Project Step 4: Beyond the Linear Model

Keon Dibley & Lucy Zhao

2023-12-15

Dataset: California Housing Prices Data Source: Kaggle (https://www.kaggle.com/datasets/fedesoriano/california-housing-prices-data-extra-features)

(We randomly select 200 observations as train data and 50 observations as test data.)?

Our data set includes California Housing data, with some relevant variables being **Median_Income**, **Population**, and **Distance_to_LA**. We are using **Median_House_Value** as a response variable and other variables as predictors. Each observation in our data set is a block of California Houses,

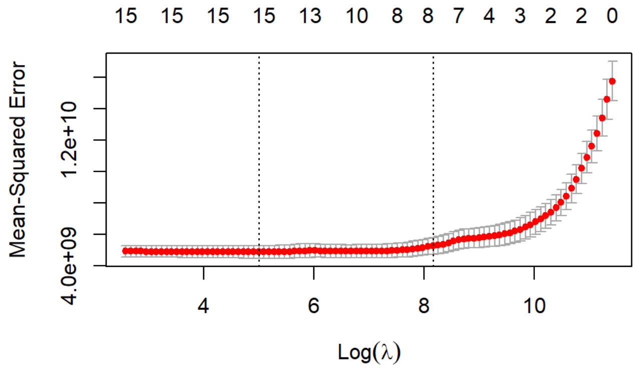
Our data set has **1 response variable**, **14 quantitative variables**(we create "Min_Distance" which equals to the min among distances to the 4 cities) and **2 categorical variables**(which we created). Each observation represents a block of California houses and has attributes like the median house value of that block, its median income, and the population of the block. From this dataset, we chose **Median_House_Value** as the response variable and used other variables as predictors to predict Median_House_Value.

Set seed for reproducibility, get training/testing data:

Ridge and Lasso Regression:

Lasso:

Using Cross Validation, I found λ to be 148.2306. In the graph below, this value corresponds roughly to the value of $Log(\lambda)$ that minimizes MSE.



Final MLR Model from Step 3:

$$\ln(\text{Median}\hat{\text{H}}\text{ouseValue}) = -14.894 + 4.663 * \text{MedianIncome} - 1.491 * \text{MedianIncome}^2 - 0.125 * \ln(\text{MinDistance}) - 0.162 * \ln(\text{Distancetoco}$$

Lasso Regression Model:

$$\text{MedianHouseValue} = 215140 + 74617 * \text{MedianIncome} + 21995 * \text{Median\_Age} + 9356 * \text{Tot\_Rooms} - 8664 * \text{Tot\_Bedrooms} - 71999 * \text{Po}$$

When we changed the value of λ, we found that Lasso Regression shrinks most of the coefficients as λ increases, punishing the relevant ones less, and even increasing the value of the most important coefficients, Median_Income and Distance_to_coast.

When comparing Lasso Regression to our MLR model, we found that our model uses many fewer variables, and much smaller coefficients. This is interesting because we thought that Lasso Regression would impact our less relevant variables more, but this is very dependent on the value of λ.

Ridge:

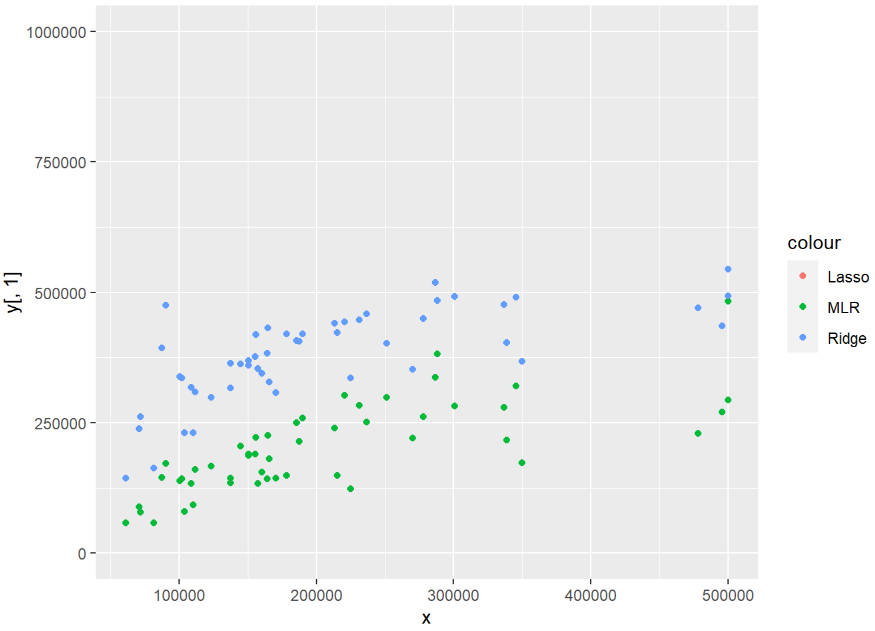
RR Model:

$$\text{MedianHouseValue} = 0 + 36046 * \text{MedianIncome} + 1848 * \text{Median\_Age} + 8.08 * \text{Tot\_Rooms} - 49 * \text{Tot\_Bedrooms} - 78 * \text{Population} + \cdots$$

In Ridge Regression, λ = 0 causes the GCV to be lowest. This is saying that the Ridge Regression doesn't reduce our coefficients at all from OLS. In this data set, there are many more samples than predictors, which validates the use of a more complex model.

Compared to our MLR model, the Ridge Regression model uses more predictors, which is typical of Ridge Regression. Some of its coefficients are still large, but most are much smaller than Lasso Regression's corresponding coefficients.

Graph of observed response vs predicted response



Lasso Regression's predictions are all so high, they cannot be shown on the graph. This is likely due to the high lambda value chosen through cross validation, which shrinks the variance, but creates a very high bias.

Both our MLR and Ridge models have low variances in prediction, but our MLR model is much more unbiased, with the observed response values matching up fairly well with the predicted response.

Conclusion

Overall, we executed Lasso and Ridge Regression to try to shrink the variance of our model, while keeping it as unbiased as possible. We found that Ridge Regression was a much more effective method for our data set, but our MLR model from Step 3 proved to be the best at predicting accurately without large variability. We were surprised by the ineffectiveness of Lasso Regression, and we hope to find more ways to decrease the variance of our model through future investigation.

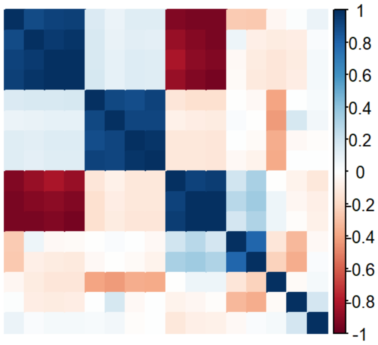
Innovation: Principal Component Regression

Principal Component Regression (PCR) is used to reduce the number of predictors in a model, creating Principal Components that act as regressors, but are linear combinations of the original predictors.

We thought this technique would be suitable for our dataset because we have many predictors that are heavily related to each other, such as Distance_to_LA, Distance_to_SanFrancisco, Distance_to_SanJose, and Distance_to_SanDiego. Another goal of PCR is to get rid of colinearities, which we think exist in our data due to correlatedness of some of the variables we just mentioned.

We thought PCR could help us see how our predictors interact with one another in our data, and that this could help us explain our response variable, Median_House_Value.

