Predictive Modeling Process

Joseph Simonian & Ellen Hwang

Oct 14, 2016

Abstract

This report will be examining the applications of ridge regression (RR), lasso regression (LR), principal components regression (PCR), and partial least squares regression (PLSR). We will also be exploring how cross validation methods apply to each of these models and how cross validation strengthens our models.

Our project will be using the Credit dataset, taken from the webpage for the text, *An Introduction to Statistical Learning*, to perform this simple linear regression. We analyze credit card debt based on a number of factors, such as income, gender, and credit limit, and create predictive models for credit card debt through a number of linear regression models. This report will include a description of the data, methods, analysis, and results for the regression models created. # Introduction

In this project, we will be exploring 4 types of regression models that strengthen model interpretability and prediction accuracy.

Two of the regression models, ridge regression and lasso regression, are shrinkage methods which constrains the coefficient estimates to shrink the coefficient estimates towards zero. This helps reduce coefficient estimates reduce their variance. They differ in the application of shrinkage - lasso implements parameter shrinkage and variable selection, whereas ridge merely modifies parameter weights.

The other two regression methods, principal components analysis and partial least squares, are dimension reduction methods. Dimension reduction methods essentially reduce the number of predictors under consideration to strengthen a model. They differ in that PLS works to maximize inter-class variance in its low-dimensional model, whereas PCA maximizes the variance of features themselves.

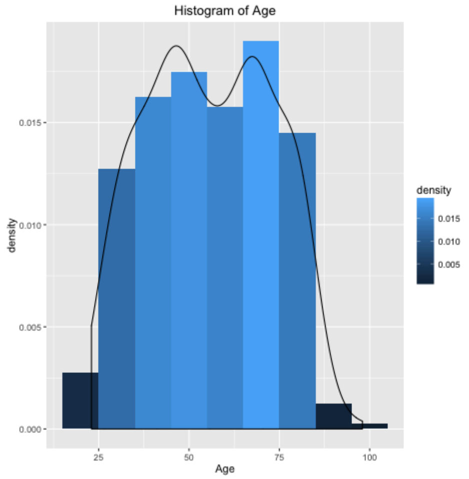
We will be applying these methods to the Credit Dataset to create various models predicting a person's Balance from various independent variables. We will discuss our methods in R, our various results, and conclusions about the relationships and statistics we find.

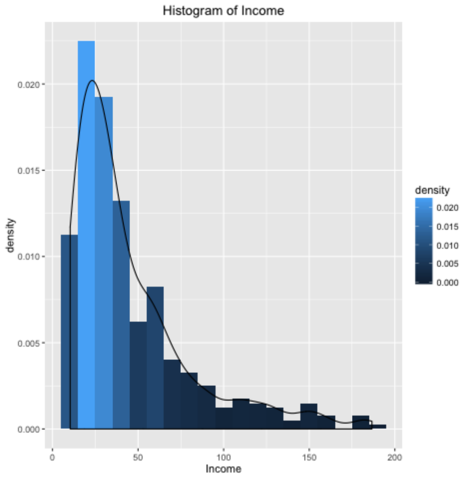
Data

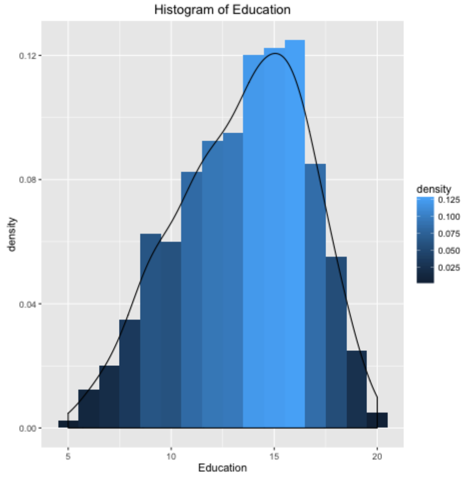
For this model, we use the Credit.csv dataset available from the Introduction to Statistical Learning website (download). This dataset is analyzed in Chapter 6 of the ISL book.

