STA 141c Project

Yikai Lu

2025-06-07

lasso and ridge regression

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidy
verse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────────
───── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to bec
ome errors
```

```
library(glmnet)
```

```
## 载入需要的程序包：Matrix
##
## 载入程序包：'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
train <- read_csv("D:/Yikai university work/spring quarter 2025/STA 141c/interaction_features_l
asso_ridge.csv")
```

```
## Rows: 1460 Columns: 290
## ── Column specification ────────────────────────────────────────
────────────────────────────
## Delimiter: ","
## dbl (290): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, M...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
y <- train$SalePrice
X <- train %>% select(-Id, -SalePrice)
cat("Summarizing missing values...\n")
```

```
## Summarizing missing values...
```

```
missing_counts <- colSums(is.na(X))
missing_pct <- (missing_counts / nrow(X)) * 100
missing_df <- data.frame(
  Variable = names(missing_counts),
  MissingCount = missing_counts,
  MissingPct = round(missing_pct, 2)
)
missing_df <- missing_df[missing_df$MissingCount > 0, ]
missing_df <- missing_df[order(-missing_df$MissingPct), ]
print(missing_df)
```

```
## [1] Variable     MissingCount MissingPct
## <0 行> (或0-长度的row.names)
```

```
# Optional: Drop columns with >90% missing (customizable)
drop_cols <- missing_df %>% filter(MissingPct > 90) %>% pull(Variable)
if (length(drop_cols) > 0) {
  cat("🗑 Dropping high-missing columns:\n")
  print(drop_cols)
  X <- X %>% select(-all_of(drop_cols))
}
# Numeric → median imputation
num_vars <- sapply(X, is.numeric)
X[num_vars] <- lapply(X[num_vars], function(col) {
  col[is.na(col)] <- median(col, na.rm = TRUE)
  col
})
# Categorical → mode imputation
cat_vars <- sapply(X, is.character)
X[cat_vars] <- lapply(X[cat_vars], function(col) {
  mode_val <- names(sort(table(col), decreasing = TRUE))[1]
  col[is.na(col)] <- mode_val
  col
})
X <- X[, sapply(X, function(col) length(unique(col)) > 1)]
X <- as.data.frame(X)
names(X) <- make.names(names(X), unique = TRUE)
if (length(names(X)) == 0) {
  stop("No valid predictors left after filtering.")
}

X_formula <- as.formula(paste("~", paste(names(X), collapse = "+")))
X_model <- model.matrix(X_formula, data = X)[, -1]
cat("Final model matrix created with", nrow(X_model), "rows and", ncol(X_model), "columns.\n")
```

```
## Final model matrix created with 1460 rows and 287 columns.
```

```
# Fit Lasso regression
set.seed(123)
lasso_cv <- cv.glmnet(X_model, y, alpha = 1)  # Lasso uses alpha = 1
# Best lambda
best_lambda_lasso <- lasso_cv$lambda.min
# Predict and calculate RMSE
lasso_preds <- predict(lasso_cv, s = best_lambda_lasso, newx = X_model)
lasso_rmse <- sqrt(mean((lasso_preds - y)^2))

