



Using Machine Learning in OBIEE for Actionable BI

By

Lakshman Bulusu

Mitchell Martin Inc./ Bank of America

Using Machine Learning in OBIEE for Actionable BI

- Using Machine Learning (ML) via Oracle R Technologies in OBIEE
- Implementing an R-based predictive analytics algorithm of ML in Oracle DB 12c
- Output of prediction visually integrated in OBIEE dashboards for actionable insight
- Key takeaways:
 - An outline of Oracle R technologies
 - Building and scoring a model using **Random Forest ML classification** algorithm for in-database predictive analytics in DB 12c
 - End-to-End implementation steps of the above and its integration with OBIEE dashboards backed by a business use case

Outline of Oracle R Technologies

- Open Source R – ready-to-use R functions, related code and data in the form of packages and views called CRAN Task views for ML and Statistical Learning (CRAN stands for Comprehensive R Archive Network , available at <http://cran.r-project.org> or <http://www.R-project.org/>)
- Oracle's R technologies
 - Oracle R Distribution
 - ROracle
 - Oracle R Enterprise (ORE)
 - Oracle R Advanced Analytics for Hadoop

Oracle R Technologies contd....

- Oracle R Distribution – v.3.2.0
 - R software redistribution of open source R (freely available) – supports dynamic loading of high-performant math libraries for multi-threaded execution, e.g., Intel Math Kernel library etc.
 - Enhancements to open source R for scalability across users and database for embedded R execution
- ROracle – An open source R package managed by Oracle that comes with a DBI compliant Oracle driver for R using OCI lib re-engineered and optimized for DB connectivity between R and Oracle.
 - SQL statements execution from R interface
 - Transactional support for DML operations
 - Scalable across all data-types including Oracle NUMBER, VARCHAR2, TIMESTAMP, and RAW datatypes as also with large result sets.
 - Connect to Oracle DB from Oracle R distribution –Oracle Instant Client or Oracle standard DB client must be installed. SQL*Plus SQL interactive interface can be used with the former.
 - ROracle v.3-1.11 availability as of this writing

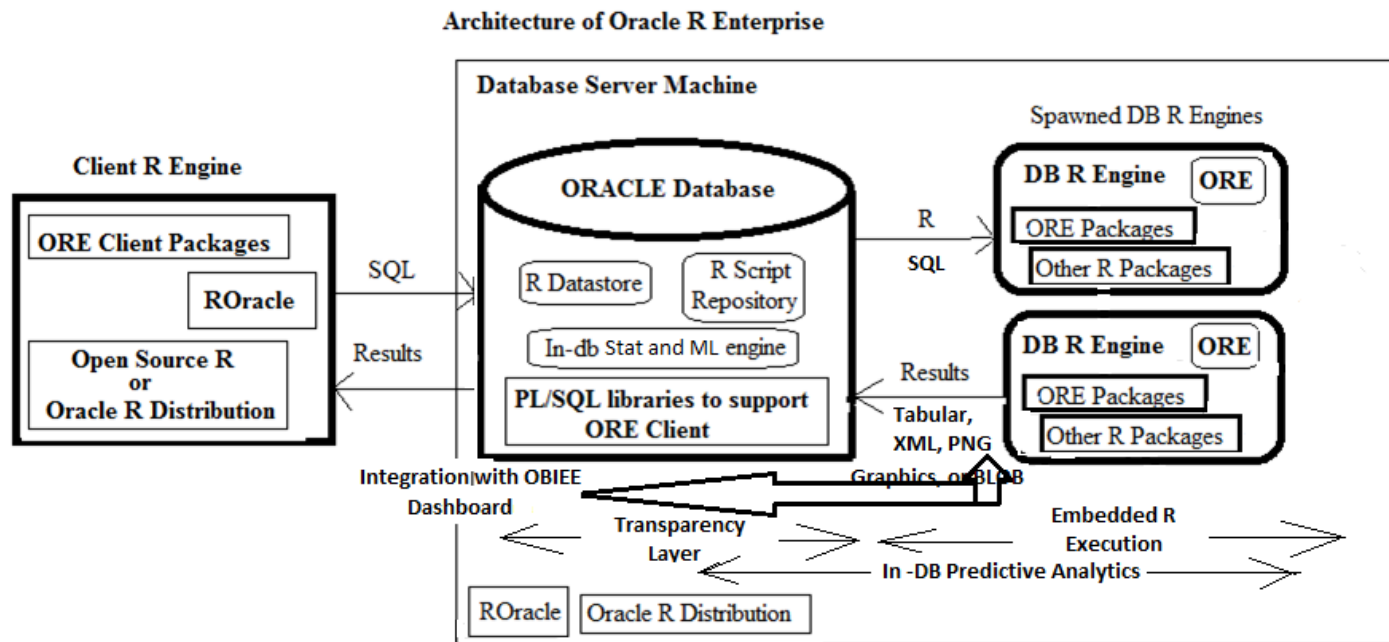
Oracle R Technologies contd....

