Industrial Organization and Data Science

Instructors: Justin Rao, Affiliate Professor & Senior Researcher, Microsoft

Jacob LaRiviere, Affiliate Professor & Senior Researcher, Microsoft

Emails: justin.rao@microsoft.com

jlariv@microsoft.com

Course Assignments & Reading

Course assignments should be printed (code, output and descriptive answers) and turned in at the start of class unless otherwise noted. Feel free to work in groups but everyone is required to turn in their own work with answers written in your own words. In both calculations and complex ideas, write down each step of logic used in reaching your conclusion. Keep in mind that in most cases a good answer is one precise sentence; quality is heavily favored over quantity. This will be graded on a full credit, half credit and no credit basis. All work must be typed

Discussion questions do not need be written out ahead of time. At the beginning of each class the professors will lead a discussion around these questions. Students will be called on, potentially at random, to add their insight. This part of class will contribute heavily to your course participation grade.

Week 5, due March 4

This homework will also serve as a useful study guide for the midterm.

Assignment to be turned in. Please turn in your R output and answers to the questions.

Let's return to the orange juice assignment and investigate how store demographics are related to demand and how we can improve model fit.

1. Let's split our data into a training set and a test set. An easy way to do this is with the sample command. The following will randomly select 20% of the rows in our data frame
   indexes = sample(1:nrow(oj), size=0.2*nrow(oj))
2. Now let's use this index to create a training and a test set, try:
   ```
   OJtest=oj[index,]
   Ojtrain=oj[-index,]
   ```

   What did this do? How many rows does the test set have? How many rows does the training set have?

3. Now let's run the very simple model `logmove ~ log(price) + brand` on the training data.
   a. Use LM on this model and report the R-squared
   b. Use `predict(model, Ojtest)` to predict log sales for the test set.
   c. Compute `cor(predicted_sales,logmove)^2` on the test set. This is our "honest R-squared". How does it compare to the value in (a)?

4. Now let's run better models.
    a. Run our "previous favorite" logmove ~ brand*log(price)*feat on the training data. Use LM to get regular R-squared. Now, follow the procedure in (3) to compute "honest R-squared". What is it? How do they compare?
    b. Now add in all the demographics, as in the last HW. What is the regular R-squared on training data? What is the honest R-squared on the test set?
5. Now let's use a regression tree and see how well we can do. Load the package rpart. Note rpart cannot handle interaction terms as input as this accommodated with the tree structure.
    a. On the training set, use rpart to run a model of `log(price)+brand+feat`. Specify `method="annova"`. See Lecture 5, slide 56 for tips.
    b. Compute the fitted values on your test set as: `testset$fitted=predict(mymodel, data=testset,type="vector")`. Now compute the correlation with logmove and square it to get "fair R-squared" (as before). How well did this model do? How does it compare to how did before?
    c. Plot the fitted values vs. the truth in the test set plot(fitted,logmove, data=test). What do you notice? What is going on with the fitted values?