cat("lasso Results:\n")
```

```
## lasso Results:
```

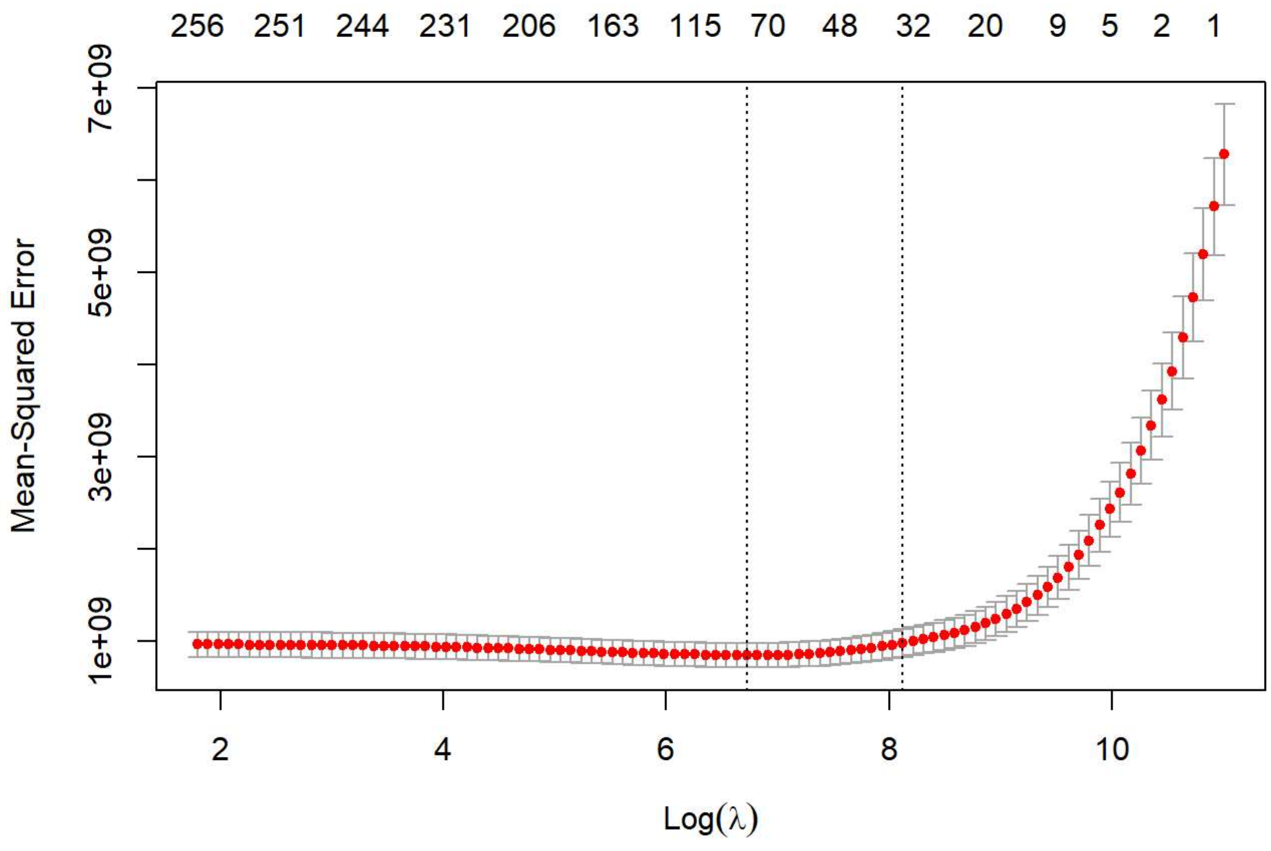
```
cat("Best lambda:", best_lambda_lasso, "\n")
```

```
## Best lambda: 834.4031
```

```
cat("Lasso RMSE:", lasso_rmse, "\n")
```

```
## Lasso RMSE: 23403.04
```

```
plot(lasso_cv)
```



```
# Fit Ridge regression-
set.seed(123)
ridge_cv <- cv.glmnet(X_model, y, alpha = 0)  # Ridge uses alpha = 0
# Best lambda
best_lambda_ridge <- ridge_cv$lambda.min
# Predict and calculate RMSE
ridge_preds <- predict(ridge_cv, s = best_lambda_ridge, newx = X_model)
ridge_rmse <- sqrt(mean((ridge_preds - y)^2))
cat("Ridge Results:\n")
```

```
## Ridge Results:
```

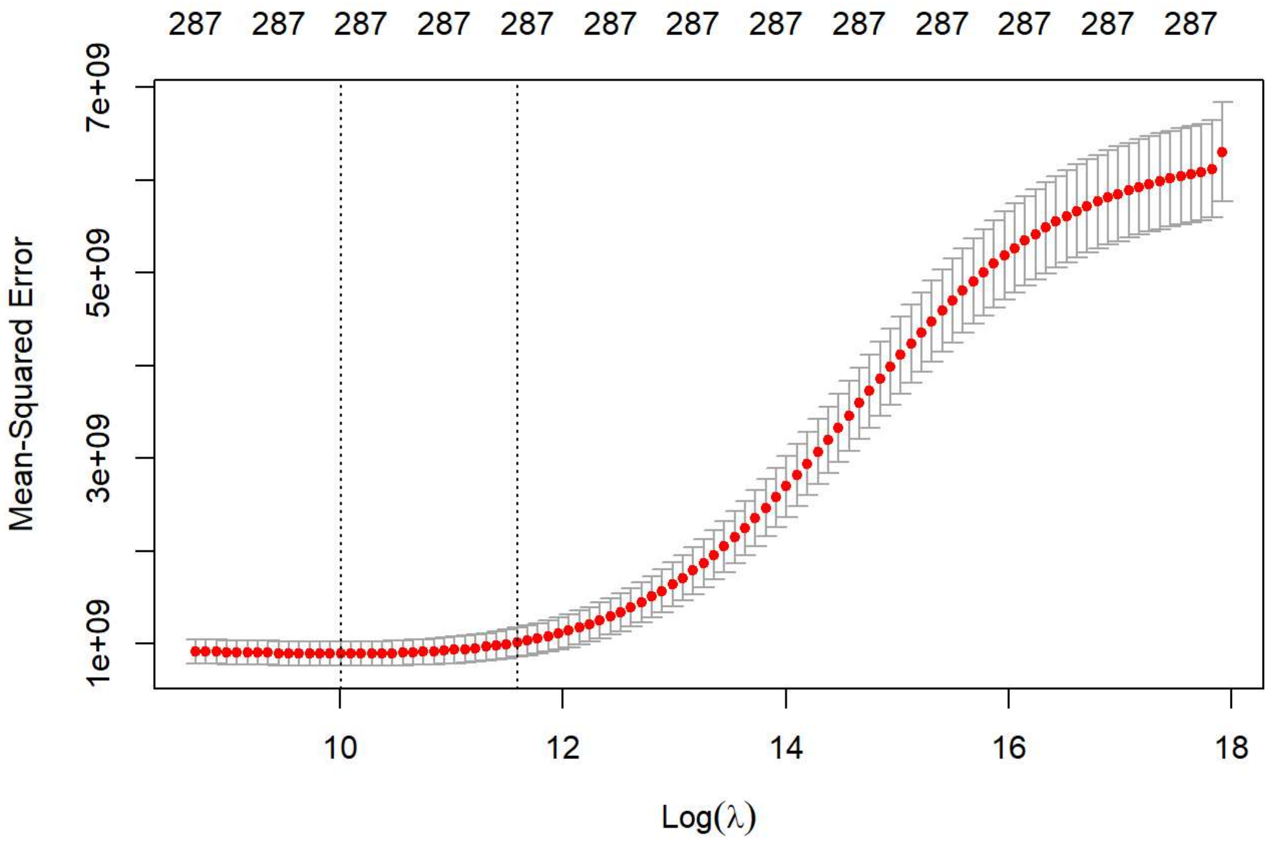
```
cat("Best lambda:", best_lambda_ridge, "\n")
```

```
## Best lambda: 22162.48
```

```
cat("Ridge RMSE:", ridge_rmse, "\n")
```

```
## Ridge RMSE: 23292.09
```

```
plot(ridge_cv)
```



```
#try to compare these two model
cat("lasso RMSE:", lasso_rmse, "\n")
```

```
## lasso RMSE: 23403.04
```

```
cat("Ridge RMSE:", ridge_rmse, "\n")
```

```
## Ridge RMSE: 23292.09
```

```
# RSS on training data
lasso_rss <- sum((lasso_preds - y)^2)
ridge_rss <- sum((ridge_preds - y)^2)
cat("Lasso Training RSS:", lasso_rss, "\n")
```

```
## Lasso Training RSS: 799645406553
```

```
cat("Ridge Training RSS:", ridge_rss, "\n")
```

```
## Ridge Training RSS: 792081637749
```

```
lasso_coef_full <- coef(lasso_cv, s = "lambda.min")
lasso_coef_df <- as.data.frame(as.matrix(lasso_coef_full))
colnames(lasso_coef_df) <- "Coefficient"
lasso_coef_df$Feature <- rownames(lasso_coef_df)

# Filter out and select top 15 absolute coefficients
top15_lasso <- lasso_coef_df %>%
  filter(Feature != "(Intercept)") %>%
  mutate(abs_coef = abs(Coefficient)) %>%
  arrange(desc(abs_coef)) %>%
  slice(1:15)

cat(" Top 15 Lasso Features:\n")
```

```
##  Top 15 Lasso Features:
```

```
print(top15_lasso)
```

```
##                        Coefficient              Feature  abs_coef
## RoofMatl_ClyTile      -310652.37     RoofMatl_ClyTile 310652.37
## Condition2_PosN       -145982.52      Condition2_PosN 145982.52
## RoofMatl_WdShngl        53766.67     RoofMatl_WdShngl  53766.67
## Functional_Sev         -40792.74       Functional_Sev  40792.74
## Neighborhood_NoRidge    33299.08 Neighborhood_NoRidge  33299.08
## Neighborhood_StoneBr    33205.35 Neighborhood_StoneBr  33205.35
## GrLivArea               24288.31            GrLivArea  24288.31
## BsmtQual_Ex             22827.59          BsmtQual_Ex  22827.59
## KitchenQual_Ex          22683.09       KitchenQual_Ex  22683.09
## MSZoning_C..all.       -18569.58     MSZoning_C..all.  18569.58
## ExterQual_Ex            18549.75         ExterQual_Ex  18549.75
## Neighborhood_NridgHt    17605.22 Neighborhood_NridgHt  17605.22
## BsmtExposure_Gd         16132.02      BsmtExposure_Gd  16132.02
## Heating_OthW           -15965.41         Heating_OthW  15965.41
## SaleType_New            15335.22          SaleType_New  15335.22
```

```
# Use only top 15 features to refit the model and predict
top15_features <- top15_lasso$Feature
X_top15 <- X_model[, top15_features]

lasso_top15_model <- glmnet(X_top15, y, alpha = 1, lambda = best_lambda_lasso)
pred_top15 <- predict(lasso_top15_model, newx = X_top15)
rmse_top15 <- sqrt(mean((pred_top15 - y)^2))

cat("\n  RMSE using Top 15 Lasso Features:", rmse_top15, "\n")
```

```
##
##   RMSE using Top 15 Lasso Features: 38207.39
```

```
set.seed(123)
top15_features <- top15_lasso$Feature
X_top15 <- X_model[, top15_features]
library(caret)
```

```
## 载入需要的程序包：lattice
```

```
##
## 载入程序包：'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

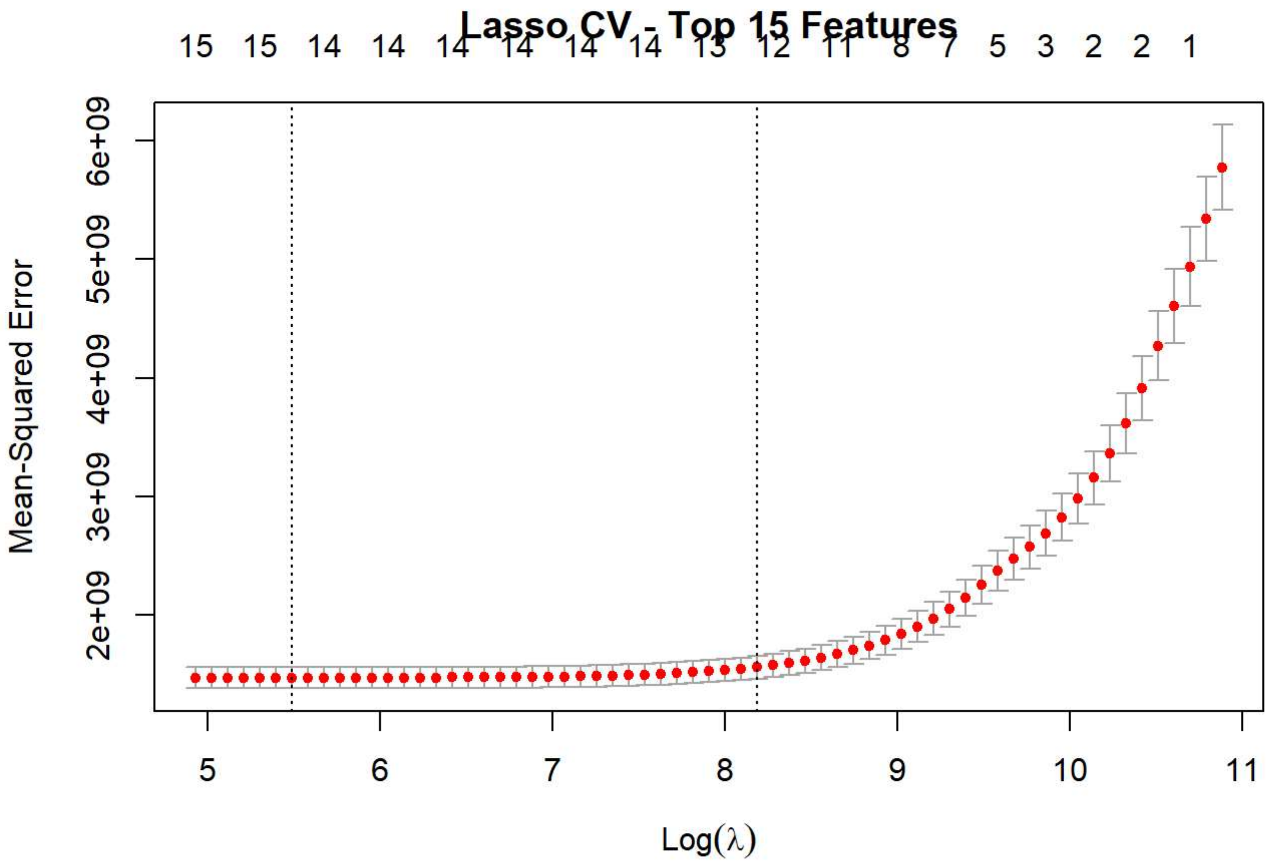
```
trainIndex <- createDataPartition(y, p = 0.8, list = FALSE)
X_train <- X_top15[trainIndex, ]
X_test <- X_top15[-trainIndex, ]
y_train <- y[trainIndex]
y_test <- y[-trainIndex]

### 1. Lasso Regression
lasso_top15_cv <- cv.glmnet(X_train, y_train, alpha = 1)
best_lambda_lasso_top15 <- lasso_top15_cv$lambda.min

lasso_preds_top15 <- predict(lasso_top15_cv, s = best_lambda_lasso_top15, newx = X_test)
lasso_rmse_top15 <- sqrt(mean((lasso_preds_top15 - y_test)^2))

cat("Lasso (Top 15 Features) RMSE:", lasso_rmse_top15, "\n")
```

```
## Lasso (Top 15 Features) RMSE: 46105.87
```

```
cat("Best lambda (Lasso Top 15):", best_lambda_lasso_top15, "\n")
```

```
## Best lambda (Lasso Top 15): 241.4758
```

```
plot(lasso_top15_cv, main = "Lasso CV - Top 15 Features")
```



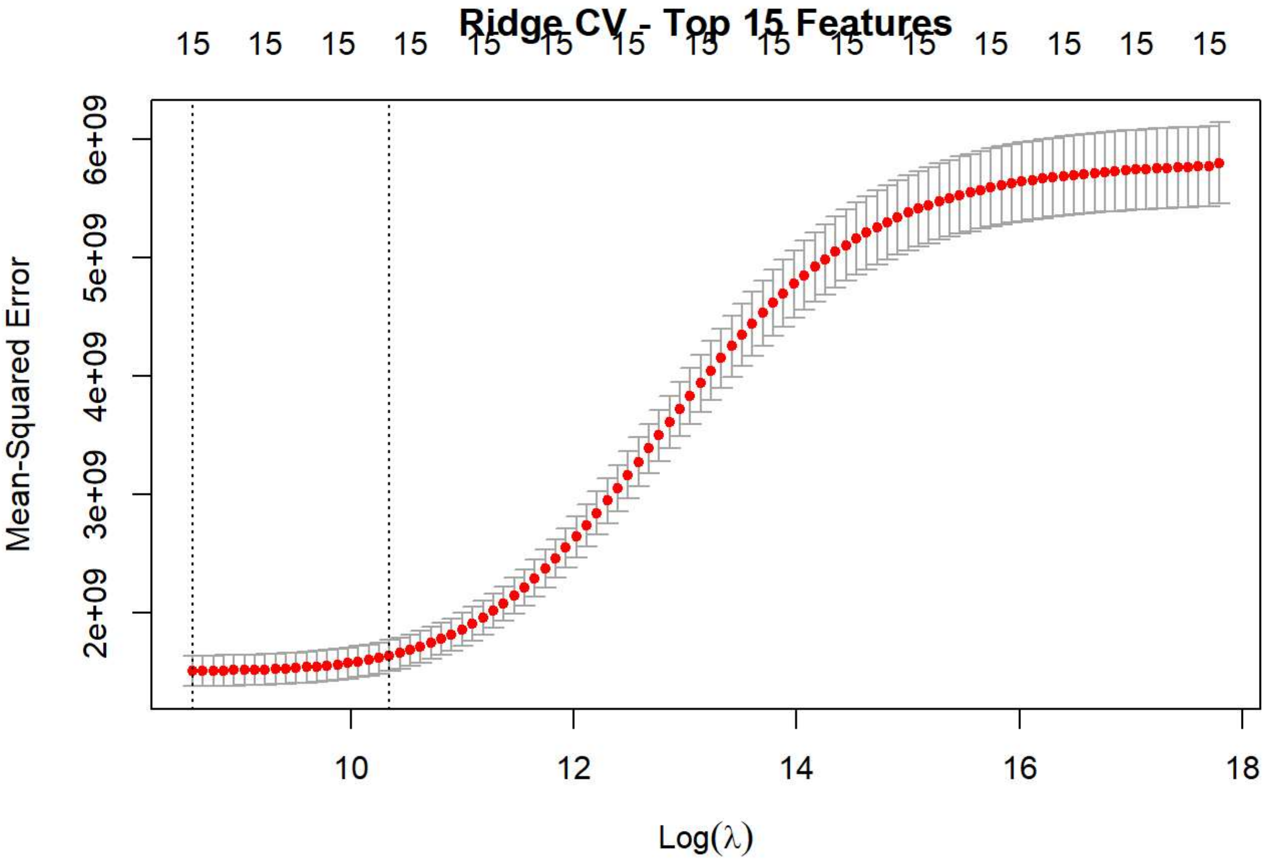
```
### 2. Ridge Regression
ridge_top15_cv <- cv.glmnet(X_train, y_train, alpha = 0)
best_lambda_ridge_top15 <- ridge_top15_cv$lambda.min

ridge_preds_top15 <- predict(ridge_top15_cv, s = best_lambda_ridge_top15, newx = X_test)
ridge_rmse_top15 <- sqrt(mean((ridge_preds_top15 - y_test)^2))

cat("Ridge (Top 15 Features) RMSE:", ridge_rmse_top15, "\n")
```

```
## Ridge (Top 15 Features) RMSE: 45855.31
```

```
cat("Best lambda (Ridge Top 15):", best_lambda_ridge_top15, "\n")
```

```
## Best lambda (Ridge Top 15): 5324.857
```

```
plot(ridge_top15_cv, main = "Ridge CV - Top 15 Features")
```



```
cat("\nFinal Model Comparison (Top 15 features):\n")
```

```
##
## Final Model Comparison (Top 15 features):
```

```
cat("Lasso RMSE:", lasso_rmse_top15, "\n")
```

```
## Lasso RMSE: 46105.87
```

```
cat("Ridge RMSE:", ridge_rmse_top15, "\n")
```

```
## Ridge RMSE: 45855.31
```