PSTAT231 HW4 Cheng Ye

Cheng Ye

2022-11-01

```
titanic <-read.csv("C:/Cheng Ye/UCSB/PSTAT 231/HW/homework-4/homework-4/data/titanic.csv") %>%
  mutate(survived = factor(survived,
                           levels = c("Yes", "No")),
         pclass = factor(pclass))
# Loading required dataset
head(titanic) # visualize data set
```

```
##   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
```

```
set.seed(231) # could be any number
```

Question 1

```
titanic_split <- initial_split(titanic, prop = 0.80, strata = survived) # split the data and stratified on survived,train 8
0%, test 20%
titanic_train <- training(titanic_split) # training the dataset
titanic_test <- testing(titanic_split) # testing the dataset
dim(titanic_train)
```

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

```
dim(titanic_test)
```

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

```
#Create a recipe identical to the recipe in HW3
titanic_survived_recipe <- recipe(survived ~ pclass + sex + age + sib_sp + parch + fare, titanic_train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~ starts_with('sex'):fare+age:fare)
summary(titanic_survived_recipe)
```

```
## # A tibble: 7 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
## 1 pclass   nominal predictor original
## 2 sex      nominal predictor original
## 3 age      numeric predictor original
## 4 sib_sp   numeric predictor original
## 5 parch    numeric predictor original
## 6 fare     numeric predictor original
## 7 survived nominal outcome   original
```

Question 2

```
survived_fold = vfold_cv(titanic_train, v=10)
survived_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
```

Question 3

```
#We are training the data by spliting it into 10 folds to evaluate the model's ability when given new data.

#The K-fold Cross-Validation is a method we use to estimate skill of machine learning models, the method is that it split th
e dataset into K number of folds and is used to evaluate the model's ability when given new data.

#Using K-fold Cross-Validation method instead of simple fit the model helps us avoid overfitting and gives the model the opp
ortunity to train on multiple train-test splits.

#If we did use the entire training set the re-sampling method would be LOOCV (leave-one-out cross validation).
```

Question 4

```
#For logistic regression model
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_survived_recipe)
log_fit <- fit(log_wkflow, titanic_train)

#For linear discriminant analysis
lda_mod <- discrim_linear() %>%
  set_engine('MASS') %>%
  set_mode('classification')
lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_survived_recipe)
lda_fit <- fit(lda_wkflow, titanic_train)

#For quadratic discriminant analysis
qda_mod <- discrim_quad() %>%
  set_engine('MASS') %>%
  set_mode('classification')
qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_survived_recipe)
qda_fit <- fit(qda_wkflow, titanic_train)
```

Question 5

```
log_fit<-fit_resamples(log_wkflow,survived_fold)
lda_fit<-fit_resamples(lda_wkflow,survived_fold)
qda_fit<-fit_resamples(qda_wkflow,survived_fold)
```

Question 6

```
log_metrics <- collect_metrics(log_fit)
lda_metrics <- collect_metrics(lda_fit)
qda_metrics <- collect_metrics(qda_fit)

log_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.807    10  0.0162 Preprocessor1_Model1
## 2 roc_auc  binary     0.842    10  0.0155 Preprocessor1_Model1
```

```
lda_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.799    10  0.0130 Preprocessor1_Model1
## 2 roc_auc  binary     0.840    10  0.0151 Preprocessor1_Model1
```

```
qda_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.778    10 0.00870 Preprocessor1_Model1
## 2 roc_auc  binary     0.842    10 0.0128  Preprocessor1_Model1
```

```
#Based on the results, we could observe that the Logistic regression is the best model in this case because Logistic regress
ion method has highest accuracy among the three models and the second lowest standard error for accuracy among the three mod
els, hence it is the most accurate in this case
```

Question 7

```
log_fit_whole<-fit(log_wkflow,titanic_train)
log_fit_whole
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 3 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
##
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)
##
## Coefficients:
##     (Intercept)              age           sib_sp            parch
##      -4.4250232        0.0619750        0.3309583        0.1461365
##            fare        pclass_X2        pclass_X3         sex_male
##       0.0051718        1.1340776        2.3152447        2.4263702
## sex_male_x_fare       fare_x_age
##       0.0090827       -0.0004124
##
## Degrees of Freedom: 711 Total (i.e. Null);  702 Residual
## Null Deviance:      948
## Residual Deviance: 630.8     AIC: 650.8
```

Question 8

```
log_prediction <- predict(log_fit_whole, new_data = titanic_test, type = "class")
bind_cols(log_prediction,titanic_test$survived)
```

```
## New names:
## * `` -> `...2`
```

```
## # A tibble: 179 x 2
##    .pred_class ...2
##    <fct>       <fct>
##  1 No          No
##  2 Yes         Yes
##  3 No          No
##  4 No          Yes
##  5 No          Yes
##  6 No          Yes
##  7 Yes         Yes
##  8 No          Yes
##  9 No          No
## 10 Yes         No
## # ... with 169 more rows
```

```
train_accuracy <- augment(log_fit_whole, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
train_accuracy
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.810
```

```
test_accuracy <- augment(log_fit_whole, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)
test_accuracy
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.804
```

```
#By observing the results we know that the test accuracy is 0.7374302 while the train accuracy is 0.8286517, so the training accuracy is higher than that of the testing accuracy.
```

Question 9

$$Since\ Y=\beta+\epsilon, \epsilon \sim N(0,\sigma^2), we\ have\ that$$
$$\mathrm{RSS}(\beta)=\sum_{i=1}^{N}\left(y_i-x_i^\top\beta\right)^2, then$$
$$\mathrm{RSS}(\beta)=\sum_{i=1}^{N}\left(y_i-x_i^\top\beta\right)^2=(y-x\beta)^\top(y-x\beta)$$
$$By\ differentiating\ it\ with\ respect\ to\ \beta, we\ have\ that$$
$$x^\top(y-x\beta)=0$$
$$So\ we\ get\ that \hat{\beta}=\left(x^\top x\right)^{-1}x^\top y$$

Question 10

$$\mathbf{Cov}(\hat{\beta}_1,\hat{\beta}_2)=\left(X^TX\right)^{-1}X^T\left(\sigma^2 I\right)\left(\left(X^TX\right)^{-1}X^T\right)^T=\sigma^2\left(X^TX\right)^{-1}X^T\left(\left(X^TX\right)^{-1}X^T\right)^T=\sigma^2\left(X^TX\right)^{-1}X^TX\left(X^TX\right)^{-1}=\sigma^2\left(X^TX\right.$$