Predicting bad loans: Ridge over OLS

Robert Kaplan and Edbert Jao

December 22, 2021

Abstract

We estimate the likelihood of default in peer-to-peer personal loans. Given that our data is plagued by multicollinearity and our specification over-fits, we use Ridge regression to obtain better predictions. We contrast Ridge with two alternatives, Lasso and principal components regression; we use a data-driven process to select the necessary hyperparameters; and we demonstrate the sensitivity of our method to the value of that hyperparameter. We also discuss the validity of our results and a possible extension.

1 Introduction

The possibility of default is threatening to lenders and borrowers alike, with significant consequences for both parties. For lenders, the costs of default are expensive, and may likely have secondary effects on its ability to raise capital in the future; for borrowers, default leads to punitive measures which have long-lasting effects. Knowing that default occurs more frequently in some types of debt than others, we seek to estimate the likelihood of default in peer-to-peer personal loans.

We demonstrate that Ridge regression is an appropriate, if not the preferred tool for our problem and its particular constraints. We show that Ridge regression can estimate the likelihood of default at least as well as alternative methods which we consider: Lasso and principal components regression. Ridge allows us to overcome the limitations in the dataset and produce a more reliable predictive model.

The structure of the paper follows. Section 2 elaborates on the problem. Section 3 explains our methodology: our motivation for choosing it in particular and its mathematical underpinnings. Section 4 presents the data and section 5 the results. In section 6 we briefly discuss the results and section 7 concludes.

2 Problem: Predicting default on personal loans

Lenders take many precautions to ensure that they only admit borrowers who will fully repay their debt. If lenders did not believe debt would be repaid, expected returns would be negative and investors would withdraw their funds. Credit would become unavailable or prohibitively costly. Despite admitting only borrowers they believe would repay, default still occurs — sometimes, very frequently. For example, in 2010 the average delinquency rate on single-family mortgages hit an all-time high, at 10.9%. Since then, it has steadily fallen to 2.5% in 2019, approximately the average during the 1990s. [**?**]. Mortgages are only one example of default in spite of extensive pre-conditions

for borrowing.

Our problem is an exercise in supervised learning: using an extensive cross-section of credit-worthy borrowers — some of whom default but most of whom do not — **we estimate the likelihood that a new borrower eventually fails to repay**. Knowing this parameter estimate has practical finance applications for a lender: for example, whether to adjust its interest rate and fees structure, or calculating Value-at-Risk and compliance with regulatory requirements on stress-testing.

3 Methodology

Our approach uses Ridge regression to solve the many-predictor problem in regression with Big Data. In section 3.1, we illustrate why typical methods do not suit our problem. In section 3.2 we introduce Ridge regression, also called l_2 regularization, and explain how it suits our problem better than typical methods. We also introduce other methods against which we may compare Ridge: namely, Lasso and principal components regression. In section 3.3, we walk through the model we use to estimate the likelihood of default.

3.1 Motivation

Regression applications usually prefer ordinary least squares (OLS) regression; when the assumptions of the Gauss-Markov theorem are satisfied, OLS is the best linear unbiased estimator. However, there are three features of our problem which preclude using OLS. Ours is a prediction problem, so we frame each in the context of generalization error.

1. *Model specification.* Polynomial models make very good predictive models, because they capture a great deal of variation in the data without requiring the modeler to possess a detailed understanding of the data generating process. This is desirable: with very many predictors, it may be unclear how to precisely specify the model. However, polynomial models risk over-fitting. This results in excessive generalization error when using OLS.

2. *Multicollinearity.* When some features are highly correlated with one another, the OLS estimator is inefficient. That is, coefficients do not have the least variance possible. As we show in section 2.2, multicollinearity is present in our data. This results in greater variance, which results in greater prediction error.

3. *Large data.* Given many features, it is unlikely that all features are equally important. Rather, it is more likely some features are (relatively) important and others are are not. As the coefficient on every feature contributes to generalization error, we can economize by weighting towards features that matter more (and away from those that matter less).

3.2 Ridge regression

The mechanics of Ridge regression differ from OLS in one key way: l_2 regularization. In either case, we use a model similar to the general form below:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + u_i \tag{1}$$

To highlight the difference between OLS and Ridge, we begin by recalling that the least squares solution to (1) amounts to minimizing:

$$\min \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{1i} - \hat{\beta}_2 x_{2i} - \cdots - \hat{\beta}_k x_{ki})^2$$
$$= \min \|Y - X\hat{\beta}\|$$

Through some familiar matrix algebra, the objective above eventually works out to the normal equation $X^T X\beta = X^T Y$. Given that the matrix $X^T X$ is invertible, we solve the solution by solving for β:

$$\hat{\beta}^{\text{OLS}} = (X^T X)^{-1} X^T Y \tag{2}$$

As established in section 3.1, $\hat{\beta}^{\text{OLS}}$ has undesirable properties in the setting of our problem. We employ Ridge regression instead. It estimates the same model as in (1), but solves for the coefficients differently. With $\lambda \geq 0$, it instead minimizes:

$$\min \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_2 - \cdots - \hat{\beta}_k x_k)^2 + \lambda \sum_{j=1}^{k} \beta_j^2$$
$$= \min (Y - X\hat{\beta})^T (Y - X\hat{\beta}) + \lambda ||\hat{\beta}||_2^2$$

The only difference between the Ridge minimization problem and the OLS minimization problem is the penalty term, made up of the squared l_2 norm and a user-defined parameter λ. As in (2), this works out to the following solution.

$$\hat{\beta}^{\text{Ridge}} = (X^T X + \lambda I)^{-1} X^T Y \tag{3}$$

Ridge regression is useful for our problem because it mitigates all three of the issues listed in section 3.1.

