cv_lasso_new

Xieyao Yin,Jeremy Liu,Rachel Rubanguka Hoops,Casey Lee

2024-12-02

```
# load packages
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.4.1
```

```
##
## Attaching package: ’dplyr’
```

```
## The following objects are masked from ’package:stats’:
##
##     filter, lag
```

```
## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(rstan)
```

```
## Warning: package ’rstan’ was built under R version 4.4.1
```

```
## Loading required package: StanHeaders
```

```
## Warning: package ’StanHeaders’ was built under R version 4.4.1
```

```
##
## rstan version 2.32.6 (Stan version 2.32.2)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using ‘reduce_sum()‘ or ‘map_rect()‘ Stan functions,
## change ‘threads_per_chain‘ option:
## rstan_options(threads_per_chain = 1)
```

```
## Do not specify ’-march=native’ in ’LOCAL_CPPFLAGS’ or a Makevars file
```

```r
library(bayesplot)
```

```
## Warning: package 'bayesplot' was built under R version 4.4.1
```

```
## This is bayesplot version 1.11.1
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
## Loading required package: lattice
```

```r
library(posterior)
```

```
## Warning: package 'posterior' was built under R version 4.4.1
```

```
## This is posterior version 1.6.0
```

```
##
## Attaching package: 'posterior'
```

```
## The following object is masked from 'package:bayesplot':
##
##     rhat
```

```
## The following objects are masked from 'package:rstan':
##
##     ess_bulk, ess_tail
```

```
## The following objects are masked from 'package:stats':
##
##     mad, sd, var
```

```
## The following objects are masked from 'package:base':
##
##     %in%, match
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.4.1

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.4.2

## Loaded glmnet 4.1-8
```

```r
library(ggplot2)
```

Pre-Processing(should be the same)

```r
cleaned_data <- read.csv("C:\\Users\\Rachel\\Desktop\\final_proj_code_datascie_451\\sampled_data4.csv")

cleaned_data <- cleaned_data %>%
  select(-Description, -Wind_Chill.F., -Start_Lat, -Start_Lng, -City, -No_Exit,
         -County, -State, -Start_Time, -End_Time, -Timezone, -Duration, -Bump, -Traffic_Calming)

binary_columns <- c("Amenity", "Traffic_Signal", "Junction", "Crossing")
cleaned_data[binary_columns] <- lapply(cleaned_data[binary_columns],
                                       function(x) as.integer(factor(x, levels = c("False", "True"), lab

numeric_columns <- c("Temperature.F.", "Humidity...", "Pressure.in.", "Visibility.mi.",
                     "Wind_Speed.mph.", "Precipitation.in.")
cleaned_data[numeric_columns] <- lapply(cleaned_data[numeric_columns], as.numeric)
cleaned_data[numeric_columns] <- scale(cleaned_data[numeric_columns])

categorical_columns <- c("Wind_Direction", "Weather_Condition", "Time_of_Day")
cleaned_data[categorical_columns] <- lapply(cleaned_data[categorical_columns],
                                            function(x) as.integer(factor(x)))

cleaned_data$Severity <- as.factor(cleaned_data$Severity)
levels(cleaned_data$Severity)[levels(cleaned_data$Severity) == "4"] <- "3"
y <- as.numeric(cleaned_data$Severity)
```

Test: Train-Test split, CV, lasso

```r
X <- cleaned_data %>% select(-Severity) %>% as.matrix()

set.seed(123)
cv_lasso <- cv.glmnet(X, y, alpha = 1, family = "multinomial", type.measure = "class")

best_lambda <- cv_lasso$lambda.min
cat("Best Lambda:", best_lambda, "\n")
```

```
## Best Lambda: 0.006184231
```

```
lasso_model <- glmnet(X, y, alpha = 1, lambda = best_lambda, family = "multinomial")

lasso_coefficients <- coef(lasso_model)

significant_predictors <- lapply(lasso_coefficients, function(class_coeff) {
  rownames(class_coeff)[class_coeff[, 1] != 0]
})

cat("Significant Predictors:\n")
```

```
## Significant Predictors:
```

```
print(significant_predictors)
```

```
## $‘1‘
##  [1] ""                  "Distance.mi."      "Temperature.F."
##  [4] "Pressure.in."      "Visibility.mi."    "Wind_Direction"
##  [7] "Wind_Speed.mph."   "Weather_Condition" "Amenity"
## [10] "Crossing"          "Junction"          "Traffic_Signal"
## [13] "Time_of_Day"
##
## $‘2‘
## [1] ""                  "Temperature.F."    "Humidity..."
## [4] "Pressure.in."      "Precipitation.in." "Time_of_Day"
##
## $‘3‘
## [1] ""                  "Visibility.mi."    "Weather_Condition"
## [4] "Amenity"           "Crossing"          "Junction"
## [7] "Traffic_Signal"
```

