Pstat231HW4

Zihao Yang

2022-05-01

```
#install.packages("tidyverse")
#install.packages("tidymodels")
#install.packages("ISLR")
#install.packages("corrr")
#install.packages("discrim")
#install.packages("poissonreg")
#install.packages("klaR")
#install.packages("corrplot")
#install.packages("ggthemes")
#tinytex::install_tinytex()
library(tinytex)
library(tidyverse)
library(tidymodels)
library(ISLR)
library(ggplot2)
library(corrplot)
library(ggthemes)
library(yardstick)
library(dplyr)
library(magrittr)
library(corrr)
library(discrim)
library(poissonreg)
library(klaR)
tidymodels_prefer()
set.seed(100)
```

```
# Get the dataset
tt <- read.csv("titanic.csv")
tt$survived <- factor(tt$survived,levels = c("Yes", "No"))
tt$pclass <- as.factor(tt$pclass)
head(tt)
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                 name    sex age sib_sp parch
```

```
## 1                              Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                               Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                             Allen, Mr. William Henry   male  35      0     0
## 6                                     Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

Q1

```
#Split the data
tt_split <- initial_split(tt, prop = 0.80,
                          strata = survived)
tt_train <- training(tt_split)
tt_test <- testing(tt_split)

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

Q2

```
tt_folds <- vfold_cv(tt_train, v = 10)
degree_grid <- grid_regular(degree(range = c(1, 10)), levels = 10)
```

Q3

The k-fold cross-validation is a resampling method.We randomly divide the data into k groups or folds of roughly equal sizes, and hold out the one of the folds as the validation set and the model is fit on the remaining k-1 folds as if they are the training set. And repeat this process for k times until each of the fold has been treated as the validation set.

We use it because it can achieve a good bias-variance tradeoff on our model, and ensure it to be not overfitting on our training set.

If we use the entire training set, then it should be the bootstrap resampling method.

Q4

```
log_reg <- logistic_reg() %>%
  set_engine("glm")
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(tt_recipe)


lda_mod <- discrim_linear() %>%
  set_engine("MASS")
lda_wkflow = workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(tt_recipe)


qda_mod <- discrim_quad() %>%
  set_engine("MASS")
qda_wkflow = workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(tt_recipe)
```

The total models that I will fit should be 30 models, since one model for each fold, and there are 10 folds in each type of model. Thus, there should be 30 models.

Q5

```
glm_cv <- log_wkflow %>%
  fit_resamples(tt_folds)

lda_cv <- lda_wkflow %>%
  fit_resamples(tt_folds)

qda_cv <- qda_wkflow %>%
  fit_resamples(tt_folds)

save(glm_cv, file = "glm.rda")
save(lda_cv, file = "lda.rda")
save(qda_cv, file = "qda.rda")
```

```
load(file = "glm.rda")
load(file = "lda.rda")
load(file = "qda.rda")
```

Q6

```
collect_metrics(glm_cv)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
```

```
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.803    10  0.0186 Preprocessor1_Model1
## 2 roc_auc  binary     0.848    10  0.0169 Preprocessor1_Model1
```

```
collect_metrics(lda_cv)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.796    10  0.0188 Preprocessor1_Model1
## 2 roc_auc  binary     0.846    10  0.0181 Preprocessor1_Model1
```

```
collect_metrics(qda_cv)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.770    10  0.0116 Preprocessor1_Model1
## 2 roc_auc  binary     0.852    10  0.0168 Preprocessor1_Model1
```

According to the output, the logistic regression model has performed the best, because it has the standard error of 0.018554874 which is not the lowest and it also has the highest mean value of 0.80320814.

Q7

```
best <- fit(log_wkflow, tt_train)
```

Q8

```
tt_predict = predict(best, new_data = tt_test) %>%
  bind_cols(tt_test)
accuracy(tt_predict, truth = survived, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.860
```

The accuracy value is 0.8603352 which is higher than the cross validation mean accuracy of 0.80320814. It implies that the model is good and fits data pretty well.

Q9

$$\frac{\partial}{\partial\hat{\beta}}\sum_{i=1}^{n}(y_i-\hat{\beta})^2=0$$
$$\sum_{i=1}^{n}-2(y_i-\hat{\beta})=0$$
$$\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}\hat{\beta}=0$$
$$\sum_{i=1}^{n}y_i-n\hat{\beta}=0$$
$$\hat{\beta}=\frac{1}{n}\sum_{i=1}^{n}y_i$$
$$\hat{\beta}=\bar{Y}$$

10

For the first fold, we have

$$\hat{\beta}^{(1)}=\frac{1}{n-1}\sum_{i=2}^{n}y_i$$

For the second fold, we have

$$\hat{\beta}^{(2)}=\frac{1}{n-2}(y_1+\sum_{i=3}^{n}y_i)$$

$$Cov(\hat{\beta}^{(1)},\hat{\beta}^{(2)})=Cov(\frac{1}{n-1}\sum_{i=2}^{n}y_i,\frac{1}{n-2}(y_1+\sum_{i=3}^{n}y_i))$$
$$=\frac{1}{n-1}Cov(\sum_{i=2}^{n}y_i,y_1+\sum_{i=3}^{n}y_i)$$

For $i=j$ we have$Cov(y_i,y_j)=var(y_i)$

For $i\neq j$, we have $Cov(y_i,y_j)=0$

Thus,

$$Cov(\hat{\beta}^{(1)},\hat{\beta}^{(2)})=\frac{1}{n-1}Cov(\sum_{i=2}^{n}y_i,y_1+\sum_{i=3}^{n}y_i)$$
$$=\frac{(n-2)\sigma^2}{n-1}$$