Supervised learning: Regression 2

Contents

Introduction

In this practical, you will learn how to handle many variables with regression by using variable selection techniques, and how to tune hyperparameters for these techniques. This practical has been derived from chapter 6 of ISLR.

One of the packages we are going to use is `glmnet`. For this, you will probably need to `install.packages("glmnet")` before running the `library()` functions.

```
library(ISLR)
library(glmnet)
library(tidyverse)
```

Best subset selection

Our goal for today is to use the `Hitters` dataset from the `ISLR` package to predict `Salary`.

Prepare a dataframe `baseball` from the Hitters dataset but without the baseball players for which the `Salary` is missing. How many baseball players are left?

```
baseball <- Hitters %>% filter(!is.na(Salary))

nrow(baseball)
```

```
## [1] 263
```

Create `baseball_train` (50%), `baseball_valid` (30%), and `baseball_test` (20%) datasets.

```
split <- c(rep("train", 132), rep("valid", 79), rep("test",  52))
baseball <- baseball %>% mutate(split = sample(split))

baseball_train <- baseball %>% filter(split == "train")
```

```r
baseball_valid <- baseball %>% filter(split == "valid")
baseball_test  <- baseball %>% filter(split == "test")
```

Create a function called `lm_mse()` with as input a formula and a training dataset and a test dataset which outputs the mse on the test dataset for predictions from a linear model.

Start like this:

```r
mse <- function(y_true, y_pred) sum((y_true - y_pred)^2)

lm_mse <- function(formula, train_data, valid_data) {
  y_name <- as.character(formula)[2]
  y_true <- valid_data[[y_name]]

  # The remainder of the function here
}
```

```r
mse <- function(y_true, y_pred) sum((y_true - y_pred)^2)

lm_mse <- function(formula, train_data, valid_data) {
  y_name <- as.character(formula)[2]
  y_true <- valid_data[[y_name]]

  lm_fit <- lm(formula, train_data)
  y_pred <- predict(lm_fit, newdata = valid_data)

  mse(y_true, y_pred)
}
```

Try out your function with the formula `Salary ~ Hits + Runs`, using `baseball_train` and `baseball_valid`.

```r
lm_mse(Salary ~ Hits + Runs, baseball_train, baseball_valid)
```

```
## [1] 14443687
```

```r
lm_mse(Salary ~ Hits, baseball_train, baseball_valid)
```

```
## [1] 14347232
```

We have pre-programmed a function for you to output *all* formulas with `p` variables as characters You can load the function into your environment by *sourcing* the `.R` file it is written in: