Homework 4: OLS vs Random Forest

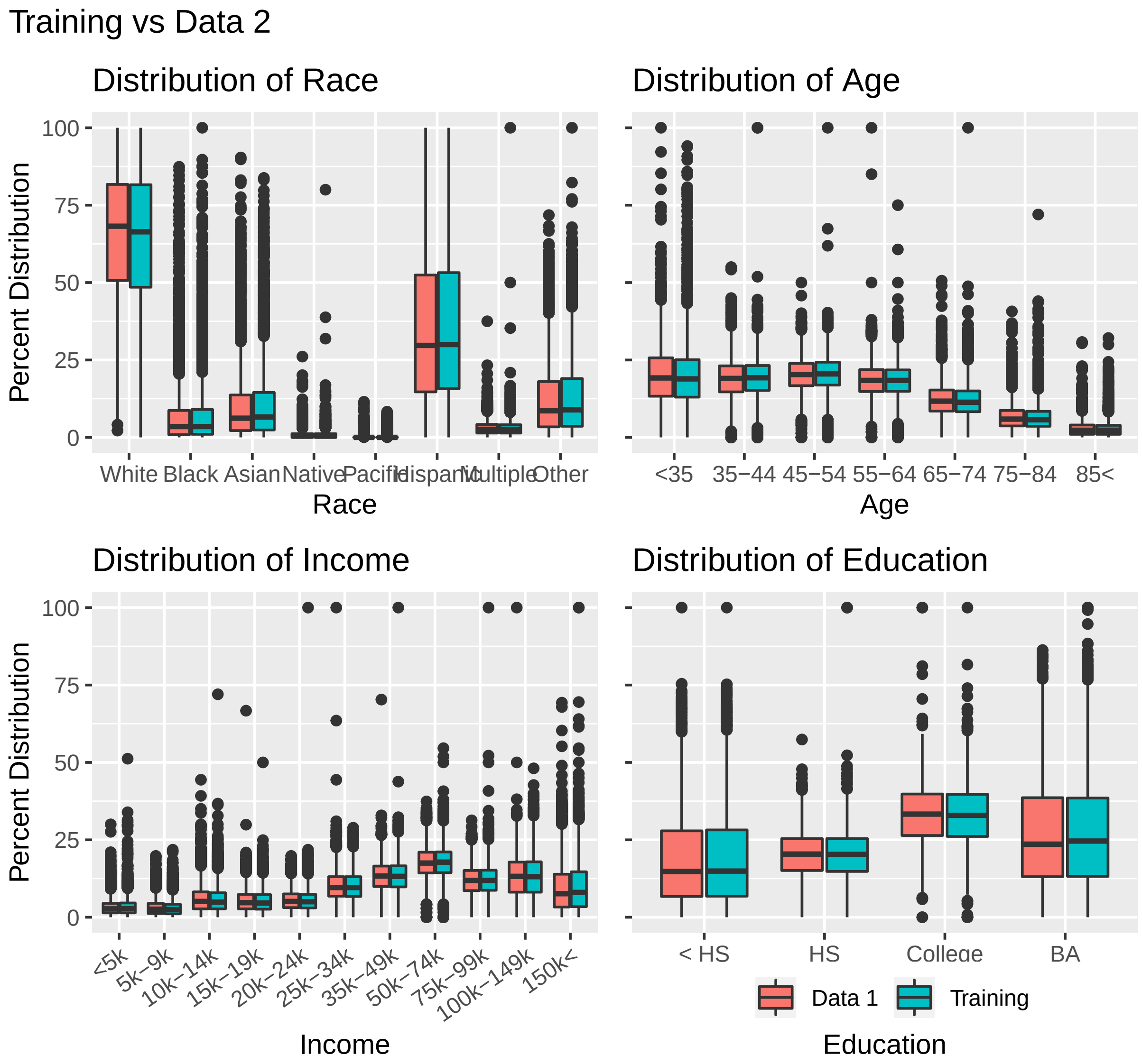
A battle for the ages

Neeraj Sharma

06/01/2020

1) OLS prediction for median housing costs in data set 2

In order to effectively apply the model I train on the training data set to experimental data set 2, I first need to substantiate my belief that the two data sets are similar. In order to accomplish this, I compare the distribution of several variable classes that appear in both data sets.