First, we examined the correlation between our predictors through a correlation plot:



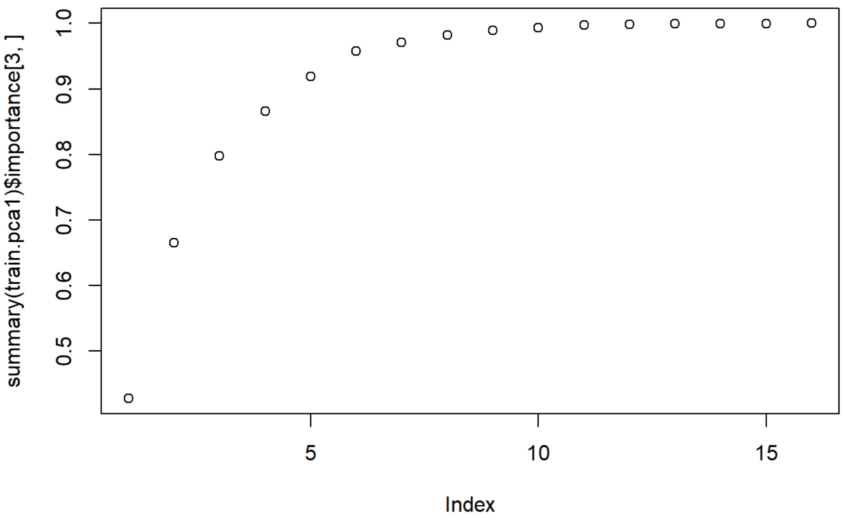
Here, we see significant correlation between some of our variables, with the darker blue and red areas that aren't on the diagonal representing higher correlations.

Next, we normalize the data, because Principal Component Analysis (PCA) is sensitive to data that hasn't been centered.

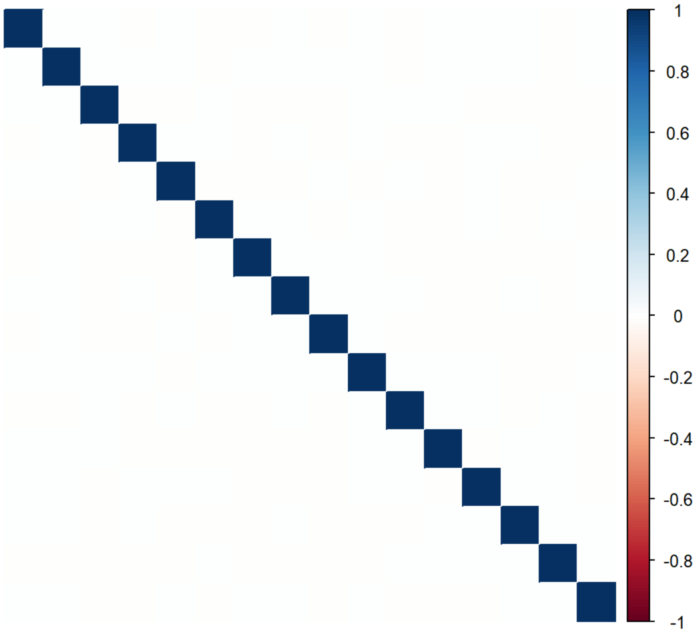
PCA (Principal Component Analysis)

We performed PCA on our data set and found 16 different components for our model. Then, we want to reduce this reduce the number of components while still maintaining as much of the original variance as possible.

Say I want to maintain at least 90% of the original variance. According to the below graph, and a summary we created in R, we then should reduce our components down from 16 to >= 6. This suggests that there is a good amount of scope to reduce the dimensionality of our model.

