Amber Gonzalez-Pacheco

Matheus Grover

Can We Predict Income?

Introduction

We want to know what the income of a third generation American is based on personal and familial demographics? The American Dream is a concept that anyone, regardless of where they come from can achieve success and increase their economic status through their hard work. We were tasked by the United States Immigration Agency to provide the world with predictions of American citizen's income using their education opportunities, family income, and other demographic data to promote the American Dream all over the globe.

Data

We are using the Panel Study of Income Dynamics dataset, which is a study that tracks families over time. Our outcome is the third generation's logged income, and we will be testing our prediction models using regressors such as each generation's education, the first and second generations income, race and sex. The education regressor includes the categories of having a college education, some college, high school education, and less than high school. The race regressor includes white, black, and other, and the sex regressor includes Female and Male. Because these are categorical variables they needed to be factored in order to be worked with in our model. Some limitations of the data are how limited the explanatory variables are. Income is not solely defined by a person's education, race or sex, there are many other omitted variables. Because of this our predictive model never manages to get as close as we'd hope to the actual income of the third generation.

An observation we made was that there is an overall positive correlation between higher levels of education in generation 2 on both the second and third generation's income. This pattern was found by creating a scatter plot (Graph #1 [left]) and identifying each different level of education by a color, confirming what we previously thought.

Since our data set includes all three generations' education as well, we wanted to know which education would be the most predictive of our outcome. In *section 1* of our code we created a simple OLS model to test this. We created indices for each education category, but left out one for that would be explained in the intercept. We found that the second generation's education was more positively correlated with the third generation's income which makes sense empirically because it is closer. Surprisingly, the college education coefficient for generation 1 had a negative correlation with the third generation's income, which led us to run another scatter plot similar to the first (Graph #1 [right]). In it, we found that the effect of generation 1's education seems less dependent on level and more random. Having less than a high school education seems the most common, and it extends through the whole plot, which is very different from the previous plot of generation two's income.

An important note is the dataset includes id numbers for generation 1 and 2 which we dropped from the dataframe because we only need the generation 3 id number. Generation 3 id number is also periodically dropped from dataframes so that it isn't used for estimations.

Methodology

The goal of prediction is to find a model that fits the new data well, not just fitting the old data. First we start out by finding which is the better model to run in our regressions. We do this manually creating a training set using cross validation in *section 2* of the code. We split the data

into four even sized folds and randomly reshuffled the folds. Then we run both the full model and a sub version of the model where we chose to only include generation 2's income, generation 3's race and level of education based on our new cross validation training set. Then the OOS deviance for both the full and sub model are calculated and stored.We repeat this process four times with a loop so that each different fold is used as the test set in order to get a sample of four estimates of OOS deviance. We are then able to average the four OOS estimates for both models to compare whether the full or sub model has a better OOS error.

In *section 3*,we use sample splitting with a fixed split of 70-30. Because IS deviance and R2 are prone to overfitting we need to find the OOS deviance and R2. We split the data set into two parts, one estimating the model with 70% of the data and the other 30% as the test to compute the OOS deviance and R squared. Because our Y variable is continuous we are going to use an OLS deviance in *section 4* to make predictions. We run a prediction with the sample split training set and find its IS and OOS R^2.

However, in practice nobody uses the simple form of sample splitting because there a problem with sample splitting is that different random splits might give very different OOS deviance estimates. Cross validation fixes this by averaging over more folds to reduce the variability of our OOS deviance. So when we run our lasso regression in *section 5* with the sample split training set, we are going to cross validate it into four different folds using the cv.glmnet command for both our lasso and ridge regressions. After creating these regressions we manually calculate their IS and OOS R^2 with the OLS deviance function.

For our next step we needed to further narrow down the set of possible regression models. This is done by adding a penalty called lasso, then one called ridge. The lasso estimator helped us narrow down which regressors to use. We started with the most complex model,

including all candidate regressors, and chose which we wanted to move forward with and interact. The other penalty we looked at was ridge, which will instead shrink the regressors but never zeroes them out. Because we want to expand the plausibility of our research question using our limited data set, we needed to test interactions that would make our model more complex. It is especially important to add the lasso penalty here because it will choose the most relevant regressors and also reduce the complexity of the estimated model.

We ran different complex models through lasso, trying interactions such as including both the second generation and first generations income. We did this because it might be possible that the effect of parental income depends on both the first and second generations, meaning that the interaction term would be able to capture a mutual effect. Similarly we wanted to add an interaction term between all three generations' education levels because it will help in measuring the effect of intergenerational education, affluence, access to resources, etc. that a family might have.

