REVOLUTION ANALYTICS

Modeling in Revolution R Enterprise for Hadoop Users Module 1: Overview and Review

Overview

After completing this course, you will be able to:

- Conduct predictive analysis on your enterprise data using regression and tree-based models.
- Implement models through embedded scoring functions in Revolution R Enterprise.
- Use advanced algorithms for unsupervised learning and data manipulation such as principal components and clustering techniques.
- Understand key concepts in coding big data functions efficiently.

Algorithm and Function Overview

The basic methods we will cover in this course are listed as follows:

- Linear Regression Modeling and Evaluation
  - Simple Regression
  - Multivariate Regression
  - Complex Formulas and Higher Order Terms
  - Stepwise Regression
- Generalized Linear Models
  - Logistic Regression
  - Additional Forms for General Linearized Models
- Data Mining using Trees and Forests
  - Tree Modeling
  - Random Decision Forest

Algorithm and Function Overview

- Unsupervised Model and Other Techniques
  - Clustering
  - Principal Components
  - Running Simulations

The Data

Throughout this course we will be using the following data set:

- Bank Marketing data set from the Machine Learning Repository at University of California, Irvine

[Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS (http://hdl.handle.net/1822/14838)

The Data: Bank Marketing Data

The Bank Marketing Data Set, which we will refer to as the Bank data, concerns the direct marketing campaigns, or phone calls, of a Portuguese banking institution to its clientele, and the success of those campaigns in causing customers to subscribe to a term deposit. Multiple types of data are collected on each bank client, such as information on one's age and marital status.

Review: Setting up Compute Context

Let's define the correct parameters of our Hadoop cluster so that we can set up our Hadoop Compute Context using Revo R:

```
# This is the same username you use to log on to the linux machine
mySshUsername <- "luba"
mySshHostname <- "master.local"
# Port number of the Hadoop Name Node
myPort <- "8020"
# Host name of the Hadoop Name Node
myNameNode <- "master.local"
# Local location for writing various files onto the HDFS from the local file
# system
myShareDir <- "/home/Ben_Examples"
# The HDFS share file location
myHdfsShareDir <- paste("/user/RevoShare", mySshUsername, sep = "/")
```

Review: Setting up the Compute Context

These commands will create a Hadoop compute context:

```
myHadoopCluster <- RxHadoopMR(hdfsShareDir = myHdfsShareDir, shareDir = myShareDir,
    sshUsername = mySshUsername, autoCleanup = FALSE, nameNode = myNameNode)
```

Then, to set the compute context:

```
rxSetComputeContext(myHadoopCluster)
```

Review: Switching between Local and Hadoop

We can set our compute context using the rxSetComputeContext function.

- To set the compute context to local (to run our computations off of the Hadoop cluster), use the following command:

```
rxSetComputeContext("local")
```

- To set the compute context back to the Hadoop cluster, first define your Hadoop environment:

```
myHadoopCluster <- RxHadoopMR()
```

- Then, execute the rxSetComputeContext function using myHadoopCluster:

```
rxSetComputeContext(myHadoopCluster)
```

Review: Creating a Data Source

Creating a data source, specifying that it is on the Hadoop Distributed File System, first create a file system object that incorporates our NameNode and port (remember RxHadoopMR):

```
hdfsFS <- RxHdfsFileSystem(hostName = myNameNode, port = myPort)
```

In our case, this information may be obtained from the RxHadoopMR command, where: - sshHostname : “master.local” * # Specifies our NameNode - port : 8020 * # Specifies our port

Review: Creating a Data Source

Finally, we can create the data source using the following commands that incorporate the previous HDFS and factor levels steps:

```
DS <- RxTextData(file = inputDir, missingValueString = "missingValueString",
    colInfo = colInfo, fileSystem = hdfsFS)
```

Review: Creating a Data Source

We can get basic information about the data using the the following commands:

```
rxGetInfo(BankDS, getVarInfo = TRUE, numRows = 6)
rxSummary(~., BankDS)
```

Recap

Let's review some of the concepts covered in this module:

- How do you switch your compute context from local to Hadoop?
- How do you create a data source?

REVOLUTION ANALYTICS

Thank you

Revolution Analytics is the leading commercial provider of software and support for the popular open source R statistics language.

www.revolutionanalytics.com, 1.855.GET.REVO, Twitter: @Revo-lutionR