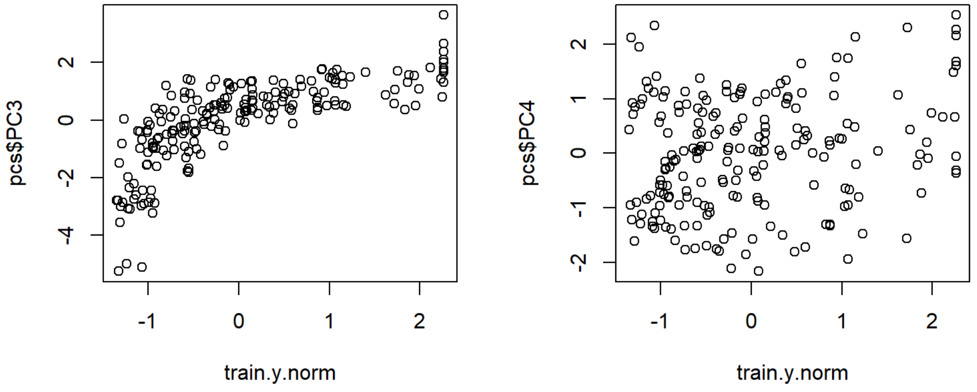


Another important point about the components is that they are always orthogonal, so there is no correlation whatsoever between them. This can be seen in the correlation plot below, where the only correlation seen is actually autocorrelation.



Regression with PCs

Below are a couple of scatterplots showing the association between the PCs and Median_House_Value. PC3 has a high correlation with Median_House_Value, while PC4 doesn’t.

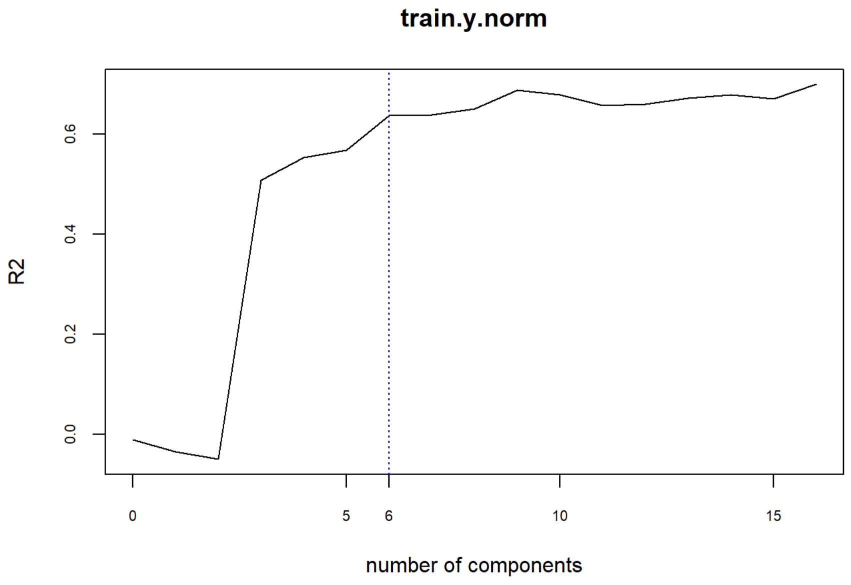


We then fitted three different linear models to Median_House_Value and our Principal Components. One was a full model, which included all 16 components. Our second model maximized adjusted R^2 and used 11 components (PC2, PC3, PC5, PC6, PC7, PC8, PC9, PC10, PC13, PC15, and PC16). Our third model used the first 6 components (PC1 - PC6), because we can capture 90% of the original variance with only these 6 predictors.

Model 1: 16 predictors, R^2 = 0.7495, adj R^2 = 0.7276
Model 2: 11 predictors, R^2 = 0.7475, adj R^2 = 0.7327
Model 3: 6 predictors, R^2 = 0.5923, adj R^2 = 0.5796

Out of these three models, we found that model 2, which maximized adjusted R^2, was the best model, because it gave the highest adj R^2 value while also having 5 fewer predictors than the full model.

To make sure we were choosing an appropriate number of PCs, we examined a graph which used cross validation to plot the increase in R^2 as we increase the number of components in our model.



In examining this graph, we found that the most dramatic change in R^2 occurred when we used 6 PCs. When we examined a model with 6 PCs, we found that we still preferred our model with 11 PCs due to its higher adj R^2, but both both models are valuable, depending if you want a model with fewer predictors or higher R^2.

This new method was very interesting to try out, and I'm pleased to say that we found a model that is comparable in accuracy and variability to our step 3 MLR model.