Homework 3

Code

Syed Immad Ali

Question 1

Split the data, stratifying on the outcome variable, `survived`. You should choose the proportions to split the data into. Verify that the training and testing data sets have the appropriate number of observations. Take a look at the training data and note any potential issues, such as missing data.

Why is it a good idea to use stratified sampling for this data?

It is important to used stratified sampling because there could be significant differences in the those who survived. Using stratified sampling helps make sure they data is similar with each other

Hide

```
titanic <-read.csv("titanic.csv")


#Set Seed
set.seed(2000)

#Split, train and test
titanic_split <- initial_split(titanic, prop = 0.7, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

We now verify all observations are accounted for and check for missing calues. We can see all observations are accounted for. There are missing data observations as seen inthe last line.

Hide

```
nrow(titanic)
```

```
## [1] 891
```

Hide

```
nrow(titanic_train)
```

```
## [1] 623
```

Hide

```
nrow(titanic_test)
```

```
## [1] 268
```

Hide

```
table(is.na(titanic_train))
```

```
##
## FALSE  TRUE
##  6877   599
```

Question 2

Using the **training** data set, explore/describe the distribution of the outcome variable `survived`.

Hide

```
survived <- factor(titanic_train$survived)
survivedCount <- table(survived)

propSurvived <- prop.table(survivedCount)
propSurvived
```

```
## survived
##        No       Yes
## 0.6163724 0.3836276
```

We can see that about 38.36% of all passengers survived the shipwreck. We will next look at those who survived based on what passenger class they were.

Hide

```
survivedClass <- table(survived,titanic_train$pclass)
survivedClass
```

```
##
## survived   1   2   3
##       No  54  71 259
##       Yes 100  61  78
```

From the table, we can see of those who survived, most of them were from the higher classes, about 100. This is likely due to the fact the higher class passengers were higher up on the ship.

Question 3

Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

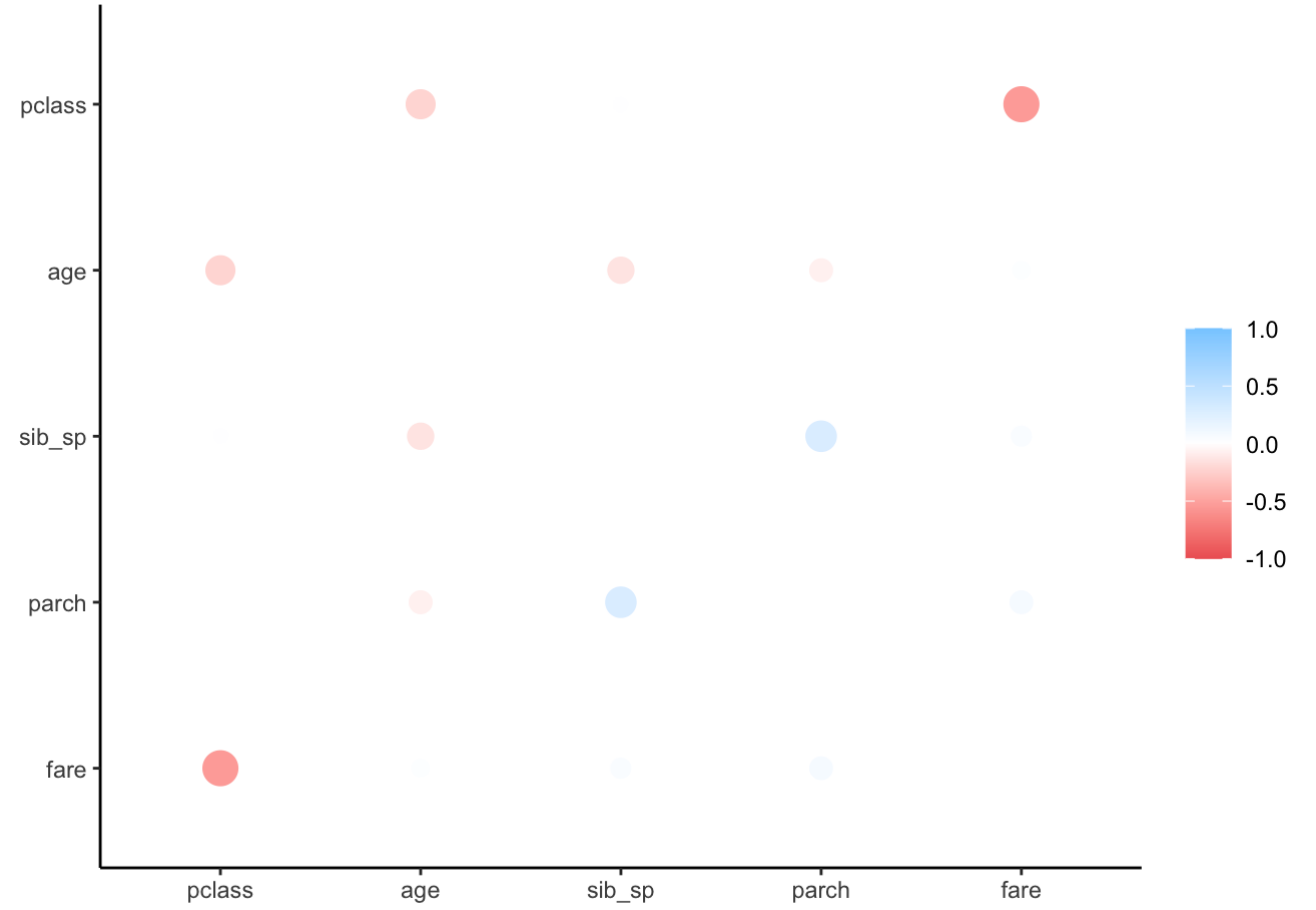
Hide