1. *Regularization.* The new penalty term, $\lambda||\hat{\beta}||_2^2$, regularizes the model by penalizes for over-fitting. It does so through "shrinkage," which reduces the magnitude of the coefficients. For polynomial specifications especially, this substantially improves generalization error while preserving the simplicity of the model. Visually, this creates an intuitive smoothing effect, as seen in figure 1. While p(1) under-fits and p(7) over-fits, the "regularization" is smoothed such that if falls somewhere in-between. This increases training error somewhat, but can greatly reduce generalization error.

2. *Bias-variance trade-off.* Ridge mitigates multicollinearity because it is a biased regression technique. This can be seen in 3, as λ adds bias to the coefficients. The consequence is that the Ridge estimator $\hat{\beta}^{\text{Ridge}}$ has greater bias and may have less variance than the OLS estimator. Since the prediction error is the sum of the bias and variance of the coefficients, Ridge can reduces prediction error and mitigate the variance-effect of multicollinearity. However, it renders meaningless point estimates of coefficients on their own.

3. *Shrinkage.* Shrinkage of particular coefficients is useful in itself. As λ grows all coefficients converge to zero, but they converge at different rates. For a given model and optimal λ^*, the model can signal which features are relatively important and which are not.

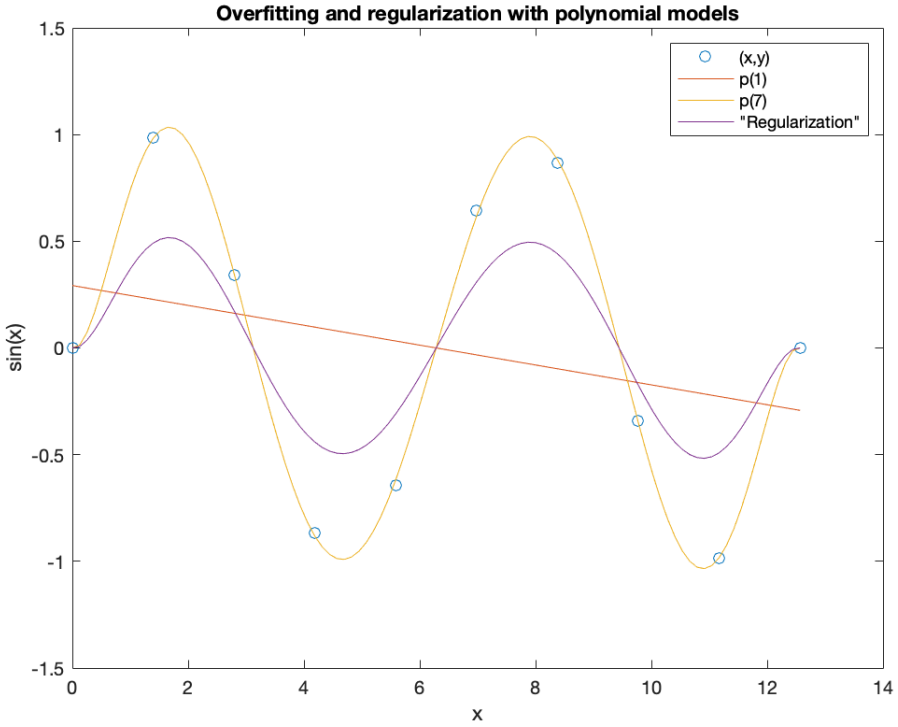

Figure 1: Visualizing weight regularization

Section 5.1 contextualizes the effectiveness of Ridge regression, by presenting the results alongside OLS and two other methods: lasso and principal components regression. We do this to assess how Ridge changes our estimates and to discuss whether Ridge provides the best estimates for our problem. What follows is a very brief introduction of these two other methods.

Lasso regression, also called l_1 regularization, is a sibling of Ridge regression, and functions nearly the same way. The Lasso coefficients are found by minimizing the following objective:

$$\min\ (Y - X\hat{\beta})^T(Y - X\hat{\beta}) + \lambda||\hat{\beta}||_1 \tag{4}$$

Whereas Ridge uses the l_2 norm in the penalty term, Lasso uses the l_1 norm. Lasso has all of the same properties as Ridge which are discussed above, with one additional note on shrinkage: Lasso regression collapses some coefficients to zero immediately (and the rest converge to zero). This can be interpreted as sparsity in the β vector. This can be useful for model selection, or inferred model selection when comparing against Ridge.

Principal components regression (PCR) works differently. Briefly put, PCR uses OLS. It regresses the dependent variable (with respect to the original basis) on the projections of the data matrix onto the principal components. Given some familiarity with principal components analysis, it immediately follows that this has one major advantage as a regression technique: dimensionality reduction. PCR uses the orthogonality of the principal components to mitigate the same issues that Ridge and Lasso do. Since the principal components are orthogonal, they are perfectly independent of one another. Since PCR uses the latent representations, it employs far fewer regressors than

Ridge would need to explain the same amount of variance; therefore, the OLS estimator will have sufficiently low variance. The major disadvantage of PCR is that there the regressors are transformed completely independent of the dependent variable; for that reason, it is unclear if the projections will be particularly effective at out-of-sample prediction. Similarly, the interpretative meaning of the coefficients is not obvious, since they relate the projections of the data to the dependent variable, not the data themselves.

3.3 Our model

To predict the likelihood of loan default, we employ a second-order polynomial linear probability model (LPM).