Across the board, data set 2 and the training set look very similar. This means that a model I train on the entirety of the training dataset will be able to be applied to data set 2 without much trouble.

a. Describe both the regression you ran and the thinking that underlay the choices of what to put in your model.

I approach creating my OLS model two ways. The first way was through using a LASSO regression to understand what variables function as effective predictors, and the second approach was to intuitively reason which variables will contribute to housing prices. In Eric's Office Hours, numerous students discussed the pros and cons of LASSO, and I was interested in trying it out to improve my skills and to identify any variables that might unexpectedly function as good predictors.

In Appendix 1, I estimate a LASSO model with the method specified in Springer Statistical Learning. This lasso finds that over 20 variables are consistently explanatory for MedianMonthlyHousingCosts. Obviously, this is an over fit. Even out of ~180 variables, estimating a model with over 20 variables is too many degrees of freedom to provide a reasonably intuitive model. This fit does reveal several important factors:

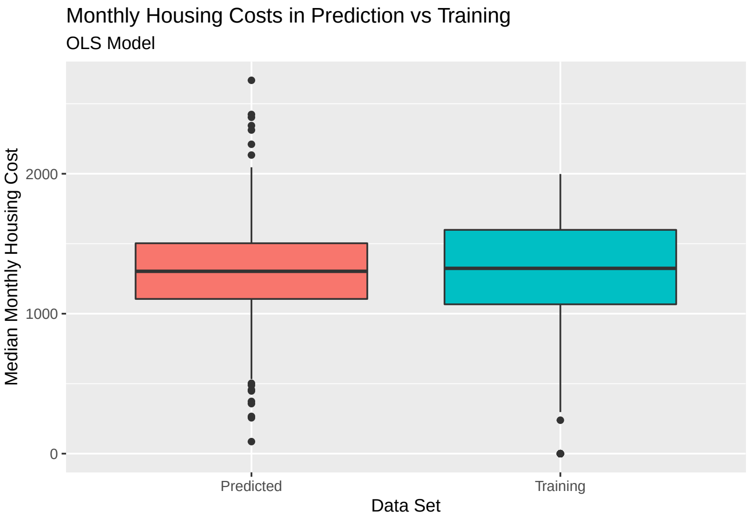
1. Income is key. The regressor with the highest coefficient was constantly Income1000000_149999. This is a uniquely high income bracket as far as the training data set goes. This means that the number of rich people in a given area is correlated with the household prices. This intuitively makes sense as areas with more rich people will tend to have higher housing prices. Furthermore, income generally is concentrated in pockets, so rich areas are typically more uniformly rich (inflating housing prices rapidly).
2. There appears to be an artificial ceiling set on MedianMonthlyHousingCosts. No observation in the training dataset cost over 2000 dollars per month. This means that the regression becomes flatter on the higher end than one might expect otherwise as further increases in representative covariates do not in turn result in higher Housing Costs.
3.

I use the insight I gain from the LASSO to generate the following model. The model combines the machine precision of the LASSO along with realistic intuition. The income variables are included based on the notion that increases in income are associated with increases in housing cost. As people have more disposable income, they spend more on housing. Furthermore, I include Income100000_149999 as a proxy for wealth concentration. MeanTravelTimeToWorkMin is included because it is highly represented in the LASSO. I include SNAP data to understand poverty and supplemental aid, as that likely impacts housing cost as well. Finally, I include variables to control for and quantify the impact of race on housing cost.