- Oracle R Advanced Analytics for Hadoop (ORAAH)
 - A component of Oracle Big Data Connectors software suite
 - Available as an Hadoop cluster integrating Hadoop and HIVE with Oracle DB (Advanced Analytics option)
 - Data sourced in HDFS in can be harnessed using Hive-based SQL for as also via HDFS data mappings as direct input to ML routines – executed as MapReduce jobs or using Apache Spark. HFS enables data preparation, joins, and view creation.
 - Source data in HDFA CSV, Hadoop table or cached into Apache Spark as RDD (Resilient Distributed Dataset)
 - Details of primary ML algorithms at <http://www.oracle.com/technetwork/database/database-technologies/bdc/r-advanalytics-for-hadoop/overview/index.html> and examples of use cases based on these algorithms at https://blogs.oracle.com/R/entry/oracle_r_advanced_analytics_for

Oracle R technologies – Oracle R Enterprise(ORE)

- Oracle R Enterprise (ORE) – the main component of Oracle R technologies available with the Oracle Advanced Analytics option
- Architecture
 - Database Server Machine with **Oracle** DB installed – contains libraries and PL/SQL programs for ORE client
 - R Engine installed on Oracle DB for **embedded R execution - in-database execution of statistics and machine learning functions and models**. Each DB R Engine has ORE Server and ORE Client packages.
 - Multiple R Engines spawned for data parallelism.
 - Native Oracle DB features for SQL and PL/SQL
 - Oracle R Distribution
 - ROracle for database connectivity
 - Client R Engine with client ORE packages, open source R (or Oracle R Distribution) and ROracle
 - In-DB R Script Repository that stores R scripts – callable by name directly from SQL
 - In-DB R datastore that stores R objects

Oracle R Enterprise (ORE) contd....



ORE contd....

- Advantages –
 - In-database execution on stored data directly
 - Minimal latency, high-performance, multi-threaded and parallel capabilities for data and models,
 - Scalability
 - Minimal memory usage
- ORE extends open source R in the following manner:
 - ORE Transparency Layer
 - Embedded R Execution (via the Embedded R Engine) - both R Interface and SQL Interface
 - Predictive Analytics
 - As of this writing, the latest version of Oracle R Enterprise is 1.5.0.
- Further details and additional enhancements to ORE can be found at <http://www.oracle.com/technetwork/database/database-technologies/r/r-enterprise/overview/index.html>.

Oracle R Enterprise contd....