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_{k+1} x_{1i}^2 + \beta_{k+2} x_{2i}^2 + \cdots + \beta_{2k+1} x_{1i} x_{2i} + \beta_{2k+2} x_{1i} x_{3i} + \cdots + u_i$$

The dependent variable is an indicator variable, which takes 1 if a loan i enters default and zero otherwise. The main predictors $\mathbf{x}_k$, which are features of each loan and their respective borrowers, are described in section 4. The second-order model means that we include the main predictors, the squares of the main predictors, and their interactions. In section 5, we estimate this model using Ridge regression and the alternative methods discussed above.

There are two important qualifications to make about our model. The first is that, in all specifications (OLS, Ridge, Lasso, PCR), the regressors have been standardized. That is, each raw feature x_n^r has been rewritten as the standardized feature x_n, where $x_n = \frac{x_n - \bar{x}}{s_n}$. $\bar{x}$ is the sample mean of that feature and s_n is the sample standard deviation of that feature. Standardization is essential for two reasons: correct shrinkage and cross-comparisons. Without standardization, Ridge or Lasso would heavily weight large magnitudes, even if those coefficient have little variance. Similarly, standard deviation units allow the coefficients of different models to be compared to one another, since the degree of the bias is unstable and sensitive to λ.

The second qualification is that the LPM is limited. LPMs are inefficient as they can predict probabilities outside of $[0, 1]$. Predicted values (as well as any coefficients) with a magnitude greater than 1 are invalid, from a probability point-of-view. They also cannot have normally distributed standard errors, so the statistical significance on the coefficients may be wrong. However, the coefficients do approximately capture the marginal linear effects of the features on the probability of the outcome variable. Statistical significance is less relevant for our problem prediction, since we are using biased regression techniques that obscure causal interpretations of each coefficient. Finally, we discuss in section 6 that all methods predict relatively few values outside of $[0, 1]$; furthermore, we show that such values are biased in such a way that does not invalidate possible classification methods.

4 Data

We use publicly available loan and borrower data from LendingClub, an early online peer-to-peer lending platform. LendingClub facilitated peer-to-peer lending from its founding in 2006 until 2020, when it concluded the program to restructure itself as a digital bank.

The data were generated as a result of the credit application and loan assignment process. First, prospective borrowers completed an application to LendingClub — it included typical credit-screening questions as well as questions regarding a borrower's interests, hobbies, and affiliations.

Accepted candidates were then assigned a loan grade which determined the interest rate and fees owed, along with an approved loan size. The better is the grade, the lower are the interest rate and fees owed to LendingClub.

What followed was the peer-to-peer component. Within the LendingClub social platform, accepted candidates posted listings for their loan. Prospective investors could then lend part or all of any loan they chose, on which they earned interest and could trade the loans on a secondary market.

Our data exist because LendingClub took advantage of its unique status in a yet-unregulated industry. It crowd-sourced data analysis by publicly sharing its anonymized borrower data. These data begin with loans (and corresponding borrowers) issued in 2007 and conclude in the 4^{th} quarter of 2018. Shortly thereafter, LendingClub removed the public data as it prepared to end the peer-to-peer platform and and restructure as a digital bank.[1] The dataset has 1,347,941 complete cross-sectional observations which reflect the status of the loan as of Q4 2018.

For several reasons, we do not use the complete dataset. The first is that it is neither necessary nor relevant, since we emphasize the usefulness of Ridge when there are many predictors relative to the sample size. It would also increase the intensiveness of computations. For this reason, we draw a random subsample of 12,000 observations. This subsample, which is the basis of our analysis, is presented below

The more relevant reason is class imbalance. Machine learning algorithms, as they seek to minimize error, predict incorrectly when there are very few of a particular class. The algorithm may avoid predicting the small class entirely, which is not useful. This is the case for our data. Therefore, we perform random under-sampling; this method randomly deletes observations in the majority class until the desired proportion is reached. Table 1 shows that our original sample of $n = 1,347,941$ has very few defaulters relative to non-defaulters. About 8.2% loans in the sample enter default, which will hinder our predictive regression techniques. Therefore, we randomly delete many of the non-defaulters until parity is reached between the two characters. Tables 8 and 9 show the means and standard deviations on a select few standardized features of the non-defaulters in both samples. The difference between them is not statistically significantly different from zero.

	Non-defaulters	Defaulters
Unbalanced	1,231,732	116,209
Balanced	116,209	116,209

Table 1: Class imbalance

Table 11 presents summary statistics on the randomly drawn subsample ($n = 12,000$) which we use in our analysis, and table 10 on the re-sampled data with parity between classes. Again, in the standardized data the differences between them are negligible.

We select 32 features to include in this sample. These form the main 32 predictors included in our model specification. These are selected using intuition about consumer finance.Table 12 presents summary statistics of these standardized features. Some of these features describe the loan at origination (e.g., the principal amount, assigned interest rate, etc.), the borrower at origination

[1]For this reason we acquired a version of the data from a Regis University professor's archive on Kaggle.com, here.

(their annual income as reported, debt-to-income ratio, revolving utilization, FICO credit score interval, etc.), and other features describe the status of the loan and borrower later on (e.g., missed payments in the previous two years, how much of the principal and interest have been repaid, etc.). Some of these features are highly correlated, as seen in figure 2, indicating the presence of multicollinearity. As discussed earlier, this is evidence which supports using Ridge regression to obtain better predictions.

