Statistical Modeling & Learning in R

Data Science and Programming Summer Workshop Series

Anh T. Bui
IEMS, Northwestern University

DATA

```
## install the following packages and load them into the environment:
install.packages("mlbench")
require(mlbench)
install.packages("mlmRev")
require (mlmRev)
# to see the list of the datasets in this package, use:
library(help="mlbench")

## load the following packages into the environment:
# for continuous response variable:
data(BostonHousing)      # predict the house price in Boston from house details (medv)
data(BostonHousing2)     # same with BostonHousing, but has 5 additional variables
data(Ozone)              # predict the daily maximum one-hour-average ozone reading (V4).
# for binary response variable:
data(PimaIndiansDiabetes)   # predict the diabetes onset in female Pima Indians (diabetes)
data(Ionosphere)         # predict high-energy structures in the atmosphere (Class)
data(Sonar)              # predict metal or rock returns from sonar return data (Class)
# for multilevel regression:
data(Exam)               # predict students' exam scores
data(InstEval)           # predict lecturers' ratings
# to see details of a dataset (e.g., the BostonHousing dataset), use:
help(BostonHousing)
```

EXERCISES

Exercise 1: Linear Regression

Use the BostonHousing dataset in the **mlbench** package and fit a linear regression model to predict the house prices (medv).

1) Data exploration:
   a) How many continuous and discrete variables are there in the dataset? Are there missing values?
   b) Produce a scatterplot of this dataset.
   c) What is the maximal correlation value between the (continues) variables?
   d) Is it necessary to transform medv?

2) Fit a linear regression model to predict the house prices using all other variables (without variable interactions)
   a) Which variables are statistically significant with 99% confidence level?
   b) What is your model adjusted R^2?
3) Use stepwise regression to select variables.
   a) Which variables are statistically significant with 99% confidence level?
   b) What is your new model adjusted R^2?
   c) Create basic diagnostic plots.
   d) Compute yhat, the predicted values for the training data.
4) Play around with variable interactions.

Homework or if time permits: Repeat with the BostonHousing2 and Ozone datasets in the **mlbench** package.

Exercise 2: Regression Tree

Use the BostonHousing dataset in the **mlbench** package and fit a regression tree to predict house prices (medv).

- Grow a tree with:
  - at least 5 observations in each leaf node
  - complexity parameter cp=0.001
  - 5 folds in the internal cross-validation (CV).
- Tree outputs:
  - Plot the tree. Which variables are shown in this tree?
  - What are the three most important variables in this tree? Do they agree with what we see in the plotted tree?
  - What are the predicted value and the residual sum of square of the first leaf node? What is the rule of this leaf node?
- Tree pruning
  - Plot (relative) CV error against tree size.
  - Prune this tree if necessary.
- Predictions:
  - Compute the predicted values for the training data.
  - Create some missing values in the training data (for variables shown in the plotted tree) and compute the new prediction values. Are they different with the predicted values without missing values? Are they the same with what you can infer from the tree summary? Hint: to quickly change some values in a data frame, we can use the function fix("data frame name").

Homework or if time permits: Repeat with the BostonHousing2 and Ozone datasets in the **mlbench** package.

Exercise 3: Random Forests

Use the BostonHousing dataset in the **mlbench** package and fit a random forest to predict house prices (medv).

- Impute the data if needed
- Check the type of the response variable
- Find the optimal "mtry"
- Fit a random forest with 100 trees
- Is it necessary to add more trees? Add more trees if needed.
- What is the model R^2?
- Which variables are important?
- Plot partial dependence plots on some variables.
- Compute predicted values for the training data.

Homework or if time permits: Repeat with the BostonHousing2 and Ozone datasets in the **mlbench** package.

Exercise 4: Binary Logistic Regression

Use the PimaIndiansDiabetes dataset in the **mlbench** package and fit a logistic regression model to predict the diabetes onset female Pima Indians.

- How many continuous and discrete variables are there in the dataset? Are there missing values?
- Fit a logistic regression model.
- Create basic diagnostic plots.
- Produce the model outputs.
- Compute the predicted probabilities for the training data.