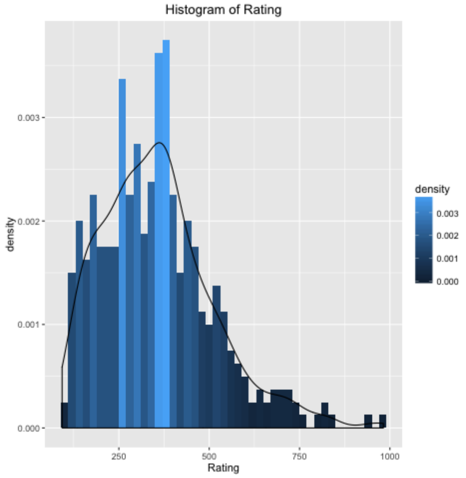
This dataset contains "Balance", representing average credit card debt across a group of people, and 10 explanatory variables, of which 6 are quantitative and 4 categorical. The quantitative variables are: Income, Credit Limit, Credit Rating, No. of Cards, Age and Years of Education. The categorical variables are: Gender, Student Status, Marital Status and Ethnicity.

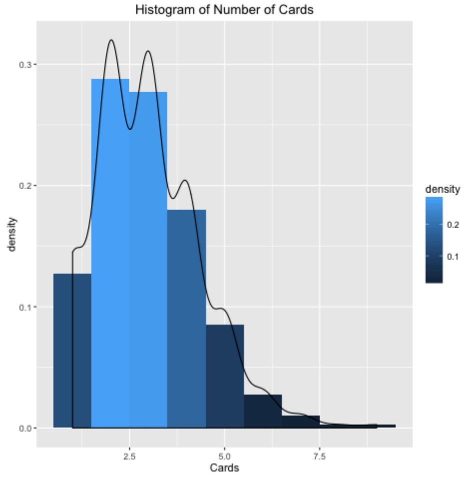
We include basic exploratory data analysis below:

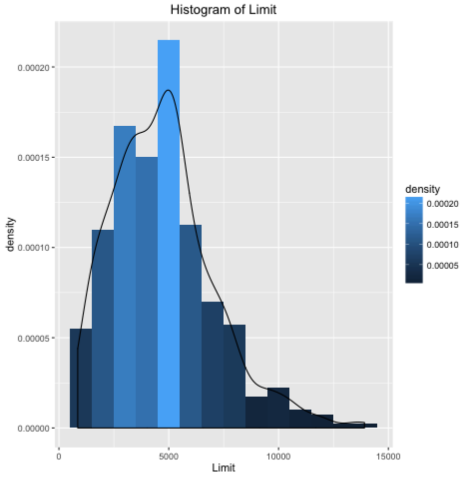

Histogram of Age
density
0.000
0.005
0.010
0.015
25
50
75
100
Age
density
0.015
0.010
0.005

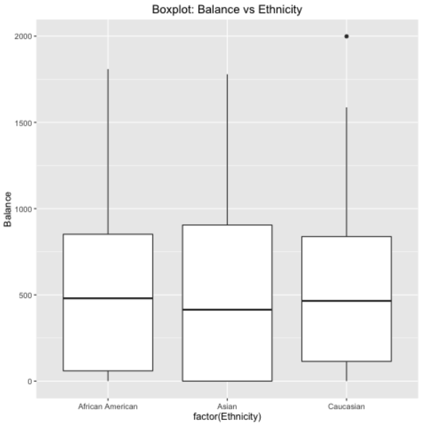

Histogram of Income
density
0.000
0.005
0.010
0.015
0.020
0
50
100
150
200
Income
density
0.020
0.015
0.010
0.005
0.000

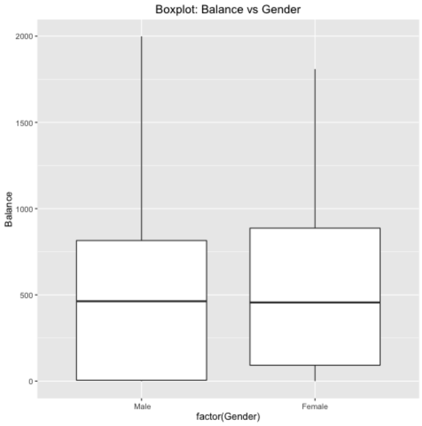

Histogram of Education
density
0.00
0.04
0.08
0.12
5
10
15
20
Education
density
0.125
0.100
0.075
0.050
0.025

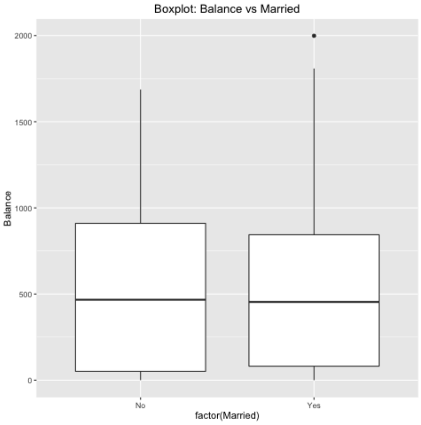

Histogram of Rating
density
0.000
0.001
0.002
0.003
250
500
750
1000
Rating
density
0.003
0.002
0.001
0.000

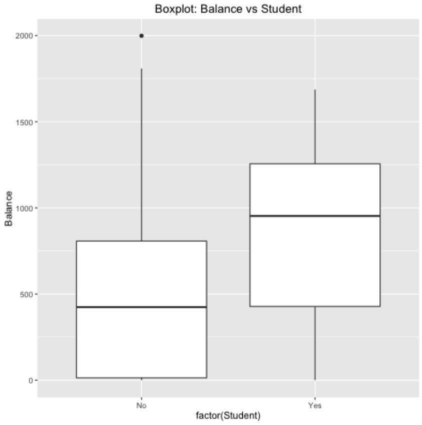

Histogram of Number of Cards
density
0.0
0.1
0.2
0.3
2.5
5.0
7.5
Cards
density
0.2
0.1


Histogram of Limit
density
0.00000
0.00005
0.00010
0.00015
0.00020
0
5000
10000
15000
Limit
density
0.00020
0.00015
0.00010
0.00005

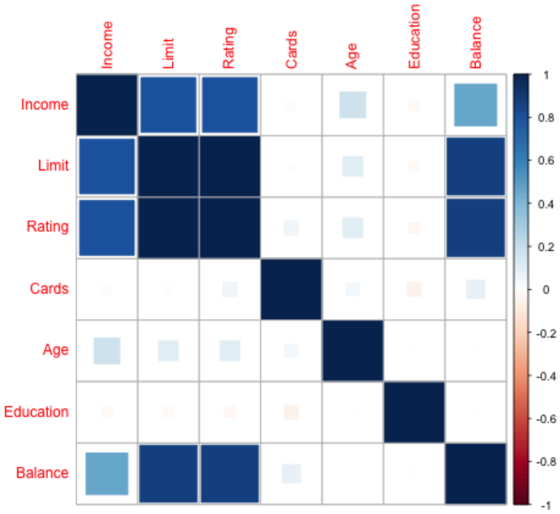


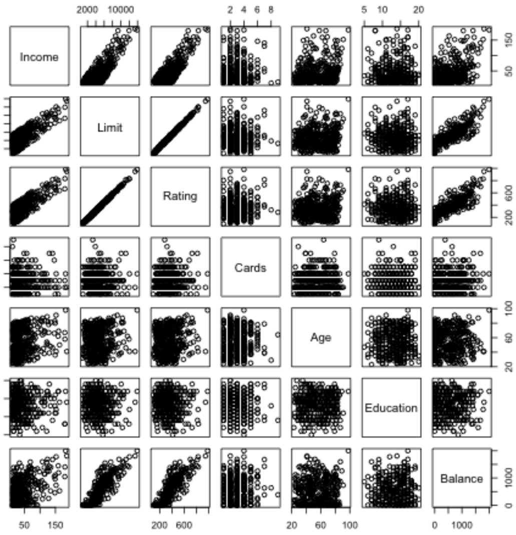






We also include correlation plots and scatterplots between quantitative variables:





Methods

Preprocessing

We use one-hot encoding to dummy out categorical variables, and perform mean-centering and standardization of all variables, setting their means to 0 and standard deviations to 1 to give us comparable scales.