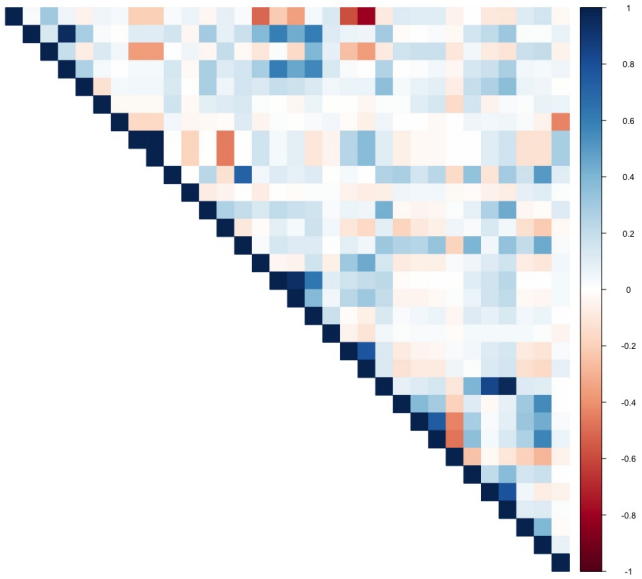

Figure 2: Correlation plot of main predictors

5 Results

Using Ridge regression to predict loan default provides a significant improvement over OLS, and performs at least as well as do competitive alternatives. In 3.1 we present these main results. We also recognize that theory does not tell us λ. In 3.2, we demonstrate a data-driven, error-minimizing method for selecting λ, through our own implement k-fold cross-validation. We also show that Ridge results are highly sensitive to λ selection.

5.1 Ridge does better than the alternatives

We run two versions of our second-order model: with and without the linear interaction terms. We then estimate that model using OLS, Ridge, Lasso, and PCR. Table 2 presents some key statistics about our results.

There are a few key trends which can be observed from the table. First is that the in-sample MSPE is usually greater than the out-of-sample MSPE. This is expected for the cross-validated methods, since the parameter-selection relies on minimizing the latter. Another is that OLS has a far greater in-sample MSPE, and the greatest share of "invalid" predicted values. This confirms that OLS is

Table 2: Regression results

	OLS	Ridge	Lasso	PCR	OLS w/ int.	Ridge w/ int.	Lasso w/ int.
λ / PCs	-	0.015	0.015	63	-	0.015	0.001
In-sample MSPE	77.844	0.0002	0.001	0	3.001	0.021	0.020
Out-of-sample MSPE	0	0.0002	0.001	0	0.668	0.023	0.021
No. nonzero coefficients	NA	0	58	NA	NA	0	262
% of $\hat{y} \notin [0, 1]$	0.625	0.288	0.022	0.510	0.463	0.374	0.461

the least useful choice of regression method. We also see that the validity of the predicted values improves in OLS when interaction terms are includes, but worsens for Ridge and Lasso. This suggests that we prefer the specifications without interaction terms, as those are likely over-fit. Given this preference, we see that Ridge, Lasso, and PCR perform very comparably in terms of error. However, Ridge and Lasso perform far better than PCR in terms of predicted value validity. While the share of invalid predicted values seems much higher in Ridge, most of those values are only slightly greater than 1 or less than zero. In Lasso, 98% of the predicted values are tightly clustered just below 1 and above zero, indicating that the Lasso regression is far less effective at estimating a range of probabilities than the Linear Probability Model than is Ridge.

5.2 Selection and sensitivity analysis of λ

Unlike OLS, Ridge and its alternatives are affected by the modeler's choice of λ (or for PCR, the number of principal components). We first outline our data-driven method for selecting those parameters — k-fold cross-validation — and then demonstrate how changes in those parameters affect the models.

We use a process, called k-fold cross validation, to select the λ^* or number of principal components, which minimizes the out-of-sample mean square prediction error (MSPE).

1. Divide the sample into k random subsets of equal size, or roughly equal size if division results in a remainder.

2. Generate a sequence of $\lambda_i{}_{i=1}^{m}$. The maximum λ is the smallest λ that forces all coefficients to converge to zero. The rest of the sequence is generated by requiring a constant ratio between each consecutive λ. In our application, we implement this using the `cv.glmnet` function in R's `glmnet` library.

3. Choose one subset to use as a test subset. Use the remaining subsets as training data for the regression, which uses the λ which was assigned to the subset in step 2.

4. For every $\lambda_i{}_{i=1}^{m}$, generate predicted values of the test subset after training with the remaining subsets. Measure MSPE for each λ_i.

5. Repeat 3 and 4 so that every subset is used as a test subset once.

6. There are now $k \times m$ MSPEs. Average MSPEs across k for each λ_i. Identify the minimum MSPE. Return the λ associated with the minimum MSPE.

The process for PCR is nearly the same; the sequence is simply the number of principal components.

Usually, the results of k-fold cross validation vary by the λ tested with as well as the value of k. Formally, "leave-one-out" cross validation, where $k = n - 1$ (i.e., one less than the sample size) is the most rigorous and thus returns a value for λ closest to the optimal λ that would minimize MSE and MSPE. However, cross validation is expensive. As we perform Ridge Regression with k-fold cross validation to determine λ, we note that it has a much greater run-time, reflecting the computational complexity of the task. While "leave-one-out" is more accurate, it is too computationally intensive to be practical. "Leave-one-out" would require many fits of the model. In this project, we set $k = 10$.

Figure 3 shows for each of the methods that the cross-validated parameter minimizes, or nearly minimizes MSPE. In Ridge, it is not clear that there is convergence in the MSPE curve. In Lasso, it is much more plausible that MSPE converges to its minimum. In PCR, the marginal benefit of an additional component is nearly zero after 57 components.