We have been using OLS deviance to find the R^2 of all of our models because our logged Y variable is continuous. However there is another way to quantify a model's predictive performance without using the OLS deviance function which is *section 6* of the code. With this we use lasso and ridge to predict log income while choosing the optimal penalty with a four fold cross validation. We use our original data set instead of a manually created training set. We then use these regressions to find their OOS R^2 with a CV error function by calculating how much better the model performs with the best lambda value chosen through Lasso regularization. It compares the CVM with the best lambda to the initial model without regularization.

We were still unhappy with the OOS R^2 we were getting after using cross-validation on the sample split training set. So in *section 7* we decided to use our manually cross validated

training set that helped us select the model earlier and plug it into our cv.glmnet function. We proceed to follow the exact same steps as before and run both lasso and ridge and calculate their IS and OOS R^2 but using the cross validated train set instead of the 70-30 sample split. In *section 8* we use three graphs to compare the regression from *section 7*. We also max the betas of the regression so we can find out some key information about the model we are testing.

Now that we have selected the better model, decided which training set to use, seen whether Lasso, Ridge, or an Interaction regression works best. We can look at predictions for certain individuals. In *section 9* we select three individuals and use the best model we selected to make a prediction for them. Then we compare that outcome with their actual generation 3 income from the dataset.

For a good predictive model, our sample needs to be representative of our population. The PSID sample is a form of area probability sampling from the University of Michigan meant to be representative of the US population, however their sample of 1365 observations is too small to be representative of the whole United States and often overfitting/bias-variance tradeoff occurs even if this was representative, which leaves our model biased. There are a few potential assumptions we violate like over-fitting. Even though cross validation may help this assumption be valid by randomly splitting the data into sections and averaging out our estimates; our very poor IS R^2 for almost every model points to the data being most likely overfit. Meaning the IS model could be fitting the noise in the training data rather than the relationship we want to predict. We also can't be sure that we don't have an overly simple model that fails to capture important relationships because of our limited regressors. Another assumption we could violate is that our regressors are a good choice to predict our outcome. There are many omitted variables that are involved with our regressors and outcome that are not included in the data frame that can

bias our predictions. OLS assumptions, by using the OLS deviance we're assuming that the relationship between our regressors and our outcome is linear, which ends up giving us poor models most of the time because that is not the true relationship of our data. We assume that assumptions of normality of residuals and homoscedasticity hold. For our models to be valid we need to assume that the predictors used in the regressions are not highly correlated. We know there could be a high correlation between our predictors, so we added lasso and ridge regularization to remove redundant predictors. Yet this might not fully address potential multicollinearity, leading to possible unreliable coefficient estimates. Finally our predictive accuracy is off: we assume at the end that we are able to predict the real world values for individuals using our best model however potential biases or violated assumptions have contributed to our decreased accuracy.

Main Results:

Now to discuss the results of all the methods we performed. Starting with using our cross validation training set in *section 2* to help us choose between the full model and a sub. The results across each of the folds are quite different for the random splits so we average across the folds to receive an OOS error of 19.361 for the full model and 13.366 for the sub model. So because the submodel has a better OOS error we will continue to use it for the rest of the regressions. Next in *section 3 and 4* the 70-30 training test split, as we discussed earlier cross validation is a much more popular and effective method than sample splitting which is proven here as our fixed split provides us with an IS R^2 of 0.225 and an OOS R^2 of 0.231 which would mean that our model is very weak at explaining the variation between our predicted and actual values for generation 3's income.

Now that we've proved that our sample splitting doesn't work well for our model we can move forward with cross-validation and manually compute their IS and OOS R^2 using the OLS deviance function from earlier. In *section 5*, for lasso we got an IS R^2 of -62.945 and an OOS R^2 of 0.647. For our ridge regression we got an IS R^2 of -65.338 and an OOS R^2 of 0.790 and finally the interaction lasso model has an IS R^2 of -4.36e+25 and an OOS R^2 of 0.540. From this set it appears that the ridge regression model is the best for predicting on new data.

In *section 6* we want to see the results if we don't use the OLS deviance functions. With our CV error function we also get OOS R^2 that are a lot lower than *section 5* and are very similar to the results from the 70-30 split. We get an OOS R^2 of 0.214 for lasso and 0.222 for ridge.

