DataRobot R Package

Ron Pearson

DataRobot
Today’s agenda:

1. What is DataRobot?
   a. A Boston-based software company
   b. A massively parallel modeling engine
   c. An R package (today’s focus)

2. An example to demonstrate the R package:
   a. Predicting compressive strength of concrete
   b. Shows typical DataRobot modeling project
   c. Demonstrates partial dependence plots
What is DataRobot?

1: Boston-based software company

2: Massively parallel modeling engine
- On multiple hardware platforms
- Under various operating systems
- Has many software components:
  - R
  - Python
  - … and various others

3: R API client - the DataRobot R package
A few key details:

➢ Package in the final stages of internal testing

➢ Will be released via R-Forge under the MIT license

➢ Two vignettes are available with the package:
  ■ “Introduction to the DataRobot R package”
  ■ “Interpreting Predictive Models, from Linear Regression to Machine Learning Ensembles”
Today’s predictive modeling example:

- Objective: predict the compressive strength of concrete
- Data source: **ConcreteFrame**
- Each record describes one laboratory concrete sample
- Target variable: Strength
- Prediction variables: Age + 7 composition variables
Creating the DataRobot modeling project

1. Upload the modeling dataframe:
   > MyDRProject <- SetupProject(ConcreteFrame,"ConcreteProject")

2. Specify the target variable to be predicted and start building models:
   > StartAutopilot(Target="Strength", Project = MyDRProject)

3. Add a custom R model to the project:
   > CreateCustomModel(MyDRProject, LogAgeFit, LogAgePredict, "R:LinearWithLogAge")

4. Retrieve the leaderboard summarizing all project models:
   > ConcreteLeaderboard <- GetAllModels(MyDRProject)
Function arguments for CreateCustomModel

LogAgeFit <- function(response, data, extras=NULL){
  #
  DF <- cbind.data.frame(data,
    CustomModelResponseY = response)
  model <- lm(CustomModelResponseY ~ . - Age +
    log(Age), data = DF)
  return(model)
}

LogAgePredict <- function(model, data){
  predictions <- predict(model, newdata = data)
  return(predictions)
}
ConcreteLeaderboard: the poorer models (ranks 16-31)
ConcreteLeaderboard: the better models (ranks 1-15)
Understanding complicated models

- Linear regression models - easily explained via coefficients
- Not true for random forests, boosted trees, SVMs, etc.
- Alternative: partial dependence plots (Friedman 2001)
  - To assess dependence on $x_j$, average the predictions over all other covariates
  - Compute this average for a representative range of $x_j$ values and plot
  - Linear model: plot is straight line with slope equal to coefficient of $x_j$
Partial dependence plots for Age
Partial dependence plots for Water.
Even larger Age shift for ENET Blender

Large average Age shift

Importance from permutations: large RMSE increase => important variable
Summary of the concrete strength example

- Best model = ENET Blender
  - Age dependence shows nonlinear hard saturation behavior
  - Water dependence is nonlinear and non-monotonic

- Combining with variable importance results:
  - Average measures: Age > Cement > Blast Furnace Slag > Water
  - ENET Blender measures: Age > Water > Cement > Blast Furnace Slag

- Best model captures Age saturation behavior and nonmonotonic Water dependence that other models can’t
Note the novel water dependence for the best model
Questions?

Vignette - Intro to DataRobot: [bit.ly/1dzgppC](http://bit.ly/1dzgppC)

Vignette - Two Applications: [bit.ly/1KtWCGI](http://bit.ly/1KtWCGI)

Want to use the DataRobot R package?
Email us at: [useR2015@datarobot.com](mailto:useR2015@datarobot.com)