R 1.5.1 and Oracle Machine Learning for Spark 2.8.2. It is intended solely to help you assess the business benefits of Oracle Machine Learning for R 1.5.1 and Oracle Machine Learning for Spark 2.8.2 and to plan your data science, machine learning, and information technology projects.

INTENDED AUDIENCE

The material in this whitepaper will benefit chief data scientists, data scientists, R users, and information technology professionals interested in leveraging the power of Oracle Database in combination with the R language and ecosystem.

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

TABLE OF CONTENTS

INTRODUCTION

R is an integrated suite of software facilities for statistical analysis, data exploration and manipulation, machine learning, and visualization formulated in 1994 as an alternative to proprietary statistical environments. This open source scripting language has become an important part of the analytical arsenal for data scientists, business and data analysts, and statisticians. With millions of R users worldwide leveraging thousands of open source R packages, the R ecosystem enhances user productivity in a wide range of domains, including pharmaceuticals, bioinformatics, spatial statistics, financial market analysis, and linear/non-linear modeling.

While data scientists often run R programs on their personal computers or workstations, increasingly they need to do advanced computations on large volumes of data quickly. To enable using R on data at scale, Oracle created a wide range of capabilities for conducting machine learning, statistical, and graphical analyses on data stored in Oracle Database and Big Data environments. These enterprise-level capabilities enable projects that require high levels of security, scalability, and performance, as well as the ability to deploy R scripts into production quickly and easily, either on-premise or in Oracle Cloud.

Oracle Machine Learning combines the advantages of R with the power and scalability of Oracle Database and Data Lake environments. Oracle Machine Learning for R (OML4R) is included with Oracle Database, both on premise and on Oracle Cloud. Oracle Machine Learning for Spark is a component of Oracle Big Data Connectors software suite for on-premise Big Data solutions, and is included with Oracle Big Data Service. R programs and packages, used in conjunction with these products, can process larger volumes of data in a secure environment. Customers can build machine learning models and execute them against local data stores as well as run R commands and scripts against data stored in Oracle Database. Using OML4R, R user-defined functions can be invoked from SQL and return structured results and images that can be immediately deployed in applications and dashboards.

These capabilities are especially important for many of today's big data projects. Data Scientists can obtain controlled access to data in Oracle Database, thereby accelerating productivity while enforcing IT security policies. Oracle's integrated approach simplifies data analysis, minimizes or eliminates data movement, and shortens the time to transform raw data into actionable information. Further, Oracle Big Data SQL, an optional component for Big Data environments on premise, extends the reach of OML4R to data lake sources by allowing R users to manipulate big data sources mapped as database tables directly from R. Using Oracle Cloud SQL, which is part of Oracle Big Data Service, users can do that on Oracle Cloud.

Oracle Machine Learning has two components that provide support for users of both SQL and R. Through Oracle Machine Learning for R (OML4R), R users transparently manipulate data in Oracle Database using standard R syntax, without data movement, leveraging Oracle Database as a high-performance compute engine. By translating R function invocations to SQL, users leverage in-database statistical techniques for enhanced scalability and performance. OML4R provides a well-integrated R API to the powerful in-database machine learning algorithms. OML4R also provides the ability to execute user-defined R functions on R engines spawned and controlled by Oracle Database. These user-defined R functions can be invoked from either R or SQL and can leverage Comprehensive R Archive Network (CRAN) and other R packages. Oracle Machine Learning users have access to a range of Oracle-provided and third-party R GUI and IDE options targeting the user spectrum from business analysts to data scientists.

Oracle Machine Learning for Spark (OML4Spark), formerly Oracle R Advanced Analytics for Hadoop (ORAAH) and one of the Oracle Big Data Connectors, enables R users to manipulate data transparently in Apache Hive and Apache Impala using standard R syntax, and manipulate data in Apache Spark via SparkSQL and Spark Dataframe functions. OML4Spark also provides a rich set of Spark-based parallelized machine learning algorithms, including many Spark MLlib algorithms exposed through a unified R interface. Further, users can execute custom MapReduce jobs from R. Users write mapper and reducer functions in R and these functions can leverage CRAN packages.

OML4R and OML4Spark can be used in conjunction with Oracle's redistribution of open-source R, called Oracle R Distribution, as well as with the open-source R distribution. Oracle R Distribution can automatically uptake high-performance libraries such as Intel’s MKL for enhanced linear algebra and matrix processing. Oracle R Distribution is supported by Oracle.

In this paper, we highlight how Oracle enhances open-source R by enabling developers to:

- Transparently analyze and manipulate data in Oracle Database or Spark/Hadoop
- Execute R scripts via Oracle Database with data and task parallelism
- Use in-database SQL-based algorithms seamlessly through R
- Score R models in the database
- Easily execute R scripts from SQL
- Integrate R into the IT software stack

By integrating R with both Oracle Database and Big Data, your organization can realize the best of both worlds: obtain a familiar yet powerful statistical environment along with vastly improved scalability, performance, and security.

THE BIG PICTURE

RUN R CODE LEVERAGING ORACLE DATABASE

Data Scientists and other R users can execute R scripts through their favorite R IDE for data exploration and preparation, and to access a wide range of machine learning capabilities on database data without the need to know SQL.

WRITE R, EXECUTE IN-DATABASE

Oracle developed a set of R packages that allows R computations to be executed within Oracle Database. This transparency layer makes Oracle tables and views accessible to the R environment as if they were native R objects through *ore.frames*, which are a subclass of R’s *data.frame*. This allows users to execute a wide range of R functionality.

R users can also leverage the OREdplyr package, which provides overloaded functionality from the popular open source R dplyr package. These features allow users to focus on data analytics opportunities rather than data access, scalability, and performance challenges.

The transparency layer allows R developers to use familiar environments, languages and tools. Under the covers, overloaded R functions execute within Oracle Database – taking advantage of database parallelism, query optimization, column indexing and partitioning – leveraging the rich in-database library of statistical functionality. R users can execute these complex computations within the database using their standard R development skills and tools.

IN-DATABASE MACHINE LEARNING ALGORITHMS

Users can build, evaluate, share, and deploy predictive analytics methodologies, while also utilizing high-performance parallel and distributed machine learning algorithms from Oracle. These in-database algorithms also accept text columns from tables and views for integrated text mining automated term and theme extraction. The extracted data is then combined with other predictors in building models and scoring data.

Further enhancing data scientist productivity, users can automatically create ensembles of models – called partitioned models – where each component model is built on a user-specified partition of the data. Scoring is enabled and simplified using a single integrated model.

Data Scientists, analysts, and application developers can easily scale their machine learning projects as data volumes increase by bringing the algorithms to where the data reside. For example, they can use native parallel distributed in-database algorithms like decision trees, support vector machine, k-means, neural networks, and random forest for scalable machine learning. They can analyze data and make predictions even faster when they run on Oracle Exadata – a powerful

Oracle engineered system - since this processing can take place at the storage tier. This allows organizations to gain further benefits from the extreme performance provided by Oracle Engineered Systems. The benefits of the Oracle approach are clear:

- Work solely from within R for data preparation, analysis, and visualization
- Use the database as a high-performance compute engine with query optimization, column indexing, and parallelism, and optional functionality for in-memory execution and partitioning
- No need to manage flat file data, or wrestle with the associated complexity of storage, backup, recovery, and security
- Minimize R memory constraints so you can handle big data requirements
- Execute R scripts from SQL for ease of deployment and integration with enterprise applications and dashboards

ENJOY USING CRAN PACKAGES

Oracle's *embedded R execution* capability allows Data Scientists to leverage thousands of specialized algorithms from the Comprehensive R Archive Network (CRAN) repository. They can write their own algorithms or download existing ones, and then install these packages in database server-side R engines. This architecture makes it easy to send and receive data securely to and from the database and feed it directly to their chosen algorithms.

Take advantage of parallel user-defined function execution with corresponding data feeds with built-in data-parallel and task-parallel infrastruture. For example, you might divide a customer table by zip code and run multiple R engines in parallel to process groups of customers from each zip code concurrently, all without leaving the R environment. R scripts that use a wide variety of statistical techniques—some accessible through the transparency layer and some through CRAN packages—can be built and stored in Oracle's in-database R script repository, then executed from R or SQL.

- Create your own packages in R or leverage CRAN open-source packages
- Execute user-defined functions at the database server machine in R engines spawned under control of Oracle Database
- Enable "lights-out" execution of R scripts via a SQL interface using Oracle Database scheduling
- Speed up large jobs with data-parallel and task-parallel user-defined R function execution
- Integrate results with applications,dashboards, and reports

DEPLOY R ANALYTICS IN PRODUCTION

Oracle Machine Learning enables R developers to use Oracle Database to execute R scripts within SQL queries. This makes it easy to operationalize R scripts in production for any analytic application. Oracle Database SQL queries can contain a call to a user-defined R function that is registered in the database R script repository. Using the script name, users can initiate a query to call that script and receive structured and image results in as a table, or combined as XML.