MedianMonthlyHousingCosts ~ Income100000_149999 + MedianHouseholdIncome + MeanTravelTimeToWorkMin + snap + BAOcc + pctRetiringAge + WhiteNotHispanicOcc + pocOcc + unemp.

Variable	Estimate	Standard Error	T Statistic	P Value
(Intercept)	952.21	65.75	14.48	0.0000000
Income100000_149999	13.08	1.04	12.63	0.0000000
MedianHouseholdIncome	0.01	0.00	14.49	0.0000000
MeanTravelTimeToWorkMin	3.92	0.73	5.37	0.0000001
snap	-1151.83	85.87	-13.41	0.0000000
BAOcc	2.63	0.36	7.23	0.0000000
pctRetiringAge	-4.14	0.46	-9.00	0.0000000
WhiteNotHispanicOcc	-2.09	0.59	-3.57	0.0003607
pocOcc	-0.52	0.44	-1.18	0.2373926
unemp	-30.57	51.01	-0.60	0.5490641

b. Guess what your performance will be in terms of R-squared and beta, when, using data set 2 we run a regression of the form: y = a + beta*y-hat where y-hat is your predicted housing costs and y is the true housing costs. We'd like numeric answers for both the r-squared and beta. Emphasize the logic of why you guessed your guesses.



Given that this regression has an R2 value of 0.54, I'd project a similar value for the R2 when we run y = a + beta*y-hat. The close similarity between the training set and data set 2, as shown in my demographic plots above, implies that the distribution of MedianMonthlyHousingCosts will also resemble the data I trained this model on. That might or might not be a fair assumption to make given MedianMonthlyHousingCosts appears to cap at $2000.

I believe that my beta value is somewhere in the range of 0.7-0.8. This means that I think the overall slope of my predictions does not increase quickly enough as the covariates move upwards, on net. The model is clearly trained on data that has an artificial cap set at $2000 for MedianMonthlyHousingCosts. What that means is that the higher values are not included in the data set so are dragged down dramatically. Thus, when compared, I believe that my model will underestimate more as MedianMonthlyHousingCosts increases.

2) Random Forest prediction for median housing costs in set 2

a. Describe both your model (as in, the regression you ran) and the thinking that underlay the choices of what to put in your model.

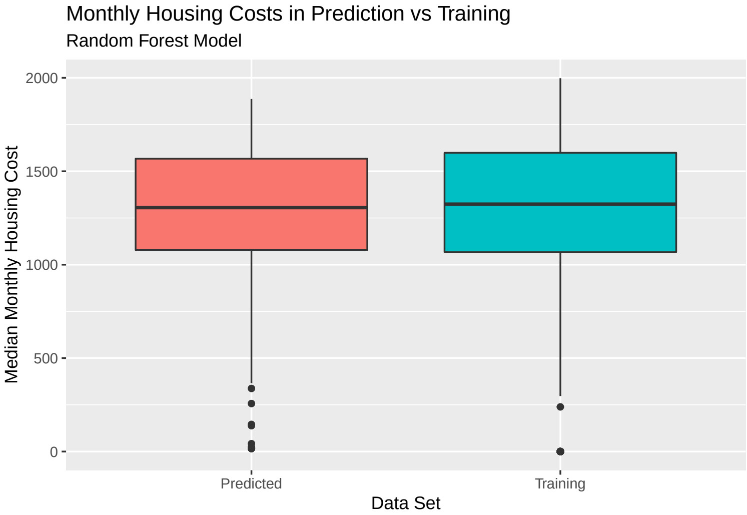
For my random forest, I predicted the model on the training data set over night in a separate R script with Median Housing Cost on the left side an all other predictors on the right side. I then saved both the random seed and output of that process to my repository and then load them here.