We then split the data into a training set (75% of total data) and a testing set (25% of total data) for the purposes of tuning hyperparameters (such as λ in lasso and ridge regressions).

Lasso Regression

With lasso regression, we first create a preliminary model using `glmnet()` on the training dataset, with $\alpha = 1$ to represent the elasticnet mixing penalty used in lasso regression. We then cross validate our model on our training set using the function `cv.glmnet()`. During cross-validation, we perform a parameter search for lambda in the range $(10^{-20}, 10^{10})$, which was an expansion on the given grid from the project spec. Lastly, we fit a model to the full dataset and predict credit card balances using our best lambda found via CV.

Partial Least Squares Regression

With PLS regression, we use the `plsr()` function modeling *Balance* as a function of all other variables. To find the best number of components, we perform cross-validation and find the minimum of the validation PRESS features. Ultimately, we perform plsr regression on the full dataset using the best number of components found via cross-validation.

Ridge Regression

As with lasso regression, we create an initial model using `glmnet()` on the training dataset, with $\alpha = 0$ to represent the elasticnet mixing penalty used in ridge regression. We then cross validate our model on our training set using the function `cv.glmnet()`. Once again, we perform a parameter search for lambda using the extended grid $(10^{-20}, 10^{10})$. Finally, we fit a model to the full dataset and predict credit card balances using our best lambda found via CV.

Principal Components Regression

Similar to with PLS regression, in PCR we `pcr()` function from the `pls` package to implement our regression. Again, we perform 10-fold cross-validation to determine the optimal number of components, and then make predictions of the full dataset based on that number of components.

Analysis

```r
load('../data/lasso-objects.RData')
load('../data/pls-objects.RData')
```

Lasso Regression

When we perform cross validation and compute the associated test error we find that the MSE is quite low at $7.0180403 \times 10^{-27}$. There are also quite a few coefficients that are zero.

```
lasso.coef
```

```
##    (Intercept)         Income          Limit         Rating          Cards
##    0.000000000   -0.598053709    0.907149087    0.433668739    0.050668895
##            Age      Education   GenderFemale     StudentYes     MarriedYes
##   -0.023089257   -0.007207687   -0.011596315    0.277829118   -0.009427733
## EthnicityAsian
##    0.016202618
```

So all in all, we find that exactly 7 predictors are relevant in the regression.

```
lasso.coef.official
```

```
##         Income          Limit         Rating          Cards            Age
##   -0.598053709    0.907149087    0.433668739    0.050668895   -0.023089257
##      Education   GenderFemale     StudentYes     MarriedYes EthnicityAsian
##   -0.007207687   -0.011596315    0.277829118   -0.009427733    0.016202618
```

Partial Least Squares Regression

Looking at the pls output summary, we see that the cross-validation error occurs when M = 57.1004389 are used. The corresponding test set MSE = 0.1691708. Finally, we perform PLS using the full data set using M = 57.1004389 which is the number found in cross-validation. Looking at the summary of this model:

```
summary(pls.fit.full)
```

```
##                 Length Class      Mode
## coefficients      36   -none-     numeric
## scores          1200   scores     numeric
## loadings          36   loadings   numeric
## loading.weights   36   loadings   numeric
## Yscores         1200   scores     numeric
## Yloadings          3   loadings   numeric
## projection        36   -none-     numeric
## Xmeans            12   -none-     numeric
## Ymeans             1   -none-     numeric
## fitted.values   1200   -none-     numeric
## residuals       1200   -none-     numeric
## Xvar               3   -none-     numeric
## Xtotvar            1   -none-     numeric
## fit.time           1   -none-     numeric
## ncomp              1   -none-     numeric
## method             1   -none-     character
## call               4   -none-     call
## terms              3   terms      call
## model             13   data.frame list
```

We see that the 57.1004389 component PLS fit explains 86.63% of the variance.