One telecommunications provider used OML4R to power complex survey research. Analysts at this firm maintain user-defined R functions in Oracle Database and then filter data and display results through a parameterized analytics dashboard. Both the database and the dashboard infrastructure are standard components of the architecture, further enhanced by their SQL-based connection to R scripts. These capabilities make R a powerful language that can execute advanced statistical models directly on database data.

RUN R WITH HADOOP AND SPARK

Oracle Machine Learning for Spark (OML4Spark), formerly Oracle R Advanced Analytics for Hadoop (ORAAH), is a component with an R package front-end that provides best-in-class Spark-based machine learning algorithms for data in big data clusters, as well as transparent access to Hadoop and data stored in HDFS, Apache Hive, Apache Impala, and Spark DataFrames. OML4Spark enables users to build R models and score efficiently against large volumes of data, as well as to leverage Spark in-memory processing without leaving the R environment. They can use R to analyze data stored in Data Lakes with Oracle-supplied machine learning algorithms plus a select number of Spark MLlib algorithms, as well as using CRAN R packages.

When it comes to machine learning, OML4Spark provides several parallelized distributed algorithms whose execution benefits from a big data Cluster with Spark. OML4Spark custom algorithms – including linear model, generalized linear model, and multi-layer perceptron neural network – scale better on Spark because they do not require that all data fit in memory at once. They also run faster than similar open-source Spark MLlib functions. OML4Spark provides enhanced interfaces to MLlib that take advantage of the full R-formula specification and surpass those provided by SparkR.

The algorithms that run in Spark support both model build and apply (prediction scoring) with input datasets in the form of HDFS, Apache Hive, Apache Impala, Spark DataFrames, and JDBC data sources. The models themselves can be stored in binary format on HDFS and the local file system for execution on a different cluster.

OML4Spark enables R commands to run on data accessible from Apache Hive and Apache Impala tables by leveraging the same transparency layer functionality supported by OML4R. This transparency layer allows R developers to use the familiar R environment and commands, while under the covers functions are automatically converted to HQL (Hive Query Language) or Cloudera Imapala SQL and are executed on the Hadoop Cluster in parallel.

Additionally, users can manipulate data in Spark Dataframes directly from R, making use of all functions for basic data transformations like *join*, *append*, *aggregate* statistical functions as well as create new columns and when desired run SparkSQL queries directly in-memory on the Spark DataFrames, as well as use SparkSQL against tables registered in HIVE.

R programs that take advantage of MapReduce can be deployed on a Hadoop cluster and benefit from the data-parallel nature of a Hadoop cluster for performance. Users do not need to know about Hadoop internals, MapReduce command line interfaces, or the IT infrastructure to create, run, and store these R scripts.

BIG DATA IOT USE CASE WITH ORACLE DATABASE

The Internet of Things (IoT) presents new opportunities for applying machine learning. Sensors are everywhere collecting data – on airplanes, trains, and cars, in semiconductor production machinery and the Large Hadron Collider, and even in our homes. One such sensor is the home energy smart meter, which can report household energy consumption periodically, perhaps as often as every 15 minutes. This data enables energy companies to model not only each customer's energy consumption patterns, but also to forecast individual customer usage. Across all customers, energy companies can compute and forecast aggregate demand, which enables more efficient deployment of personnel, redirection or purchase of energy, and so on, often a few days or weeks out.

Building one predictive model per customer, when an energy company may have millions of customers, poses some interesting challenges. Consider an energy company with 1 million customers. Over the course of a single year, these smart meters could collect over 35 billion readings. Each customer, however, generates only about 35,000 readings. On most hardware, R can easily build a forecast model on 35,000 readings. Note that if each model requires even only 10 seconds to build, doing this serially will require roughly 116 days to build all models. Since the results are needed a few days or weeks out, a delay of months makes this project a non-starter. If powerful hardware, such as Oracle Exadata on premise or on Oracle Cloud, can be leveraged to build these models in parallel, say with degree of parallelism of 128, all models can be computed in less than one day.

