Problem Set 3: Prediction and Regularization

Gov 51 — Spring 2026

2026-04-02

Reading: Baker (2025), "Statistical Learning Can Help the Judiciary Fulfill Its Gatekeeping Role Over Expert Witnesses," 22 *Berkeley Bus. L.J.* 264.

Data: `compas_clean.csv` — cleaned version of the ProPublica COMPAS recidivism dataset.

Submission: Push your completed `.qmd` file and rendered PDF to your GitHub repository by 11:59 PM on April 2.

Packages: You will need `tidyverse`, `glmnet`, and `knitr`. Install `glmnet` with `install.packages("glmnet")` if needed.

1 Setup and Data Exploration (10 points)

1.1 Load packages and data

Load the `tidyverse`, `glmnet`, and `knitr` packages. Then load the COMPAS dataset from `data/compas_clean.csv`.

```
library(tidyverse)
library(glmnet)
library(knitr)

compas <- read_csv("data/compas_clean.csv")
```

1.2 How many observations and variables are in the dataset? What is the unit of observation?

1.3 Examine the outcome variable

The variable `recidivism` equals 1 if the person was rearrested within two years and 0 otherwise.

(a) What is the overall recidivism rate in the data?
(b) Is this outcome continuous or binary? Why does this matter for how we interpret predictions from a linear model?

1.4 Summary statistics

Create a summary statistics table showing the mean and standard deviation of all variables in the dataset. Briefly note which variables have the most variation and which have the least.

1.5 Explore key predictors

Create a bar chart showing the recidivism rate by number of prior offenses (`priors_count`). Group `priors_count` into categories: 0, 1–2, 3–5, 6–10, and 11+. What pattern do you see?

2 OLS Baseline (15 points)

2.1 Train/test split

Set your seed to `51` for reproducibility. Split the data into a training set (80%) and a test set (20%).

```
set.seed(51)
n <- nrow(compas)
train_idx <- sample(1:n, size = floor(0.8 * n))
train <- compas[train_idx, ]
test  <- compas[-train_idx, ]
```

How many observations are in each set?

2.2 Fit OLS with main effects only

Fit an OLS regression of `recidivism` on all other variables in the training data using `lm(recidivism ~ ., data = train)`. This gives you a model with 16 predictors.

(a) Which predictors are statistically significant at the 5% level?
(b) Interpret the coefficient on `priors_count` in one sentence.

2.3 In-sample vs. out-of-sample RMSE

Root Mean Squared Error (RMSE) measures average prediction error:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

(a) Compute the in-sample RMSE (predictions on the **training** data).
(b) Compute the out-of-sample RMSE (predictions on the **test** data).
(c) How large is the gap between in-sample and out-of-sample RMSE? Is OLS overfitting with 16 predictors?

Hint: Use `predict(model, newdata = ...)` to generate predictions, then compute RMSE manually.

2.4 Cross-validation by hand

A single train/test split depends on which observations happened to land in each group. **Cross-validation** averages over multiple splits to give a more stable estimate of out-of-sample performance.

Implement 5-fold cross-validation for the simple OLS model:

1. Randomly assign each training observation to one of 5 folds.
2. For each fold $k = 1, \ldots, 5$: fit OLS on the other 4 folds, predict fold k, compute RMSE.
3. Average the 5 RMSE values.

```
# Assign folds
set.seed(51)
folds <- sample(rep(1:5, length.out = nrow(train)))

rmse_cv <- numeric(5)
for (k in 1:5) {
  train_k <- train[folds != k, ]
  test_k  <- train[folds == k, ]
  model_k <- lm(recidivism ~ ., data = train_k)
  pred_k  <- predict(model_k, newdata = test_k)
  rmse_cv[k] <- sqrt(mean((test_k$recidivism - pred_k)^2))
}

mean(rmse_cv)
```

(a) What is the 5-fold CV estimate of RMSE? How does it compare to your single-split out-of-sample RMSE?

(b) In your own words, why is cross-validation more reliable than a single train/test split?

(c) When we use `cv.glmnet()` later, it does this same process internally to choose the penalty parameter λ. Why is cross-validation the right tool for choosing λ?

3 The Kitchen Sink: What Happens When You Add Too Many Predictors? (15 points)

The simple OLS model uses 16 main effects. But maybe interactions between variables would help. What if being young AND having prior offenses is different from the sum of each effect alone? Let's find out what happens when we throw in *everything*.

3.1 Build the expanded model

The formula below creates all pairwise interactions plus squared terms for the continuous variables. This expands our feature set from 16 to about 141 predictors.

```
# Full interaction formula
f_full <- recidivism ~ (.)^2 +
  I(age^2) + I(priors_count^2) +
  I(juv_fel_count^2) + I(juv_misd_count^2) +
  I(juv_other_count^2)

# Create design matrices for glmnet (we'll need these later)
X_train <- model.matrix(f_full, data = train)[, -1]
X_test  <- model.matrix(f_full, data = test)[, -1]
y_train <- train$recidivism
y_test  <- test$recidivism

# How many features now?
ncol(X_train)
```

(a) How many predictors does this expanded model have?
(b) What is the ratio of training observations to predictors? (Compare to the simple model.)

3.2 Fit OLS with the kitchen sink

Fit OLS on the expanded feature set: `ols_full <- lm(f_full, data = train)`.

(a) Compute the in-sample RMSE for this model.
(b) Compute the out-of-sample RMSE.
(c) How does the in-sample/out-of-sample gap compare to the simple 19-predictor OLS? What happened and why?

3.3 The overfitting lesson

In 2–3 sentences, explain why adding 180 interaction terms made the model's **in-sample** fit better but its **out-of-sample** predictions worse. What is this phenomenon called?

4 Ridge Regression: Shrink Everything (15 points)

Ridge regression adds an L2 penalty that shrinks all coefficients toward zero. Larger λ means more shrinkage. The key idea: Ridge trades a small amount of bias for a large reduction in variance.

4.1 Fit Ridge with cross-validation

Use `cv.glmnet()` with `alpha = 0` to fit Ridge regression on the **expanded** training data (the `X_train` matrix with ~141 features). Use 10-fold cross-validation (the default) to select the optimal λ.

```
set.seed(51)  # for reproducible CV folds
cv_ridge <- cv.glmnet(X_train, y_train, alpha = 0, nfolds = 10)
```

(a) What is the optimal λ selected by cross-validation? (Use `lambda.min`.)
(b) Plot the cross-validation curve using `plot(cv_ridge)`. What does this plot show?

4.2 Ridge RMSE

Compute the out-of-sample RMSE for Ridge. Compare to both the simple OLS (16 vars) and the kitchen-sink OLS (~141 vars). What happened?

4.3 Did Ridge set any coefficients to zero?

Extract the Ridge coefficients using `coef(cv_ridge, s = "lambda.min")`. Did Ridge eliminate any variables? Why or why not?

5 LASSO Regression: Shrink and Select (15 points)

LASSO (Least Absolute Shrinkage and Selection Operator) uses an L1 penalty. Unlike Ridge, LASSO can shrink coefficients all the way to **exactly zero**, effectively performing variable selection.

5.1 Fit LASSO with cross-validation

Use `cv.glmnet()` with `alpha = 1` on the expanded training data. Remember to set your seed first for reproducibility.

```
set.seed(51)
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1, nfolds = 10)
```

(a) What is the optimal λ?
(b) Plot the cross-validation curve.

5.2 LASSO variable selection

Extract the LASSO coefficients at `lambda.min`.

(a) How many of the ~141 variables survived (non-zero coefficient)? How many were eliminated?
(b) List a few of the surviving interaction terms. Do they make substantive sense?
(c) Baker (2025) argues that LASSO reduces "subjective discretion" in model building. How does what you see here illustrate that argument?

5.3 LASSO RMSE

Compute the out-of-sample RMSE for LASSO. How does it compare to OLS (both versions) and Ridge?

6 Model Comparison and Interpretation (30 points)

6.1 Elastic Net

Fit an Elastic Net with `alpha = 0.5` on the expanded training data. Remember to set your seed first. Compute its out-of-sample RMSE and count how many variables it retains.

```
set.seed(51)
cv_enet <- cv.glmnet(X_train, y_train, alpha = 0.5, nfolds = 10)
```

6.2 The big comparison table

Fill in this table with your results:

Model	Predictors	Out-of-Sample RMSE
OLS (main effects, 16 vars)		
OLS (kitchen sink, ~141 vars)		
Ridge		
LASSO		
Elastic Net		

(a) Which model has the lowest out-of-sample RMSE?
(b) Which model has the best in-sample fit? Is that the same model?
(c) In 2–3 sentences, explain what this table teaches about the bias-variance tradeoff.

6.3 Baker article connection

In Baker (2025), two experts in the *Halliburton* securities case selected different peer firms and reached different conclusions about whether a stock drop was statistically significant.

(a) How did LASSO help narrow the disagreement between dueling experts?
(b) Based on your experience with LASSO in this problem set, what aspect of the traditional OLS approach gives the analyst the most room for manipulation?

6.4 Prediction vs. explanation

We used these models to *predict* recidivism. Suppose instead you wanted to know whether `priors_count` *causes* higher recidivism.

(a) Would the LASSO coefficient on `priors_count` answer this causal question? Why or why not?
(b) What is the difference between using regression for prediction versus using it for causal inference?

6.5 Algorithmic fairness

Algorithms like COMPAS are used in real courtrooms to inform bail and sentencing decisions.

(a) Using your simple OLS model (16 vars), compute the mean predicted recidivism probability separately for Black defendants and white defendants in the test set. Is there a difference?

Hint: White is the reference category (all race dummies equal 0). You can identify white defendants with `black == 0 & hispanic == 0 & asian == 0 & other_race == 0 & native_american == 0`. (b) Now compute the actual recidivism rate for each group. Compare the predictions to the actual rates. (c) In 2–3 sentences: What are the tradeoffs of using prediction algorithms in criminal justice? Reference something specific from your analysis.