I chose to run my random forest over all predictors instead of a small subset of predictors because of the ensemble method behind random forests. An issue with estimating decision trees is that any individual split can be critical and suddenly drive the output of the tree to a locally (but not globally) optimal level. When you run only one tree, random chance and a small sample size mean that it's possible to rely extremely heavily on a single predictor or overfit the model. Random forests control for this by aggregating trees, ensuring that only features that repeatedly emerge as important are considered as such. This means that it's theoretically legitimate to run the model over all regressors, as those with more predictive ability overall are selected equally as those that lack predictive ability. I till thing overfitting occurs with my approach, however.

Unfortunately, the output of the random forest exceeds the memory capacity of TeX so I'm unable to get good output. To visualize this, you can open the RMD and then run the following code chunk and get some good insight.

```r
load("~/Desktop/ECON21300/Homework 4/Saved/trainingrf.RData")
load("~/Desktop/ECON21300/Homework 4/Saved/trainingrf_oob.RData")

trainingrf <- train(MedianMonthlyHousingCosts ~ .,
                        data = training,
                        method = "rf",
                        ntree = 600)

trainingrf_oob <- train(MedianMonthlyHousingCosts ~ .,
                    data = training,
                    method = "rf",
                    ntree = 600,
                    trControl = trainControl(method = "oob"))

# These cause the tex to vomit.
trainingrf$finalModel
randomForest::varImpPlot(trainingrf$finalModel)

rf_predictions <- data1 %>%
  mutate(Proj = predict(trainingrf, newdata = .)) %>%
  select(Proj, everything())
```

b. Guess what your performance will be in terms of R-squared and beta, when, using data set 2 we run a regression of the form y = a + beta*y-hat where y-hat is your predicted housing costs and y is the true housing costs. We'd like numeric answers for both the r-squared and beta. Emphasize the logic of why you guessed your guesses.



According to the model, 83.4% of out-of-bag samples are properly explained by the random forest model I generate. Because the testing data set closely models the distribution of variables in the training data set, I think that an arbitrary bag from the training data set will also approximate the distribution in the testing data set. I believe that can be proven via the central limit theorem more rigorously, but that's a little besides the point. Thus, I believe that my model will have an R2 value of around 0.8. I imagine it will be slightly lower to be conservative in my estimate. I am also worried about the impact of multicolinearity.