In *section 7* where we manually cross validated the training set, we get far better OOS values for every regression. With lasso we got an IS R^2 of -20.254 and an OOS R^2 of 0.895. For our ridge regression we got an IS R^2 of -12.403 and an OOS R^2 of 0.888 and finally the interaction lasso model has an IS R^2 of -452.907 and an OOS R^2 of 0.916. All of the models got better at predicting out of sample with the manually cross validated training set.

As we can see in every regression model the IS R^2 shows the model fits the IS data so poorly we would have been better off using the null model of a straight line through the data instead. IS deviance is a poor measure of predictive ability anyways because the models the minimize it tend to overfit data. Even though our models performed poorly on the training set data it did very well on capturing the effect of new unseen data. The OOS R^2 of all models in *section 7* suggest the model may be too complex for the training data but still is able to do well with new data. We have mentioned that a limitation of our data set is the simplicity of the regressors, but by testing a complex model with interactions, we hope it will lead to a stronger

predictive model. In *section 7* the interactions ended up increasing the OOS R^2 to an impressive 0.916, the best throughout our whole data. Complex models do not always give better results, they can be prone to overfitting, meaning they become too sensitive to every individual movement in a model and miss the overall pattern we want with a machine learning prediction model. Luckily we are happy with our OOS R^2 and choice to include these interactions.

In *section 8* we create multiple graphs (graph 2 at the bottom). Here we can compare the different regressions. On the left side of these plots the penalty is low so the coefficients are different non-zero numbers and as you go to the right of the plot the penalty increases and leads to smaller coefficients until they all reach zero when the penalty is the largest. The higher lambda will zero out more betas which makes the model simple which results in a higher bias. When the lambda is smaller the model is more complex because of more regressors and will have higher variance. These help visualize how our lambda is chosen.

Also in *section 8*, the coefficients of our final predictive model gave us a good understanding of the best regressors in our model. Our most predictive regressor was the interaction term between the third generation having less than a high school education and the second generation having some college education. Its predictive effect on the third generation's income decreases it by 48%. The next strongest coefficient was that if the third generation individual has a college education, their income increases by 34%, which is good proof of the educational opportunity we want to bring awareness to. A one percentage point increase in g2 log income is associated with a .34% increase in g3 log income, but we definitely expected there to be a stronger correlation. The interaction term between generation 1 and 2's income shows that for a one percentage point increase in it, there is a 0.006% increase in generation 3's income. We were hoping for a stronger prediction from this interaction. Similar to the initial basic OLS

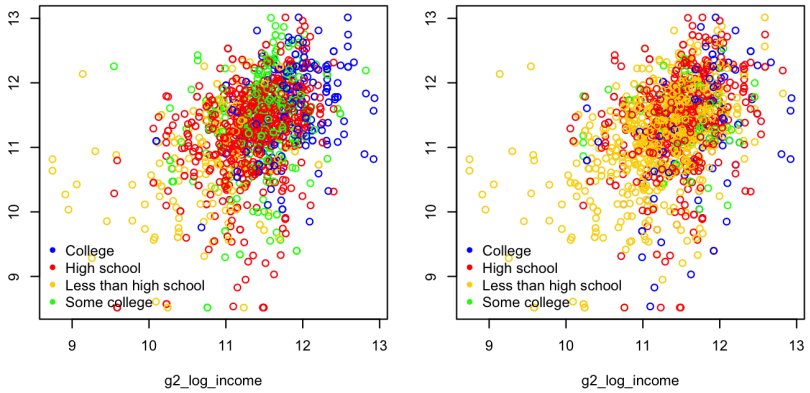
regression we run in *section 1*, these coefficients show upward trends for having higher education levels, except some outliers like the first generation having a college education.

Finally in *section 9*, we can forecast what the income of someone from generation 3 should be and compare it to their actual income. We picked three random individuals to compare their logged incomes. Person 1 has a predicted income of 10.914 and actual of 10.742, person 696 has a predicted income of 10.951 and actual of 11.094, and person 800 has a predicted income of 11.990 and actual of 11.329.

What all of these different methods proves is the most important part of machine learning is having good data. Even though we picked the better training set, used the model with a lower OOS error, created a model that while it did poorly on our training set was good at predicting out of sample we were only able to predict close to their actual incomes. All of these methods don't make our regressors more informative than they are. We tried to select the best model out of potential worse models. One cannot expect to perfectly predict an individual's income based on such little provided regressors.

Link to data:
https://www.openicpsr.org/openicpsr/project/185941/version/V2/view?path=/openicpsr/185941/fcr:versions/V2/for_students.zip&type=file

Graph #1



Graph #2