Homework or if time permits: Repeat with the Ionosphere and Sonar datasets in the **mlbench** package.

Exercise 5: Decision Tree

Use the PimaIndiansDiabetes dataset in the **mlbench** package and fit a decision tree to predict the diabetes onset female Pima Indians.

- Grow a tree with:
  - at least 5 observations in each leaf node
  - complexity parameter cp=0.001
  - 5 folds in the internal cross-validation (CV).
- Tree outputs:
  - Plot the tree. Which variables are shown in this tree?
  - What are the three most important variables in this tree? Do they agree with what we see in the plotted tree?

  - What are the predicted class of the second leaf node? What is the rule of this leaf node?
- Tree pruning
  - Plot (relative) CV error rate against tree size.
  - Prune this tree if necessary.
- Predictions:
  - Compute the predicted classes for the training data.
  - Create some missing values in the training data (for variables shown in the plotted tree), and compute the new prediction values. Are they different with the predicted classes without missing values? Are they the same with what you can infer from the tree summary?

Homework or if time permits: Repeat with the Ionosphere and Sonar datasets in the **mlbench** package.

Exercise 6: Random Forests

- Impute the data if needed
- Check the type of the response variable
- Find the optimal "mtry"
- Fit a random forest with 100 trees
- Is it necessary to add more trees? Add more trees if needed.
- What is the model classification error rate?
- Which variables are important?
- Plot partial dependence plots on some variables.
- Compute predicted classes for the training data.

Homework or if time permits: Repeat with the Ionosphere and Sonar datasets in the **mlbench** package.

Exercise 7: Multilevel Regression

Use the Exam dataset in the **mlmRev** package and fit multilevel models to predict students' exam scores.

- Data exploration:
  - How many continuous and discrete variables are there in the dataset?
  - Plot normexam against standLRT for each school. Is multilevel modeling necessary?
- Fit a varying-intercept with **fixed-mean** model using school as the group: $\text{normexam}_i = \alpha_{j[i]} + \varepsilon_i$,
- Fit a varying-intercept with **fixed-slope** (for standLRT) model using school as the group: $\text{normexam}_i = \alpha_{j[i]} + \beta \text{standLRT}_i + \varepsilon_i$

- Fit a varying-intercept and varying-slope (for standLRT) model using school as the group: $\text{normexam}_i = \alpha_{j[i]} + \beta_{j[i]}\text{standLRT}_i + \varepsilon_i$
- Which model is the best according to the REML criterion?
- Produce basic diagnostic plots and predictions for the training data.
- Play around to see if more variables can be added to the existing model.

Homework or if time permits: Repeat with the InstEval dataset in the **mle4** package.

Exercise 8: Model Selection using Cross-validation (Optional)

Cross-validation (CV) is a useful method to estimate the expected prediction error, and hence can be used to select the best (in terms of prediction accuracy) model. Use k-fold CV to select the best model for the problems in Exercises 1—3.

The correct way to select model using CV is that the (random) division of the data into k disjoint subsets must be done at the very beginning (i.e., before variable selection). In addition, each comparison of the CV error must be done using the same division (e.g., randomly dividing the data into 5 folds to compute CV error for a linear regression model and then randomly dividing the data into 5 different folds to compute CV error for a regression tree is not good).

For simplicity the following procedure can be used to select the best model:

```
# randomly divide the data into 5 disjoint subsets.

for (rep in 1:num_reps)

   for (i in 1:5) {

     # prepare a training set that includes subsets that are not subset i.

     # use subset i as the test set


     # fit the best linear regression model you can using the training set

     # evaluate the error rate of the best linear regression


     # fit the best regression tree you can using the training set

     # evaluate the error rate of the best regression tree


     # fit the best random forest you can using the training set
```

```
    # evaluate the error rate of the best random forest
  }
  Compute the average CV errors for the inner loop
}
```

Compute the average CV errors for the outer loop and select the model that has smallest average CV error (of the outer loop).