- ORE Transparency Layer –
 - In-database computation and analysis of data through R is done transparently.
 - Basic **data in R operations can be coded as R data.frames** and the corresponding R functions are exposed as overloaded function definitions and their execution pushed to Oracle DB where the statistical and machine learning operations are executed on data in Oracle tables.
 - **ORE provides ore.frame objects (a sub-class of data.frame) that get substituted for database tables, thereby SQL being generated, for in-db execution on data stored in Oracle.**
 - This kind of *Transparency Layer* provides reduced latency and optimal performance in terms of operational efficiency and at big-dataset scale.
- Embedded R Execution –
 - Embedded R engine(s) spawned by Oracle DB allow data- and task- parallelism via its *Embedded R Execution* support. Embedded R Execution is exposed through both R and SQL APIs. SQL APIs allow seamless integration with other applications such as OBIEE RPDs and dashboards by passing results from R for business intelligence and advanced analytics. Table 2 lists these APIs (both R Interface and SQL Interface) for ORE v1.5.0
 - Advantages include more memory availability and database-enabled data parallelism
 - **Light-out processing - Embedded R execution enables execution of R scripts (including those that are open source CRAN packages based) embedded in SQL and PL/SQL routines.**
 - **R scripts can be stored in the database R Script Repository and invoked by name in calling SQL statements. This can be done dynamically too. The output of such execution can be structured data, XML representation of R objects and graphics, or PNG graphics via BLOB columns in Oracle table(s)..**
- Predictive Analytics – ORE supports three types of predictive analytics – ore.eda for exploratory data analysis, ore.dm for data mining, and ore.predict for additional predictive analytics. More information can be found at <http://www.oracle.com/technetwork/database/database->

ORE - Primary R objects

- `data.frame` – data participating in R operations
- `ore.frame` – objects of a subclass of `data.frame` is unique to ORE that get substituted for database tables.
- Transparent mapping of semantic data mappings from Oracle tables to R `data.frame` objects and vice-versa.
- R scripts in the ORE script repository – R routines that are stored in-database and get executed via Embedded R execution. They get called using one of ore or SQL APIs.
- R datastore that store R data sets in Oracle DB.
- `ore.push`, `ore.pull` that inject and output `data.frame` objects to Oracle DB and `ore.frame` objects to `data.frame` R objects respectively
- The primary ORE packages are listed in Table 1 and the ORE embedded execution APIs are listed in Table 2.

ORE contd....

Table 1 Primary ORE packages

Package	Functionality
DBI	Supplemental packages for Database Interface for R
ORE	Oracle R Enterprise
OREbase	Functionality related to R's base package
OREgraphics	Functionality related to R's graphics package
OREstats	Functionality related to R's statistics package
OREeda	Package for exploratory data analysis
OREdm	Package for data mining (Oracle Data Mining)
OREpredict	Package for scoring data in Oracle database using R Model predictions
ORExml	Functionality for XML generation within R
bitops	Functions for bitwise operations
png	Supplemental package for Read/write png images from/to R and Oracle DB
OBIEEAdvancedAnalytics	Advanced Analytics and Machine Learning R functions supported in ORE for integration with OBIEE

Table 2 Embedded R Execution ORE and SQL APIs

API R Interface	API SQL Interface	Brief Description
In the description of below API, the function f refers to the R function associated with each of the API.		
ore.doEval	rqEval	Executes function f without input data argument
ore.tableApply	rqTableApply	Executes function f with ore.frame input data argument provided via first parameter to f (as data.frame)
ore.groupApply	rqGroupApply	Executes function f by partitioning data as per values in an index column. Each data partition is provided as a data.frame argument to the first parameter of f. Parallel execution of each call of f is enabled
ore.rowApply	rqRowEval	Executes function f by passing a chunk of rows of the provided input data. Each input chunk of data is passed as a data.frame to the first parameter of f. Parallel execution of each call of f is enabled
ore.indexApply	N/A	Executes function f without input data argument but with an index of the execution calls 1 through n where n is the number of function calls. Parallel execution of each call of f is enabled
ore.scriptCreate	sys.rqScriptCreate	Stores the R script into the ORE R script repository with the given name and the associated function
ore.scriptDrop	sys.rqScriptDrop	Drops the R script from the ORE R script repository with the given name
ore.scriptLoad		Loads the R script with the given name from the ORE R script repository for subsequent execution of the function associated with the R script

Building and scoring a model using *Random Forest ML classification*

- Building a Random Forest R classification model for predicting Product Source
- Input – CSV containing Product related data with **one predictor variable or independent variable called Product rank** and **one response or outcome variable called Source (e.g., Source of product)**. The Product is predicted based on the product rank and classified as one of SourceA, SourceB, or SourceC
- Methodology
- Use R's *randomForest()* ML algorithm executes inside one of ORE's spawned R engines.
- Model Importance variables *MeanDecreaseAccuracy* and *MeanDecreaseGini*
- Plot the model, the importance variables, and pairs plot of the 'Source' vs. model 'prediction'
- ORE code and output plots are shown in the demo – Demo1.R.

Building and scoring a model using Random Forest ML classification contd....

```
library(ORE)
ore.connect(user="<username>", sid="<SID>", host="localhost",
password="<pw>")
ore.doEval(function () { library(randomForest)
productdata <- read.csv("productdata.csv", header=TRUE, sep=',')
head(productdata)
productdata$Source <- ifelse(productdata$rank == 1, 'SourceA',
ifelse(productdata$rank == 2, 'SourceB',
ifelse(productdata$rank == 3, 'SourceC', '')))
productdata$Source <- as.factor(productdata$Source)
```

Building and scoring a model using Random Forest ML classification contd....

```
set.seed(123)
sample_size <- 0.70 * nrow(oriductdata)
sampleset <- sample(seq_len(nrow(productdata)), sample_size)
training_set <- productdata[sampleset, ]
test_set <- productdata[-sampleset, ]
formula <- Source ~ . - rank
Source.rf <- randomForest(formula, data=training_set, ntree=100,
importance=TRUE, proximity=TRUE) # Building the Random Forest Model
head(Source.rf)
print(Source.rf) # Print RF model to see important features
plot(Source.rf) # Plot RF Model to see the corresponding RF model graph
importance(Source.rf) # See the importance of the variables
```

Building and scoring a model using Random Forest ML classification contd....

```
varImpPlot(Source.rf) # Plot RF Model see the variable Importance
SourcePred <- predict(Source.rf, newdata = test_set) # Scoring the model
head(SourcePred)
table(SourcePred, test_set$Source)
plot(margin(Source.rf, test_set$Source)) # Plot margin of error for accuracy
library(AppliedPredictiveModeling)
transparentTheme(trans = .4)
pairs(table(SourcePred, test_set$Source), main="Product Source
Prediction")
}, ore.graphics=TRUE, ore.png.height=600, ore.png.width=500)
```

End-to-end implementation of Random Forest algorithm using the business use case of sourcing Product

1. Create a PL/SQL block that contains a R script that gets stored in ORE R script repository. This script builds and scores the Random Forest model for predicting Product Source using the ORE SQL Interface. The corresponding code is in file Demo2.sql
2. Create a second PL/SQL block that prepared the input data and calls the above script using the ORE SQL API - the output of which can be used to generate a PNG graph of the prediction. The corresponding code is in file Demo3.sql
3. Verify the output of CallSourcePred, run the below SQL to get a tabular output:

select *

from table(rqEval(

NULL,

'select CAST("a" as VARCHAR2(50)) "SourcePred", CAST("b" AS VARCHAR2(50)) "Var2", 1 as "Freq" from dual',
'CallSourcedPred'));

End-to-end implementation of Random Forest algorithm using the business use case of sourcing Product

4. Run a query that generates an IMAGE column that stores the RF output as a graph in PNG format:

```
select *  
from table(rqEval(  
    NULL,  
    'PNG',  
    'CallSourcePred'));
```

The graph output is same as the one obtained earlier.

Integration with OBIEE

Pre-config steps required for OBIEE 12c integration

Before creating a new repository or downloading the SampleAppLite repository from the OBIEE12c server, a few pre-config steps must be done so that OBIEE 12c can communicate with the test database and schema.

1. In the NQSConfig.INI file located in the

<FusionMiddlewareHome>\user_projects\domains\bi\config\fmwconfig\biconfig\OBIS folder, make the following changes:

In the [ADVANCE_ANALYTICS_SCRIPT] section:

i. set the value of TARGET = "ORE";

ii. set the value of connection pool as

CONNECTION_POOL = "ORCL"."<uname>/<pwd>:1521";

Make sure the ORCL alias is defined in the tnsnames.ora file of the Oracle DB 12c server.

