hw4

Seongu Lee

5/2/2022

```
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidymodels 0.2.0 --
```

```
## v broom        0.7.12     v recipes      0.2.0
## v dials        0.1.1      v rsample      0.1.1
## v dplyr        1.0.8      v tibble       3.1.5
## v ggplot2      3.3.5      v tidyr        1.2.0
## v infer        1.0.0      v tune         0.2.0
## v modeldata    0.1.1      v workflows    0.2.6
## v parsnip      0.2.1      v workflowsets 0.2.1
## v purrr        0.3.4      v yardstick    0.0.9
```

```
## Warning: package 'broom' was built under R version 4.0.5
```

```
## Warning: package 'dials' was built under R version 4.0.5
```

```
## Warning: package 'scales' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'infer' was built under R version 4.0.5
```

```
## Warning: package 'modeldata' was built under R version 4.0.5
```

```
## Warning: package 'parsnip' was built under R version 4.0.5
```

```
## Warning: package 'purrr' was built under R version 4.0.5
```

```
## Warning: package 'recipes' was built under R version 4.0.5
```

```
## Warning: package 'rsample' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'tune' was built under R version 4.0.5

## Warning: package 'workflows' was built under R version 4.0.5

## Warning: package 'workflowsets' was built under R version 4.0.5

## Warning: package 'yardstick' was built under R version 4.0.5

## -- Conflicts --------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```
#install.packages("ISLR")
#install.packages("ISLR2")
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.5
```

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.0.5

##
## Attaching package: 'ISLR2'

## The following objects are masked from 'package:ISLR':
##
##     Auto, Credit
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages ---------------------------------------- tidyverse 1.3.1 --

## v readr   2.0.2     v forcats 0.5.1
## v stringr 1.4.0

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

```
set.seed(731)
titanic <- read.csv("C:/Users/sungu/OneDrive/Desktop/titanic.csv")
titanic$survived = factor(titanic$survived,levels = c("Yes","No"))
titanic$pclass = factor(titanic$pclass)

split <- initial_split(titanic, prop = 0.80,strata = survived)
train <- training(split)
test<- testing(split)
dim(test)
```

```
## [1] 179  12
```

```
dim(train)
```

```
## [1] 712  12
```

```
reciped <- recipe(survived ~ pclass+sex+age+sib_sp+parch+fare, data = train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(~ starts_with("sex"):fare) %>%
  step_interact(~ age:fare)
reciped
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          6
##
## Operations:
##
## Linear regression imputation for age
## Dummy variables from all_nominal_predictors()
## Interactions with starts_with("sex"):fare
## Interactions with age:fare
```

2,

```
fold <- vfold_cv(train, v = 10)
fold
```

```
## #  10-fold cross-validation
## # A tibble: 10 x 2
##    splits           id
##    <list>           <chr>
##  1 <split [640/72]> Fold01
##  2 <split [640/72]> Fold02
##  3 <split [641/71]> Fold03
##  4 <split [641/71]> Fold04
##  5 <split [641/71]> Fold05
##  6 <split [641/71]> Fold06
##  7 <split [641/71]> Fold07
##  8 <split [641/71]> Fold08
##  9 <split [641/71]> Fold09
## 10 <split [641/71]> Fold10
```

3.

k-Fold Cross-Validation is a strategy to build more efficient model using selected data set. (from https://towardsdatascience.com/k-fold-cross-validation-explained-in-plain-english-659e33c0bc0)

K-Fold cross- Validation has less biased results and less optimistic estimate of the model than simply fitting or entire training set. (From https://machinelearningmastery.com/k-fold-cross-validation/#:~:text=It%20is%20a%20popular%20method,a%20simple%20train%2Ftest%20split.)

Validation set approach will be used for entire training set

```
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
```

```
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(reciped)
```

```
lin_mod <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")
```

```
lin_wkflow <- workflow() %>%
  add_model(lin_mod) %>%
  add_recipe(reciped)
```

```
qd_mod <- discrim_quad() %>%
  set_mode("classification") %>%
  set_engine("MASS")
```

```
qd_wkflow <- workflow() %>%
  add_model(qd_mod) %>%
  add_recipe(reciped)
```

There are 10 folds and 3 models each. So 30 models will be total

5.

```
log_fit <-
  log_wkflow %>%
  fit_resamples(fold)

lin_fit <-
  lin_wkflow %>%
  fit_resamples(fold)

qd_fit <-
  qd_wkflow %>%
  fit_resamples(fold)
```

6.

```
collect_metrics(log_fit)
collect_metrics(lin_fit)
collect_metrics(qd_fit)

(0.790+0.790+0.767)/3
```

logistic regression has highest accuracy and lowest std err.

7.

```
log_fit <- fit(log_wkflow, train)
```

8.

```
log<- bind_cols(predict(log_fit, new_data = test), test%>%dplyr::select(survived))
log_acc <- log %>%
  accuracy(truth = survived, estimate = .pred_class)
log_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.827
```

Accuracy of the model with testing set is 0.8268. And the average accuracy for folds is 0.782. So the model is working better for the testing set. So, model performed well.