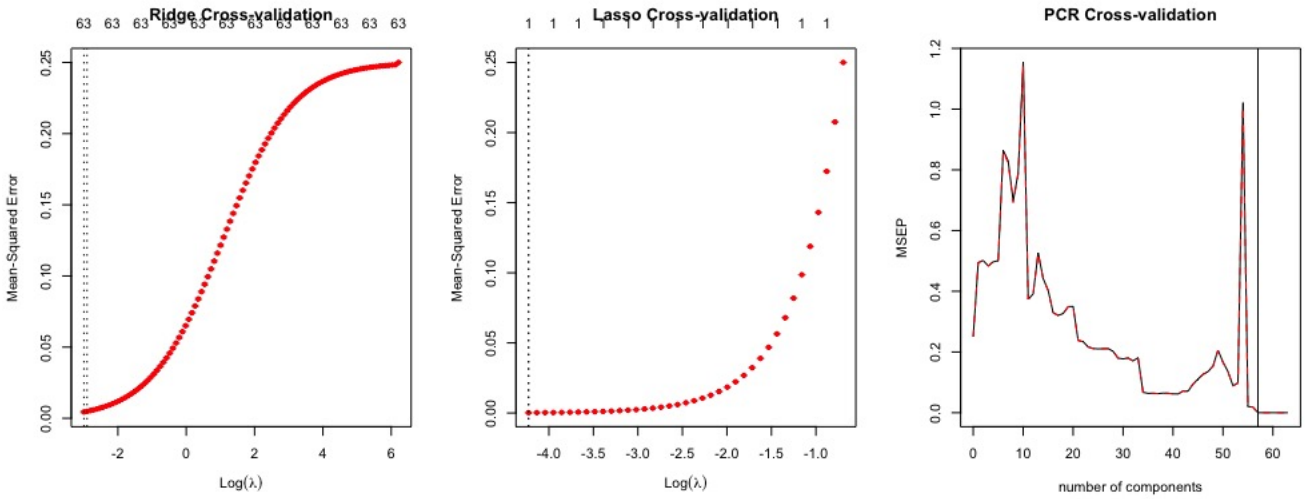

Figure 3: Cross-validation plots for Ridge, Lasso, and PCR

We also consider how sensitive to λ are the Ridge and Lasso estimates. The reliability of a Ridge regression model is indicated by the stability of the coefficients with respect to λ; if they fluctuate wildly in magnitude and sign until suddenly converging, then the coefficients may not be particularly useful for prediction, since predicted values will fluctuate accordingly. On the other hand, if coefficients quickly settle on a stable path of convergence towards zero, then the choice of λ clearly only affects the variance of the parameters.

Figure 4 shows how the magnitude of the coefficients change as λ increases. From the plots, we can determine that Ridge is far more stable and reliable a method than Lasso for our problem. All coefficients paths smoothly approach zero, and all do so monotonically at approximately $log(\lambda) = -1$, or when $\lambda = 1$. We cannot say the same for Lasso: one coefficient actually increases, and then reverses direction to approach zero. Note that the upper x-axis reports how many non-zero coefficients there are for a given level of $log(\lambda)$. Lasso collapses most of the coefficients in the 63-regressor specification to zero immediately.

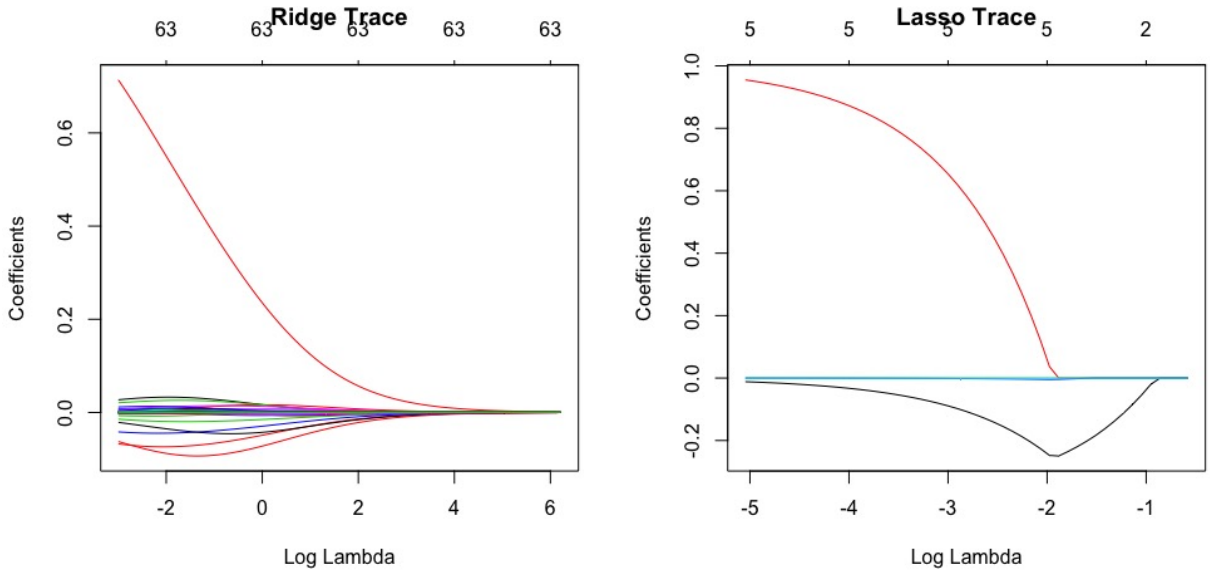

Figure 4: Effect of λ on coefficients

6 Discussion

Feelings about the results go here, once we have them.

As we mentioned in section 3.3, the linear probability model we estimate is flawed. It is possible that predicted values outside of [0,1], and table 2 shows that this does happen. One way that we can retain the LPM used by all methods in section 5.1 is by omitting those values when cross-validating. However, this does not prevent them from occurring again when predicting out of sample, which is essential to our problem. Alternatively, we reasonably assume that the magnitude of invalid predicted values are overestimated (versus a logistic regression approach, which strictly bounds the dependent variable between 0 and 1). For example, we would not take $\hat{y}_i = 1.23$ on face value, but treat it as some high probability close to 1; similarly, we would treat $\hat{y}_i = -0.01$ as some very low probability close to 0.

With this rounding assumption, we can then use a threshold rule to classify borrowers by their default-likelihood. That is, we say that any borrower whose predicted default-likelihood is greater than or equal to some p is "default-prone," and all others are "safe." We can also introduce additional thresholds and multiple classes; for example, specifying four probability thresholds that assign borrowers to one of five risk classes. Note: this is essentially a negative version of the loan grade assigned by LendingClub in section 4, and is often employed in credit profiling and risk assessment.

For two classes, it is unclear how to determine the threshold; for many thresholds, classification amounts to simplifying the *relative* risk of a particular borrower. Although we do not apply a threshold rule to classify borrowers, it would be very straightforward to include.

7 Conclusion

This paper has shown that Ridge regression is an effective tool for predicting the likelihood of default on a cross-section of borrowers and their corresponding loans. We contextualized Ridge alongside alternative methods, Lasso and principal components regression; we outlined an error-minimized cross-fold validation method for selection the necessary parameters; and we illustrated the sensitivity of the Ridge regression method to the value of the user-defined shrinkage parameter. We discussed the validity of the results given the probabilistic nature of the problem, and how demonstrated how a classification rule might be applied to our approach. The product of our paper — a method for predicting the likelihood of default — is an essential component of risk calculation and asset management strategy for financial firms, even if those firms do not lend directly to consumers.

8 Contributions

Edbert did about 60% of the initial coding implementation, debugged and commented for all code to prepare it for submission, did some of the math research for least squares regression, Ridge regression, and k-fold cross validation, and helped Robert prepare table 2. Robert researched Ridge to correctly design the model, obtained the data, performed exploratory data analysis and the majority of data cleaning, planned and wrote the presentation, and wrote this paper. Robert was the creative mind that most guided the direction this project took.

9 Appendix

Table 3:

Statistic	Mean	St. Dev.
payment_status	0.000	0.000
loan_amnt	15,286.620	9,674.561
int_rate	12.686	4.956
annual_inc	81,512.120	129,412.800
dti	19.458	16.853
delinq_2yrs	0.301	0.868
fico_range_high	705.997	34.649
revol_util	47.037	24.778
total_rec_prncp	8,053.515	7,900.010

Table 4:

Statistic	Mean	St. Dev.
payment_status	0.000	0.000
loan_amnt	15,277.990	9,679.847
int_rate	12.678	4.940
annual_inc	81,568.630	85,594.490
dti	19.391	15.691
delinq_2yrs	0.301	0.873
fico_range_high	705.821	34.527
revol_util	47.146	24.713
total_rec_prncp	8,011.836	7,862.078

Table 5:

Statistic	Mean	St. Dev.	Min	Max
Payment status	0.500	0.500	0	1
Loan amount	−0.000	1.000	−1.531	2.581
Int. rate	−0.000	1.000	−1.639	3.025
Annual inc.	−0.000	1.000	−1.018	104.440
Debt-income ratio	−0.000	1.000	−1.376	63.596
Delinquent w/in 2 yrs	0.000	1.000	−0.364	30.855
FICO score, u.b.	−0.000	1.000	−1.112	4.752
Revolving util.	−0.000	1.000	−2.009	4.599
Total princ. paid	−0.000	1.000	−0.886	5.292

Table 6:

Statistic	Mean	St. Dev.	Min	Max
Payment status	0.494	0.500	0	1
Loan amount	−0.006	0.993	−1.531	2.581
Int. rate	0.001	1.000	−1.639	3.025
Annual inc.	0.012	1.240	−0.993	102.132
Debt-income ratio	−0.018	0.857	−1.311	47.231
Delinquent w/in 2 yrs	−0.008	1.011	−0.364	16.860
FICO score, u.b.	0.013	1.017	−1.112	4.752
Revolving util.	−0.009	0.998	−2.009	3.351
Total princ. paid	0.009	1.004	−0.886	5.292

Table 7:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
payment_status	10,000	0.49	0.50	0	0	1	1
loan_amnt	10,000	−0.01	0.99	−1.53	−0.79	0.57	2.58
int_rate	10,000	0.001	1.00	−1.64	−0.70	0.57	3.03
installment	10,000	−0.01	1.00	−1.54	−0.73	0.54	4.31
annual_inc	10,000	0.01	1.24	−0.99	−0.40	0.21	102.13
dti	10,000	−0.02	0.86	−1.31	−0.48	0.36	47.23
delinq_2yrs	10,000	−0.01	1.01	−0.36	−0.36	−0.36	16.86
fico_range_low	10,000	0.01	1.02	−1.11	−0.80	0.46	4.72
fico_range_high	10,000	0.01	1.02	−1.11	−0.80	0.46	4.75
open_acc	10,000	−0.01	1.00	−1.87	−0.67	0.52	9.07
pub_rec	10,000	−0.002	0.95	−0.38	−0.38	−0.38	14.04
revol_bal	10,000	0.002	0.98	−0.73	−0.47	0.16	24.65
revol_util	10,000	−0.01	1.00	−2.01	−0.75	0.73	3.35
total_acc	10,000	−0.01	1.00	−1.76	−0.76	0.49	5.97
out_prncp	10,000	−0.01	0.97	−0.51	−0.51	0.08	4.96
total_pymnt	10,000	0.01	1.01	−1.11	−0.70	0.37	5.32
total_rec_prncp	10,000	0.01	1.00	−0.89	−0.64	0.25	5.29
total_rec_int	10,000	0.01	1.01	−0.96	−0.66	0.32	9.04
total_rec_late_fee	10,000	−0.01	0.90	−0.21	−0.21	−0.21	29.47
last_fico_range_low	10,000	0.01	0.98	−3.41	−0.28	0.60	1.40
last_fico_range_high	10,000	0.01	1.00	−1.52	−0.90	0.84	2.43
tot_cur_bal	10,000	0.005	1.03	−0.89	−0.69	0.40	17.00
open_acc_6m	10,000	−0.001	1.00	−0.86	−0.86	0.77	8.11
open_il_12m	10,000	−0.002	1.00	−0.79	−0.79	0.22	14.33
open_il_24m	10,000	−0.001	1.00	−1.05	−0.45	0.15	11.46
mths_since_rcnt_il	10,000	0.01	1.03	−0.78	−0.55	0.12	14.58
total_bal_il	10,000	−0.01	0.99	−0.85	−0.61	0.20	13.37
avg_cur_bal	10,000	0.01	1.05	−0.83	−0.63	0.29	20.50
tot_hi_cred_lim	10,000	0.005	1.02	−0.98	−0.69	0.41	16.45
inq_last_12m	10,000	0.01	1.01	−0.90	−0.51	0.26	10.70
acc_open_past_24mths	10,000	0.001	1.00	−1.47	−0.60	0.56	6.92
pct_tl_nvr_dlq	10,000	0.01	1.01	−8.24	−0.31	0.67	0.67
payment_statussq1	10,000	0.49	0.50	0	0	1	1
loan_amntsq1	10,000	0.99	1.30	0.0000	0.14	1.23	6.66
int_ratesq1	10,000	1.00	1.50	0.001	0.09	1.34	9.15
installmentsq1	10,000	1.00	1.58	0.0000	0.13	1.16	18.57
annual_incsq1	10,000	1.54	104.39	0.00	0.03	0.31	10,430.89
dtisq1	10,000	0.74	22.96	0.00	0.04	0.50	2,230.80
delinq_2yrssq1	10,000	1.02	7.85	0.13	0.13	0.13	284.28
fico_range_lowsq1	10,000	1.03	2.04	0.0001	0.10	0.91	22.29
fico_range_highsq1	10,000	1.03	2.04	0.0001	0.10	0.91	22.58
open_accsq1	10,000	1.00	2.37	0.0001	0.11	1.03	82.24
pub_recsq1	10,000	0.91	5.51	0.15	0.15	0.15	197.07
revol_balsq1	10,000	0.96	10.03	0.00	0.04	0.34	607.62
revol_utilsq1	10,000	1.00	1.11	0.0000	0.13	1.53	11.23
total_accsq1	10,000	0.99	1.84	0.0001	0.12	1.19	35.69
out_prncpsq1	10,000	0.95	2.55	0.0000	0.26	0.26	24.62
total_pymntsq1	10,000	1.02	2.42	0.0000	0.11	0.82	28.28
total_rec_prncpsq1	10,000	1.01	2.76	0.00	0.11	0.60	28.01
total_rec_intsq1	10,000	1.02	3.29	0.0000	0.10	0.69	81.68
total_rec_late_feesq1	10,000	0.82	10.66	0.04	0.04	0.04	868.20
last_fico_range_lowsq1	10,000	0.97	2.71	0.0000	0.06	0.47	11.60
last_fico_range_highsq1	10,000	1.01	0.90	0.0000	0.26	1.52	5.89
tot_cur_balsq1	10,000	1.06	4.65	0.0000	0.18	0.65	289.01
open_acc_6msq1	10,000	1.01	2.51	0.002	0.002	0.75	65.83
open_il_12msq1	10,000	0.99	3.06	0.05	0.05	0.62	205.31
open_il_24msq1	10,000	1.00	2.79	0.02	0.20	1.09	131.37
mths_since_rcnt_ilsq1	10,000	1.05	4.84	0.0000	0.06	0.39	212.48
total_bal_ilsq1	10,000	0.97	4.43	0.00	0.07	0.61	178.69
avg_cur_balsq1	10,000	1.11	6.77	0.0000	0.13	0.52	420.39
tot_hi_cred_limsq1	10,000	1.04	4.48	0.00	0.15	0.68	270.57
inq_last_12msq1	10,000	1.03	3.28	0.02	0.07	0.81	114.49
acc_open_past_24mthssq1	10,000	1.00	2.12	0.0005	0.10	1.29	47.91
pct_tl_nvr_dlqsq1	10,000	1.01	3.02	0.0000	0.14	0.45	67.89

Table 8: Non-defaulters in the unbalanced sample

Statistic	Mean	St. Dev.
payment_status	0.000	0.000
loan_amnt	15,286.620	9,674.561
int_rate	12.686	4.956
annual_inc	81,512.120	129,412.800
dti	19.458	16.853
delinq_2yrs	0.301	0.868
fico_range_high	705.997	34.649
revol_util	47.037	24.778
total_rec_prncp	8,053.515	7,900.010

Table 9: Non-defaulters in the balanced sample

Statistic	Mean	St. Dev.
payment_status	0.000	0.000
loan_amnt	15,277.990	9,679.847
int_rate	12.678	4.940
annual_inc	81,568.630	85,594.490
dti	19.391	15.691
delinq_2yrs	0.301	0.873
fico_range_high	705.821	34.527
revol_util	47.146	24.713
total_rec_prncp	8,011.836	7,862.078

Table 10: Select statistics of the balanced sample (standardized)

