Homework Assignment 4

Vasiliy Ostapenko (774 970 8)

April 21, 2022

Load Data

```r
DATA_FOLDER = "./data"
IMAGES_FOLDER = "./images"
TITANIC_FNAME = file.path(DATA_FOLDER, "titanic.csv")
data = read.csv(TITANIC_FNAME)
data$survived = as.factor(data$survived)
data$pclass = as.factor(data$pclass)
```

Question 1

```r
titanic_split = data %>%
  initial_split(prop=0.8, strata="survived")

titanic_train = training(titanic_split)
titanic_test = testing(titanic_split)
```

```r
titanic_recipe = recipe(survived ~ pclass+sex+age+sib_sp+parch+fare,
                        data=titanic_train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact( ~ starts_with("sex"):fare) %>%
  step_interact( ~ age:fare)
```

Question 2

```r
titanic_folds = vfold_cv(titanic_train, v=10)
```

Question 3

K-fold cross validation is a method of cross validation in which the training set is split into K partitions. We then iterate over each partition, using it as the validation set while every other partition is included in the training set. This allows us to test model architecture and optimize hyperparameters without letting the model see the actual test data, whcih would be considered cheating. Using any type of cross validation will also speed up the model training process and will also allow the model to generalize better (prevents overfitting). Using the entire training set is essentially 1-Fold CV.

Question 4

3 models across 10 folds means there will be a total of 30 models fit to some subset of the training data.

```r
# Logistic Reg
glm_model = logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

glm_workflow = workflow() %>%
  add_model(glm_model) %>%
  add_recipe(titanic_recipe)

# LDA
lda_model = discrim_linear() %>%
  set_engine("MASS") %>%
  set_mode("classification")

lda_workflow = workflow() %>%
  add_model(lda_model) %>%
  add_recipe(titanic_recipe)

# QDA
qda_model = discrim_quad() %>%
  set_engine("MASS") %>%
  set_mode("classification")

qda_workflow = workflow() %>%
  add_model(qda_model) %>%
  add_recipe(titanic_recipe)
```

Question 5

```r
glm_tune = glm_workflow %>%
  tune_grid(resamples=titanic_folds)

lda_tune = lda_workflow %>%
  tune_grid(resamples=titanic_folds)

qda_tune = qda_workflow %>%
  tune_grid(resamples=titanic_folds)

save(glm_tune, glm_workflow, file="./data/glm_tune.rda")
save(lda_tune, lda_workflow, file="./data/lda_tune.rda")
save(qda_tune, qda_workflow, file="./data/qda_tune.rda")
```

Question 6

```r
glm_metrics = collect_metrics(glm_tune)
lda_metrics = collect_metrics(lda_tune)
qda_metrics = collect_metrics(qda_tune)
```

```r
print(glm_metrics[glm_metrics[".metric"]=="accuracy",
                  c("mean", "std_err")] %>% unlist())
```

```
##    mean std_err
##  0.8006  0.0168
```

```
print(lda_metrics[lda_metrics[".metric"]=="accuracy",
                  c("mean", "std_err")] %>% unlist())
```

```
##    mean std_err
## 0.78793 0.01503
```

```
print(qda_metrics[qda_metrics[".metric"]=="accuracy",
                  c("mean", "std_err")] %>% unlist())
```

```
##    mean std_err
##  0.7711  0.0183
```

Using the accuracy criterion across 10 folds, we see that the logistic regression model performed the best. It has the highest mean accuracy relative to the other two models while the standard error is about in line with the standard errors of the two other model accuracies.

Question 7

```
glm_fit = glm_workflow %>%
  fit(titanic_train)
```

Question 8

```
bound_test_data = bind_cols(predict(glm_fit, titanic_test),
                            titanic_test$survived)
colnames(bound_test_data) = c("GLM Predict", "True")
print(accuracy(bound_test_data,
               truth="True", estimate="GLM Predict")$.estimate)
```

```
## [1] 0.838
```

Test set accuracy of 83.8% is slightly higher, but in line with, the average validation accuracy of 80.1%. This is probably due to random chance in the way the data was split.