flight_delays

Introduction

This vignette demonstrates predictive modeling of flight delays using Ridge Regression with different values of the regularization parameter, `lambda`, to find the optimal model for our dataset.

Step 1: Data Preparation

```
library(AdvanceRAssignment4)
library(nycflights13)
library(dplyr)
```

```
##
## 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(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
# Prepare the data
flights_weather <- flights %>%
  left_join(weather, by = c("year", "month", "day", "hour", "origin")) %>%
  select(dep_delay, arr_delay, temp, humid, wind_speed, visib) %>%
  mutate(
    temp_humid = temp * humid,
    wind_vis = wind_speed * visib
  ) %>%
  filter(!is.na(dep_delay)) %>%
  na.omit()
```

Step 2: Data Splitting

```r
set.seed(1)
train_index <- createDataPartition(flights_weather$dep_delay, p = 0.8, list = FALSE)
train_data <- flights_weather[train_index, ]
temp_data <- flights_weather[-train_index, ]

# Split temp_data into validation and test sets (15% validation, 5% test)
validation_index <- createDataPartition(temp_data$dep_delay, p = 0.75, list = FALSE)
validation_data <- temp_data[validation_index, ]
test_data <- temp_data[-validation_index, ]
```

Step 3: Model Training with Different Lambda Values

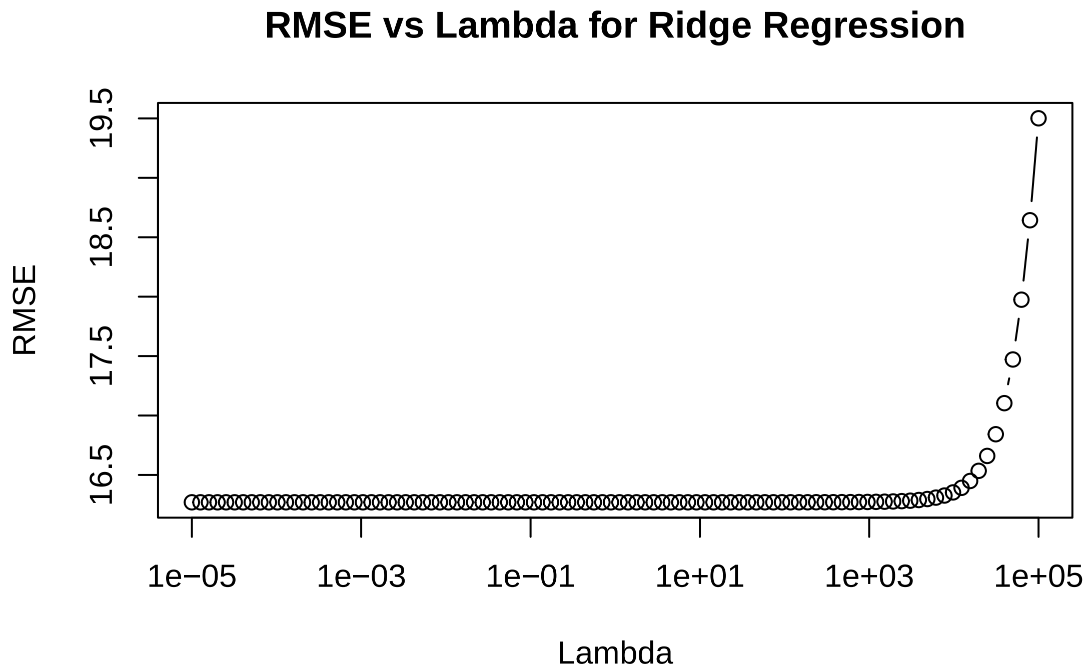
We train models with different lambda values and calculate the Root Mean Squared Error (RMSE) for each on the validation set.

```r
lambdas <- 10^seq(-5, 5, length = 100)
results <- data.frame(lambda = lambdas, RMSE = NA)

for (i in seq_along(lambdas)) {
  model <- ridgereg$new(dep_delay ~ arr_delay + temp + humid + wind_speed + visib + temp_humid + wind_v
                        data = train_data, lambda = lambdas[i])

  coefficients <- model$unScaledCoefficients

  validation_matrix <- model.matrix(dep_delay ~ arr_delay + temp + humid + wind_speed + visib + temp_hum
                                    data = validation_data)

  predictions <- as.vector(validation_matrix %*% coefficients)
  rmse <- sqrt(mean((validation_data$dep_delay - predictions)^2, na.rm = TRUE))
  results$RMSE[i] <- rmse
}

# Show the RMSE values for different lambda values
interval <- 10
results_subset <- results[seq(1, nrow(results), by = interval), ]
results_subset
```

```
##          lambda     RMSE
## 1  1.000000e-05 16.26968
## 11 1.023531e-04 16.26968
## 21 1.047616e-03 16.26968
## 31 1.072267e-02 16.26968
## 41 1.097499e-01 16.26968
## 51 1.123324e+00 16.26968
## 61 1.149757e+01 16.26970
## 71 1.176812e+02 16.26995
## 81 1.204504e+03 16.27351
## 91 1.232847e+04 16.39163
```

Plotting RMSE vs Lambda

```
plot(results$lambda, results$RMSE, type = "b", log = "x", xlab = "Lambda", ylab = "RMSE",
     main = "RMSE vs Lambda for Ridge Regression")
```



Step 4: Find Optimal Lambda and Test Set Evaluation

```
best_lambda <- results$lambda[which.min(results$RMSE)]
cat("Optimal lambda:", best_lambda, "
")
```

```
## Optimal lambda: 1e-05
```

```
final_model <- ridgereg$new(dep_delay ~ arr_delay + temp + humid + wind_speed + visib + temp_humid + wi
                            data = train_data, lambda = best_lambda)

coefficients <- final_model$unScaledCoefficients

test_matrix <- model.matrix(dep_delay ~ arr_delay + temp + humid + wind_speed + visib + temp_humid + wi
                            data = test_data)

final_predictions <- as.vector(test_matrix %*% coefficients)
final_rmse <- sqrt(mean((test_data$dep_delay - final_predictions)^2, na.rm = TRUE))
cat("Final RMSE on test data with optimal lambda:", final_rmse, "
")
```

```
## Final RMSE on test data with optimal lambda: 16.09389
```

This vignette provides a step-by-step process for finding the optimal lambda for ridge regression and evaluating the model on a test dataset, concluding with a plot of RMSE values across different lambdas.