Statistic	Mean	St. Dev.	Min	Max
payment_status	0.500	0.500	0	1
loan_amnt	−0.000	1.000	−1.531	2.581
int_rate	−0.000	1.000	−1.639	3.025
annual_inc	−0.000	1.000	−1.018	104.440
dti	−0.000	1.000	−1.376	63.596
delinq_2yrs	0.000	1.000	−0.364	30.855
fico_range_high	−0.000	1.000	−1.112	4.752
revol_util	−0.000	1.000	−2.009	4.599
total_rec_prncp	−0.000	1.000	−0.886	5.292

Table 11: Select statistics of the subsample (standardized)

Statistic	Mean	St. Dev.	Min	Max
payment_status	0.494	0.500	0	1
loan_amnt	−0.006	0.993	−1.531	2.581
int_rate	0.001	1.000	−1.639	3.025
annual_inc	0.012	1.240	−0.993	102.132
dti	−0.018	0.857	−1.311	47.231
delinq_2yrs	−0.008	1.011	−0.364	16.860
fico_range_high	0.013	1.017	−1.112	4.752
revol_util	−0.009	0.998	−2.009	3.351
total_rec_prncp	0.009	1.004	−0.886	5.292

Table 12: Summary statistics of the subsample (standardized)

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
payment_status	10,000	0.49	0.50	0	0	1	1
loan_amnt	10,000	−0.01	0.99	−1.53	−0.79	0.57	2.58
int_rate	10,000	0.001	1.00	−1.64	−0.70	0.57	3.03
installment	10,000	−0.01	1.00	−1.54	−0.73	0.54	4.31
annual_inc	10,000	0.01	1.24	−0.99	−0.40	0.21	102.13
dti	10,000	−0.02	0.86	−1.31	−0.48	0.36	47.23
delinq_2yrs	10,000	−0.01	1.01	−0.36	−0.36	−0.36	16.86
fico_range_low	10,000	0.01	1.02	−1.11	−0.80	0.46	4.72
fico_range_high	10,000	0.01	1.02	−1.11	−0.80	0.46	4.75
open_acc	10,000	−0.01	1.00	−1.87	−0.67	0.52	9.07
pub_rec	10,000	−0.002	0.95	−0.38	−0.38	−0.38	14.04
revol_bal	10,000	0.002	0.98	−0.73	−0.47	0.16	24.65
revol_util	10,000	−0.01	1.00	−2.01	−0.75	0.73	3.35
total_acc	10,000	−0.01	1.00	−1.76	−0.76	0.49	5.97
out_prncp	10,000	−0.01	0.97	−0.51	−0.51	0.08	4.96
total_pymnt	10,000	0.01	1.01	−1.11	−0.70	0.37	5.32
total_rec_prncp	10,000	0.01	1.00	−0.89	−0.64	0.25	5.29
total_rec_int	10,000	0.01	1.01	−0.96	−0.66	0.32	9.04
total_rec_late_fee	10,000	−0.01	0.90	−0.21	−0.21	−0.21	29.47
last_fico_range_low	10,000	0.01	0.98	−3.41	−0.28	0.60	1.40
last_fico_range_high	10,000	0.01	1.00	−1.52	−0.90	0.84	2.43
tot_cur_bal	10,000	0.005	1.03	−0.89	−0.69	0.40	17.00
open_acc_6m	10,000	−0.001	1.00	−0.86	−0.86	0.77	8.11
open_il_12m	10,000	−0.002	1.00	−0.79	−0.79	0.22	14.33
open_il_24m	10,000	−0.001	1.00	−1.05	−0.45	0.15	11.46
mths_since_rcnt_il	10,000	0.01	1.03	−0.78	−0.55	0.12	14.58
total_bal_il	10,000	−0.01	0.99	−0.85	−0.61	0.20	13.37
avg_cur_bal	10,000	0.01	1.05	−0.83	−0.63	0.29	20.50
tot_hi_cred_lim	10,000	0.005	1.02	−0.98	−0.69	0.41	16.45
inq_last_12m	10,000	0.01	1.01	−0.90	−0.51	0.26	10.70
acc_open_past_24mths	10,000	0.001	1.00	−1.47	−0.60	0.56	6.92
pct_tl_nvr_dlq	10,000	0.01	1.01	−8.24	−0.31	0.67	0.67

References

[1] J. E. Anderson, "Ridge estimation of house value determinants," *Journal of Urban Economics*, vol. 9, no. 3, pp. 286–297, May 1981, doi: 10.1016/0094-1190(81)90028-0.

[2] E. Cule and M. De Iorio, "Ridge Regression in Prediction Problems: Automatic Choice of the Ridge Parameter," *Genet. Epidemiol.*, vol. 37, no. 7, pp. 704–714, Nov. 2013, doi: 10.1002/gepi.21750.

[3] R. W. Hoerl, "Ridge Regression: A Historical Context," *Technometrics*, vol. 62, no. 4, pp. 420–425, Oct. 2020, doi: 10.1080/00401706.2020.1742207.

[4] T. O. Kvålseth, "Ridge regression models of urban crime," *Regional Science and Urban Economics*, vol. 9, no. 2, pp. 247–260, May 1979, doi: 10.1016/0166-0462(79)90015-2.

[5] S. M. Lawrence and J. K. Smith, "Projecting the net migration rate of the school age population," *Socio-Economic Planning Sciences*, vol. 18, no. 1, pp. 1–14, Jan. 1984, doi: 10.1016/0038-0121(84)90023-5.

[6] R. Ramanathan, "E-commerce success criteria: determining which criteria count most," *Electron Commer Res*, vol. 10, no. 2, pp. 191–208, Jun. 2010, doi: 10.1007/s10660-010-9051-3.

[7] J. H. Stock and M. W. Watson, *Introduction to econometrics*, Fourth edition, Global edition. New York: Pearson, 2020.