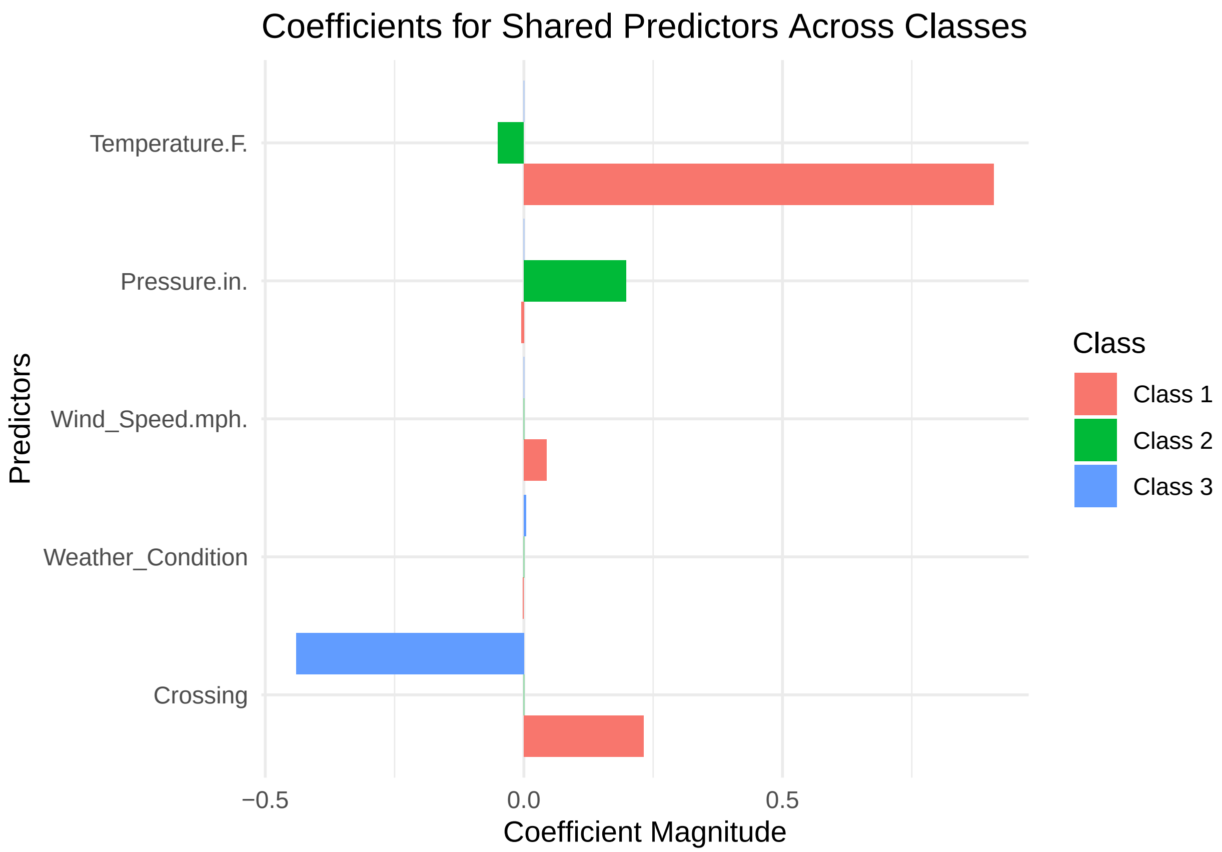
```
cv_misclassification_rate <- cv_lasso$cvm[cv_lasso$lambda == best_lambda]
cat("Cross-Validation Misclassification Rate:", cv_misclassification_rate, "\n")
```

```
## Cross-Validation Misclassification Rate: 0.41625
```

Focus on Shared Predictors: Temperature.F., Pressure.in., Weather_Condition, Wind_Speed.mph., Crossing

```
shared_predictors <- c("Temperature.F.", "Pressure.in.", "Weather_Condition",
                       "Wind_Speed.mph.", "Crossing")

coeff_data <- data.frame()

for (i in seq_along(lasso_coefficients)) {
  class_coeff <- as.matrix(lasso_coefficients[[i]])
  filtered <- class_coeff[rownames(class_coeff) %in% shared_predictors, , drop = FALSE]

  if (nrow(filtered) > 0) {
    temp_data <- data.frame(Predictor = rownames(filtered),
                            Coefficient = filtered[, 1],
                            Class = paste("Class", i))
```

```r
    coeff_data <- rbind(coeff_data, temp_data)
  }
}

if (nrow(coeff_data) > 0) {
  ggplot(coeff_data, aes(x = reorder(Predictor, Coefficient), y = Coefficient, fill = Class)) +
    geom_bar(stat = "identity", position = "dodge") +
    coord_flip() +
    labs(title = "Coefficients for Shared Predictors Across Classes",
         x = "Predictors",
         y = "Coefficient Magnitude",
         fill = "Class") +
    theme_minimal()
} else {
  cat("No coefficients found for the specified shared predictors.\n")
}
```


Coefficients for Shared Predictors Across Classes
Predictors
Temperature.F.
Pressure.in.
Wind_Speed.mph.
Weather_Condition
Crossing
Coefficient Magnitude
Class
Class 1
Class 2
Class 3