2. Restart Weblogic server and the associated services by issuing the stop.cmd command followed by start.cmd command. These are found in <FusionMiddlewareHome>\user_projects\domains\bi\bitools\bin folder.

3. Restart Oracle Business Intelligence service (from Taskmanager-> services if on Windows)

4. Next download the server RPD file corresponding to SampleAppLite using the following command:

```
<FusionMiddlewareHome>\user_projects\domains\bi\bitools\bin\datamodel.cmd downloadrpd -O obieenew.rpd -W  
<Admin pw> -U weblogic -P <weblogic_password> -SI ssi
```

Integration with OBIEE contd....

- Integration steps in a nutshell – shown in detail in Demo
 1. Open downloaded .rpd in OBIEE Admin tool. In the Physical Layer, choose New Database, specify values; In the Connection Pools tab, create new connection pool with data source as the complete TNS descriptor string for the alias ORCL and username/password.
 2. Create new Physical Schema followed by New Physical Table. In the New Physical Table dialog box, specify name as **‘CallSourcePred’**, table type as **‘select’** and Initialization String as the following query:
select id, image from table(rqEval(NULL, 'PNG', 'CallSourcePred'))

Integration with OBIEE contd....

3. Click the New Schema created and drag-n-drop to the Business Model and Mapping Layer. Edit the lookup expression builder for image column as follows:

```
lookup("orcl_db","","<new schema  
name>"."CallSourcePred"."image","orcl_db"."<new schema  
name>"."CallSourcePred"."id")
```

4. Remaining steps to complete the Presentation Layer as shown in demo.

5. Save the .rpd and upload it to OBIEE using the uploadrpd command.

Integration with OBIEE contd....

6. Open the OBIEE Presentation Services. Click on Administration->Reload Files and Metadata. Then create an Analysis by clicking New -> Analysis. The new schema created in OBIEE Admin Tool now appears under available Subject Areas.

7. Drag-n-drop the id and image columns from the CallSourcePred under the new schema subject area to the Selected Columns area. Click on Results. The pairs plot of the Product SourcePrediction is displayed.

- To create a new dashboard, click New->Dashboard and specify <new schema name> Dashboard as the name and choose a folder name where it will be saved. For this a folder in My Folders or Shared Folders can be chosen.
- In the Catalog section on below left, choose the analysis named "Pairs Plot of Product Source Prediction", and drag-n-drop the same onto the empty area to the right of the Dashboard. Click Run to view the Dashboard. The pairs plot is displayed as the output.

Uses of the BI Dashboard for actionable decision-making

- Predicting Product source by analyzing the pairs plot enables the business to
 1. Influence the propensity to buy
 2. Forecasting yield based on Source
 3. Forecasting Product Quality
 4. Recommending Product usage

Propensity to buy can be predicted by using another ML algorithm called Logistic Regression or the Generalized Linear Model (GLM). This can be done in OBIEE using the built-in Regression Analytics function or using R or ORE based Logistic regression model and then using the probabilistic value to determine the outcome.

Forecasting yield can also be done right in OBIEE using past buying trends in combination with Time series analysis or the Evaluate Script analytics function with a SQL based model for the same obtained by scoring the model. Or a R or ORE function can be used similar to as illustrated for predicting Product Source.

Predicting Product Quality can be done using the same Random Forest ML model based on the Source predictor variable.

Recommending Product usage can be done by using ML algorithm used for recommendation engines and integrating the same OBIEE dashboards.

Demo

Demo

Thanks

Thanks for coming

Further Reading

- Foster Provost & Tom Fawcett, Data Science for Business, O'Reilly Media, Inc., 2013
- [Oracle R Enterprise Documentation Media Library, Release 1.5](https://docs.oracle.com/cd/E67822_01/OREUG/toc.htm), R Enterprise Users Guide, https://docs.oracle.com/cd/E67822_01/OREUG/toc.htm
- <http://www.oracle.com/technetwork/database/database-technologies/r/r-technologies/documentation/documentation-2166653.html>
- <http://analyticsvidhya.com/>
- <https://datascienceplus.com/>
- <https://blogs.oracle.com/r/>

Q and A

Q and A