Simple ML pipeline

Naphon Seeluang

2023-09-22

Classification And REgression Tree => caret

```
library(caret)
```

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

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Simple ML pipeline

subset only columns we want

```
full_df <- mtcars
```

check NA

```
full_df %>%
  complete.cases() %>%
  mean()
```

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

drop rows with NA

```
clean_df <- full_df %>%
  drop_na()
```

1. split data 80% train, 20% test

```
split_data <- function(df) {
    set.seed(42)
    n <- nrow(df)
    train_id <- sample(1:n, size = 0.8*n)
    train_df <- df[train_id, ]
    test_df <- df[-train_id, ]
    # return
    list(training = train_df,
                 testing = test_df)
}

prep_data <- split_data(clean_df)
train_df <- prep_data[[1]]
test_df <- prep_data[[2]]
```

2. train model

```
set.seed(42)
lm_model <- train(mpg ~ .,
                  data = train_df,
                  # ML algorithm
                  method = "lm")

lm_model
```

```
## Linear Regression
##
## 25 samples
## 10 predictors
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 25, 25, 25, 25, 25, 25, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   6.781758  0.4917605  5.285518
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

3. score model

```
p <- predict(lm_model, newdata=test_df)
```

4. evaluate model - actaul(test__df$mpg) compared with prediction(p)

```
# mean absolute error
(mae <- mean(abs(p - test_df$mpg)))
```

```
## [1] 3.963577
```

```r
# root mean square error
(rmse <- sqrt(mean((p - test_df$mpg)**2)))
```

```
## [1] 4.876167
```

```r
# optional - check variable importance
varImp(lm_model)
```

```
## lm variable importance
##
##      Overall
## gear  100.00
## carb   98.57
## wt     73.05
## cyl    67.31
## am     67.19
## qsec   56.81
## drat   44.73
## vs     32.23
## disp   12.19
## hp      0.00
```