**Comparsion of RevoScaleR’s regressions:**

Imagine, you need to predict any numerical value e.g. number of produced cars, revenue, sold items etc. Such a task can be answered with any kind of regression model. Microsoft’s machine learning library “MicrosoftML” offers 4 out-of-the-box models:

* rxFastTrees()
* rxFastForest()
* rxFastLinear()
* rxNeuralNet()

In order to predict numerical values, type=”regression” must be specified.

You need Microsoft’s R-Server (**on-premise** or **IaaS**).

**On-premise**: <https://mran.microsoft.com/download/mro-for-mrs/>

Please follow the instruction for installation on the website

**Azure (IaaS):** Search in Azure portal for R-Server and deploy the VM.

Then, you will need an R Client. You can either use Microsoft’s R Client: <https://azure.microsoft.com/en-us/updates/preview-microsoft-r-client/>

or RStudio to develop R-Scripts.

Make sure MicrosoftML is installed.

R>install.packages("MicrosoftML")

**Dataset (rxImport)**

The sample uses “freeny” dataset, which is a simple dataset for statistical purposes. To exchange the dataset, you will need to read your new dataset into R environment. Depending on the dataset type ODBC, CSV, TEXT… you need to choose the according R-function and assign the result into **dataset** variable (line 10).

**Split Dataset for Machine Learning (rxSplit)**

The dataset gets split in 70% Training **dataTrain** and 30% Testing **dataTest**. Depending on your need, you change the portion as required in dataProb (line 16). 70% and 30% for Training and Testing is a good recommendation for the beginning.

**Model Definition**

In line 29, you will see the formula for the model. This line needs to be adapted in accordance to your data and prediction target variable. Pattern: variable ~ variable1 + variable2….+ variableN

**Model Execution**

Line 32 until line 48 contain the machine learning and scoring for four regressions.

**rxDataStep and rxSummary**

Line 51 until line 59 contain the definition of the result dataset and call of rxSummary(). rxSummary() returns descriptive statistics about the regression performances (MEAN, StdDev, MIN, MAX).

**FastTree**

rxSummary("Score.FastTree ~ lag.quarterly.revenue",predictionResults)

Rows Read: 12, Total Rows Processed: 12, Total Chunk Time: 0.001 seconds

Computation time: 0.002 seconds.

Call:

rxSummary(formula = "Score.FastTree ~ lag.quarterly.revenue",

data = predictionResults)

Summary Statistics Results for: Score.FastTree ~ lag.quarterly.revenue

Data: predictionResults

Dependent variable(s): Score.FastTree

Number of valid observations: 12

Name Mean StdDev Min Max ValidObs MissingObs

Score.FastTree:lag.quarterly.revenue 87.32918 5.662925 78.51682 93.68909 12 0

**FastForest**

rxSummary("Score.FastForest ~ lag.quarterly.revenue",predictionResults)

Rows Read: 12, Total Rows Processed: 12, Total Chunk Time: 0.001 seconds

Computation time: 0.002 seconds.

Call:

rxSummary(formula = "Score.FastForest ~ lag.quarterly.revenue",

data = predictionResults)

Summary Statistics Results for: Score.FastForest ~ lag.quarterly.revenue

Data: predictionResults

Dependent variable(s): Score.FastForest

Number of valid observations: 12

Name Mean StdDev Min Max ValidObs MissingObs

Score.FastForest:lag.quarterly.revenue 87.01722 5.260585 79.13412 92.96806 12 0

**NeuralNet**

rxSummary("Score.NeuralNet ~ lag.quarterly.revenue",predictionResults)

Rows Read: 12, Total Rows Processed: 12, Total Chunk Time: Less than .001 seconds

Computation time: 0.001 seconds.

Call:

rxSummary(formula = "Score.NeuralNet ~ lag.quarterly.revenue",

data = predictionResults)

Summary Statistics Results for: Score.NeuralNet ~ lag.quarterly.revenue

Data: predictionResults

Dependent variable(s): Score.NeuralNet

Number of valid observations: 12

Name Mean StdDev Min Max ValidObs MissingObs

Score.NeuralNet:lag.quarterly.revenue 86.24728 2.971643 81.29696 90.15542 12 0

**FastLinear**

rxSummary("Score.FastLinear ~ lag.quarterly.revenue",predictionResults)

Rows Read: 12, Total Rows Processed: 12, Total Chunk Time: Less than .001 seconds

Computation time: 0.002 seconds.

Call:

rxSummary(formula = "Score.FastLinear ~ lag.quarterly.revenue",

data = predictionResults)

Summary Statistics Results for: Score.FastLinear ~ lag.quarterly.revenue

Data: predictionResults

Dependent variable(s): Score.FastLinear

Number of valid observations: 12

Name Mean StdDev Min Max ValidObs MissingObs

Score.FastLinear:lag.quarterly.revenue 87.00195 5.238538 78.3579 93.99972 12 0