While users can leverage parallelism enabled by various R packages, several factors need to be considered. For example, what happens if certain models fail? Will the models be stored as 1 million separate flat files – one per customer? For flat files, how will backup, recovery, and security be handled? How can these models be used for forecasting customer usage and where will the forecasts be stored? How can these R models be incorporated into a production environment where applications and dashboards normally work with SQL?

Using the embedded R execution capability of OML4R, data scientists can focus on the task of building a model for a single customer in a user-defined R function. This function is stored in the R Script Repository in Oracle Database. OML4R enables invoking this function using one of the embedded R execution functions, such as *ore.groupApply*. Embedded R execution in Oracle Database manages spawning multiple R engines, loading one partition of data from the specified database table to the function produced by the data scientist, and then store the resulting model immediately in the R Datastore, again in Oracle Database. This greatly simplifies the process of building and storing models. Moreover, standard database backup and recovery mechanisms—already in place—can be used to avoid having to devise separate specialized practices. Forecasting energy consumption using these models is handled in an analogous way.

To put these user-defined R functions into production, users can invoke the same R functions produced by the data scientist from SQL, both for model building and forecasting. The forecasts can be immediately available as a database table for use by applications and dashboards, or used in other SQL queries. In addition, SQL statements that invoke the R functions can be scheduled for periodic execution using the DBMS_SCHEDULER package of Oracle Database.

Leveraging the built-in functionality of Oracle Machine Learning, data scientists, data anlysts, application developers, and administrators do not have to reinvent complex code and testing strategies, as is normally done for each new project. Instead, they benefit from Oracle's integration of R with Oracle Database to easily design and implement R-based solutions for use with applications and dashboards, and scale to the enterprise.

BIG DATA USE CASES WITH ORACLE DATABASE AND HADOOP

Oracle's big data technologies are designed to easily move data between Big Data environments, R, and Oracle Database. Data scientists can access data stored in Oracle or a Hadoop cluster and can code MapReduce processes, Apache Hive or Spark queries, and run machine learning algorithms in R without having to resort to Java or Scala. As described below, this flexible architecture enables organizations to analyze large tables and large data sets easily. In addition to SQL, R is now a good option for enterprise analytics to solve the pressing big data challenges of today.

Use Case 1: Analyzing Credit Risk

Banks continually offer new services to their customers, but the terms of these offers vary based on each customer's credit status. Do they pay the minimum amount due on credit balances, or more? Are their payments ever late? How much of their credit lines do they use and how many other credit lines do they have? What is the overall debt-to-income ratio?

All of these variables influence policies about how much credit to award to each customer, and what type of terms to offer them. A bureau like Equifax or Transunion examines an individual's overall credit history, but banks can examine a much more detailed set of records about their customers—down to the level of every discrete transaction. They need big data analytics to get down to this level of precision with this volume of data.

For example, one Oracle customer in Brazil is running multiple neural network algorithms against hundreds of millions of records to examine thousands of attributes about each of its customers. Previously the bank had trouble crunching this massive volume of data to generate meaningful statistics. They solved this problem by running a specialized algorithm using OML4Spark to analyze the data in parallel on the same cluster that is running the production Hadoop file system, Apache Hive, and other tools. OML4Spark enables data scientists to execute R analyses, statistics and models on tables stored in the bank's large Hadoop file systems. They can now run complex statistical algorithms against these files systems and Apache Hive tables.

The algorithms use standard R approaches, such as building models using the R formula object. Behind the scenes, OML4Spark provides the interface for executing Spark-based implementations or MapReduce jobs in parallel on multiple processors throughout the bank's cluster. Data scientists can create these MapReduce processes and Spark algorithms in R and store them in Hadoop as well as easily surface these models for review, plotting and analysis—and then push the results to Oracle Database—without having to utilize Java.

Use Case 2: Detecting Fraud

Another popular use case involves detecting fraud by analyzing financial transactions. Banks, retailers, credit card companies, telecommunications firms and many other large organizations wrestle with this issue. When scoring data to detect possible fraud, you typically study transactions as they occur within customer accounts (scoring refers to predicting outcomes using a machine learning model.)

Once you understand normal customer behavior, you can then recognize unusual patterns and suspicious transactions. For example, if you normally shop in Los Angeles and there is a sudden series of transactions in Rome this would indicate a high likelihood of fraud. Or would it? If you are somebody who travels a lot, is a surge of activity in Rome an anomaly or a regular pattern? By capturing all previous transactions and studying these patterns, you can develop a model that reflects normal behavior.

While R has algorithms and the environment for creating a predictive model that can analyze these transactions, the algorithms as found in CRAN packages are typically not multi-threaded. Hence, the algorithm is limited by the memory and single CPU processing power of the machine on which it runs. R typically does not leverage the CPU capacity of even a multi-processor laptop without special packages and programming.

Oracle Machine Learning can handle the massive computational requirements associated with analyzing customer-purchasing patterns using the R language to define scripts that are stored in the R Script Repository and run in the database.

Organizations can leverage Oracle Exadata and Oracle Big Data Appliance engineered systems to scale the effort on premise, or rely on the scalability of the Oracle Cloud Infrastructure, and integrate Oracle Machine Learning results with applications like Oracle Analytics Server and Oracle Data Visualization Desktop on premise, or Oracle Analytics Cloud to display the results. R developers can use the results of a fraud model built in Hadoop using R and deploy that model in Oracle Database where it can rapidly predict behavior at the transactional level and be part of an enterprise application for real-time predictions.

Real-time analysis is important in fraud prevention. It is one thing to identify a fraudulent transaction that happened eight hours ago (the length of time it might take to stage data into a laptop and run a detailed analysis of yesterday's activities). It is clearly much more valuable to score that transaction against a model in real-time, with the potential to block the transaction or flag it for further scrutiny.

Use Case 3: Preventing Customer Churn

Customer churn is a major problem for many businesses, especially in highly competitive markets such as telecommunications. For example, as a mobile phone user, if you have problems with reception or experience too many dropped calls you might think about looking for another service provider. Your existing service provider is constantly analyzing your behavior to predict how likely you are to defect. They have a statistical model that says, "90% of our customers with similar issues and behavior have left us for another service provider." They can apply that model to your data to create a score that reveals your likelihood of defecting. Your score relates you to millions of other customers with similar behavior.

Using Oracle Machine Learning, you can run these R models while a customer is browsing a webpage or using a mobile app and then make on-the-spot recommendations based on current actions and real time analytics against an operational data store or data warehouse.

Scoring can also be done offline – in batches. For example, you might want to predict which of your 100 million customers will respond to each of a dozen offers so you can identify which customers should be targeted with a special offer or ad campaign. With enough processing power and the right predictive model, data scientists can provide insight not only into what the churn rate is but also into the reasons behind the churn. One telecommunications company used OML4Spark to make richer, more informed decisions by examining payment records, calling plans, and service histories to detect similarities and trends within its customer databases. This permitted them to run batch jobs in parallel on a large Big Data cluster.

CONCLUSION

Most organizations depend on databases to store information securely with rigorous, enterprise-level controls. Oracle has made R highly compatible with large-scale machine learning, analytics tools, and big data initiatives. Developers can use the familiar R environment in conjunction with Oracle Database, Big Data clusters, and analytics tools as they leverage Oracle's massive scalability and performance to solve enterprise big data problems. Data scientists and analysts who are accustomed to working with file extracts can adopt a database-centric architecture, pushing data from their desktop R implementations to the database and processing that data in Oracle Database.

R-to-SQL transparency improves user efficiency by allowing data scientists to use R directly against data in an Oracle database, Apache Hive or Apache Impala. R users can leverage in-database SQL analytic and machine learning functions and open source R packages in combination with Oracle Database for data-parallel and task-parallel execution.

With Oracle Machine Learning, you can remain in the R environment. You can leverage CRAN packages and other R assets that you have created, and invoke user-defined R funtions from SQL--enabling deployment of R-based analytics into production applications and dashboards. For big data problems, you can leverage the scalability of OML4Spark with multiple Big Data clusters.

Customers that license Oracle Database, Oracle Big Data Connectors, or Oracle Linux, as well as customers running Oracle Database Cloud Service or Oracle Big Data Service, receive enterprise class support for Oracle R Distribution.

In summary, by extending R to work with Oracle Database and Big Data you can bring your analysis to the data, rather than the other way around. By pushing R functionality to Oracle Database via Oracle Machine Learning and invoking Spark jobs from R on Big Data cluster nodes via OML4Spark, data scientists can minimize data movement and decrease latency from raw data to actionable information. Integrating Oracle Database and Big Data environments provides a powerful, cost-effective solution for big data analytics.

FOR FURTHER READING

ORACLE MACHINE LEARNING

https://oracle.com/machine-learning

ORACLE MACHINE LEARNING FOR R

https://oracle.com/goto/R

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Bringing R to the Enterprise
July, 2020