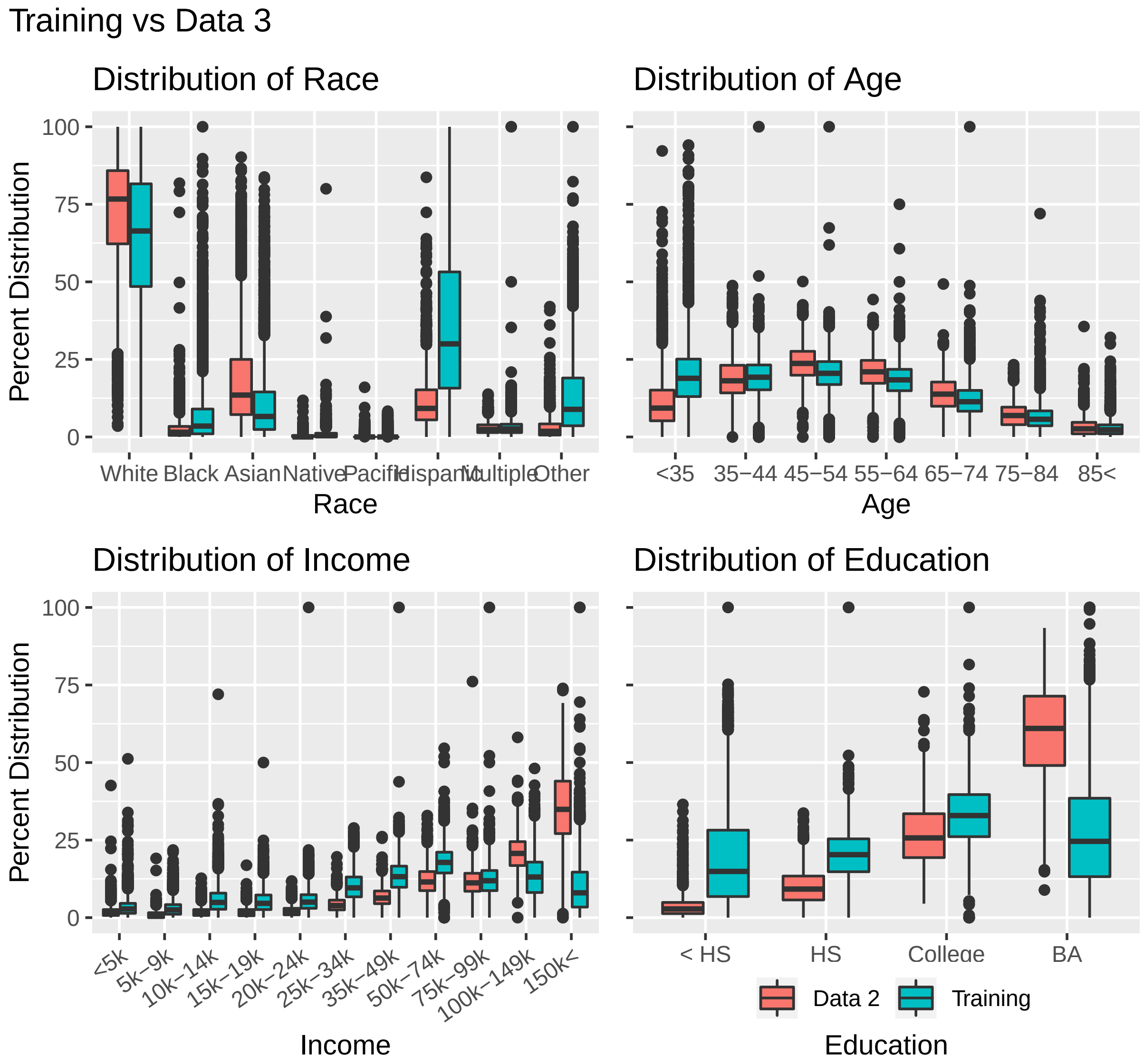
Secondly, I believe that my beta value will be close to 1 for the random forest. While it might systematically over or underestimate the housing cost (which would be represented by a change in alpha), I believe that the random forest's bootstrapping algorithm and success in out-of-bag modeling implies that it is sufficiently capable at modeling more diverse growth trends of covariates than OLS. Logically, the model was success over a diverse subsets of its own training data. The data set we are attempting to analyze resembles those subsets. Thus, I believe it accurately captures the underlying relationships.

c. Which do you think will do better in out-of-sample predictions, random forest or OLS?

I think that my random forest will provide a better fit to data set 2 than the OLS model. This is primarily due to the fact that I think my OLS model is severely underestimating the upper end of Housing Cost due to the cap it is forced to include in its fitting. Random forests are able to bootstrap samples unlike OLS which means that the effect of this top-end underestimation will be reduced. I think that is a key difference between OLS and random forest in this instance that will make random forest superior. Even though I believe my random forest is overfitting, I think the relative impact of the training sampling affects OLS more.

3) What happens when your prediction data doesn't mirror your test data?

In comparison to data set 2, data set 3 looks very different than the training data set.



If I were to run the model I define over the entire training data set and use to solve the first two problems on data set 3, I will not draw effective conclusions as the underlying samples do not overlap. I need to train a model on a subset of the training data frame that resembles data set 3 in order to accurately effectively predict the median housing costs of homes in data set 3.

a. OLS - Describe your modeling approach and the thinking that underlies your choices. Specifically, make sure you address what you did to go from a good prediction in-sample (training data) to a good prediction out-of-sample (data set 3).

Given that the data given as training data is clearly is not representative of the data we are given in set 3, I produce a subset of training data that as closely approximates the distribution of variables in data set 3 as I can produce while still achieving a large sample size. I select between 5% and 30% of the top richest observations in the training data set as a bootleg version of formal bootstrapping. My hypothesis is that data set 3 comes from a richer area given that the data is overwhelmingly skewed towards people with incomes over $150,000. Obviously, this is a flawed model as it will be biased towards the lower end of some variables, and the higher end of some other variables, but this spread will be significantly less than the spread that currently exists with the training data compared to data set 3.

I choose the variables for my OLS here based less on intuition and more on predictive power to attempt to minimize the impact sampling bias can possibly have.

b. RF - Describe your modeling approach and the thinking that underlies your choices. Specifically, make sure you address what you did to go from a good prediction in-sample (training data) to a good prediction out-of-sample (data set 3).

I cannot use the same approach as I used for OLS because the data set will then be too small to run a forest over. I choose to take a subset of the data that is as large as possible while still allowing it to generate at least 200 trees. In my experimentation, I found that was the top 43% of income data.

This is also not a very precise method, but it's the best I could come up with.

c. Which do you think will give better out-of-sample predictions now, random forest or OLS?

Given that my attempt at bootstrapping is significantly more effective in the OLS model because I have enough data to modify the distribution meaningfully, I think that OLS will be more accurate than the Random Forest at predicting for data set 3.

d. Explain why answering this question is harder with random forest than with OLS.

We simply do not have a large enough sample realistic training data we can pull from the training set provided to bootstrap enough decision trees that model the distribution of data set 3. OLS, on the other hand, is able to work with any N sample size large enough. Because resampling is not an issue with OLS, working with OLS when you don't have a realible training set to work with is easier.

My method to solve this problem is a good attempt, but probably falls short. I don't think my method of validating if the subsetted data is closer to data set 3 is effective and reliable, which is important for OLS.

Appendix 1 - Exploratory LASSO

Table 1: LASSO Model as Exploratory Analysis

Variable Name	Coefficient
(Intercept)	902.948
Income100000_149999	11.259
Income75000_99999	2.306
MeanTravelTimeToWorkMin	1.433
IncomeMore150000Owner	0.925
Households150000_199999	0.372
BlackOwnerOcc	0.269
Income50000_74999Renter	0.223
X45to54YrsOwnerOcc	0.170
PopScientificProfessional	0.119
PopSelfEmployed	0.082
PopSalesOfficeJobs	0.041
PopSalaryJobs	0.039
PopOtherExceptGovernment	0.032
PopManufacturing	0.005
MedianHouseholdIncomeRenter	0.002
MedianHouseholdIncome	0.001
MedianHouseholdIncomeOwner	0.000
MedianFamilyIncome	0.000
MedianNonfamilyHouseholdIncome	0.000
Households10000_14999	-0.051
PopAgForestryFishHuntingMining	-0.105
Income10000_14999Owner	-0.135
pctRetiringAge	-0.215
WhiteNotHispanicOwnerOcc	-0.247
WhiteOwnerOcc	-0.317
X65to74YrsOcc	-0.325
HSOcc	-0.730
Income15000_19999Owner	-1.516
NativeOcc	-2.329
Income20000_24999	-3.572
Income15000_19999	-3.625
Income10000_14999	-4.296
snap	-585.742

Appendix 2 - Producing the combined DF we need to submit

```r
data2_ols <- ols_predictions %>%
  select(random_id, ols_predictions = Proj)
data2_rf <- rf_predictions %>%
  select(random_id, rf_predictions = Proj)

data2_complete <- inner_join(data2_ols, data2_rf)

data3_ols <- data3_ols_predictions %>%
  select(random_id, ols_predictions = Proj)
data3_rf <- data3_rf_predictions %>%
  select(random_id, rf_predictions = Proj)

data3_complete <- inner_join(data3_ols, data3_rf)

submission <- bind_rows(data2_complete, data3_complete)

write_csv(submission, here("Homework 4", "neeraj_predictions.csv"))
```