```
# Only select variables where age is not missing
titanic_trainNum <- titanic_train %>% select(where(is.numeric))

#Deselect passenger ID
titanic_trainNum <-subset(titanic_trainNum, select = -c(passenger_id))

correlation_titanic_train <- titanic_trainNum %>% correlate()

rplot(correlation_titanic_train)
```



Looking at the plot, any variables that lie above the diagonal are negatively correlated, while the variables below the diagonal are positively correlated. Age and PClass are negatively correlated, along with other variables like age and parch. In terms of positively correlated variable,s we can see fare and sib_sp are positively correlated.

Question 4

Using the **training** data, create a recipe predicting the outcome variable `survived`. Include the following predictors: ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare.

Hide

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

Question 5

Specify a **logistic regression** model for classification using the `"glm"` engine. Then create a workflow. Add your model and the appropriate recipe. Finally, use `fit()` to apply your workflow to the **training** data.

Hide

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

log_wkflw <-workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

log_fit <- fit(log_wkflw, titanic_train)

log_fit %>%
  tidy()
```

```
## # A tibble: 9 × 5
##   term             estimate std.error statistic  p.value
##   <chr>               <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      5.19      0.787        6.60  4.09e-11
## 2 pclass          -1.14      0.180       -6.35  2.19e-10
## 3 age             -0.0480    0.0130      -3.69  2.20e- 4
## 4 sib_sp          -0.407     0.132       -3.08  2.10e- 3
## 5 parch           -0.305     0.152       -2.01  4.41e- 2
## 6 fare             0.00421   0.0110       0.383 7.02e- 1
## 7 sex_male        -2.49      0.318       -7.84  4.42e-15
## 8 sex_male_x_fare -0.0114    0.00876     -1.30  1.92e- 1
## 9 age_x_fare       0.000277  0.000195     1.42  1.54e- 1
```

Question 6

Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the `"MASS"` engine.

Hide

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

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_recipe)

lda_fit <- fit(lda_wkflow, titanic_train)
```

Question 7

Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the `"MASS"` engine.

Hide

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

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_recipe)

qda_fit <- fit(qda_wkflow, titanic_train)
```

Question 8

Repeat Question 5, but this time specify a naive Bayes model for classification using the `"klaR"` engine. Set the `usekernel` argument to `FALSE`.

Hide

```
nb_mod <- naive_Bayes() %>%
  set_mode("classification") %>%
  set_engine("klaR") %>%
  set_args(usekernel = FALSE)

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(titanic_recipe)

nb_fit <- fit(nb_wkflow, titanic_train)
```

Question 9

Now you've fit four different models to your training data.

Logistic Regression

Hide

```
logisticReg <- predict(log_fit, new_data = titanic_train, type = "prob")

logisticReg <-bind_cols(logisticReg, titanic_train %>% select(survived))

logisticRegAug <- augment(log_fit, new_data = titanic_train) %>%
  accuracy(truth = factor(survived), estimate = .pred_class)
```

IDA

Hide

```
ldapred <- predict(lda_fit, new_data = titanic_train, type = "prob")

ldapred <-bind_cols(ldapred, titanic_train %>%select(survived))

ldaacc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = factor(survived), estimate = .pred_class)
```

QDA

Hide

```
qdaPred <- predict(qda_fit, new_data = titanic_train, type = "prob")

qdaPred<-bind_cols(qdaPred, titanic_train %>%select(survived))

qdaAcc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = factor(survived), estimate = .pred_class)
```

Naive

Hide

```
#Naive Bayesian
nbPred <- predict(nb_fit, new_data = titanic_train, type = "prob")

nbPred<-bind_cols(nbPred, titanic_train%>%select(survived))

nbAcc <- augment(nb_fit, new_data = titanic_train) %>%
  accuracy(truth = factor(survived), estimate = .pred_class)
```

Hide

```
accuracies <-c(logisticReg$.estimate, ldaacc$.estimate,
               qdaAcc$.estimate, nbAcc$.estimate)

models <- c("Logistic Regression", "LDA", "QDA")

results <- tibble(accuracies = accuracies, models = models)

results %>%
  arrange(-accuracies)
```

```
## # A tibble: 3 × 2
##   accuracies models
##        <dbl> <chr>
## 1      0.799 Logistic Regression
## 2      0.798 LDA
## 3      0.777 QDA
```

The logisitic model seems to have performed the best on the training set, accuracy of .7993.

Question 10

Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

Hide

```
# PREDICTED VALUES AND ACCURACY
logReg_pred_test <- predict(log_fit, new_data = titanic_test, type = "prob")
head(logReg_pred_test)
```

```
## # A tibble: 6 × 2
##   .pred_No .pred_Yes
##      <dbl>     <dbl>
## 1   0.0627    0.937
## 2   0.891     0.109
## 3   0.923     0.0768
## 4   0.278     0.722
## 5   0.141     0.859
## 6   0.238     0.762
```

Hide

```
logReg_pred_test_acc <- augment(log_fit, new_data = titanic_test) %>% accuracy(truth = factor(survived), estimate = .pred_class)
logReg_pred_test_acc
```

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.799
```

Hide

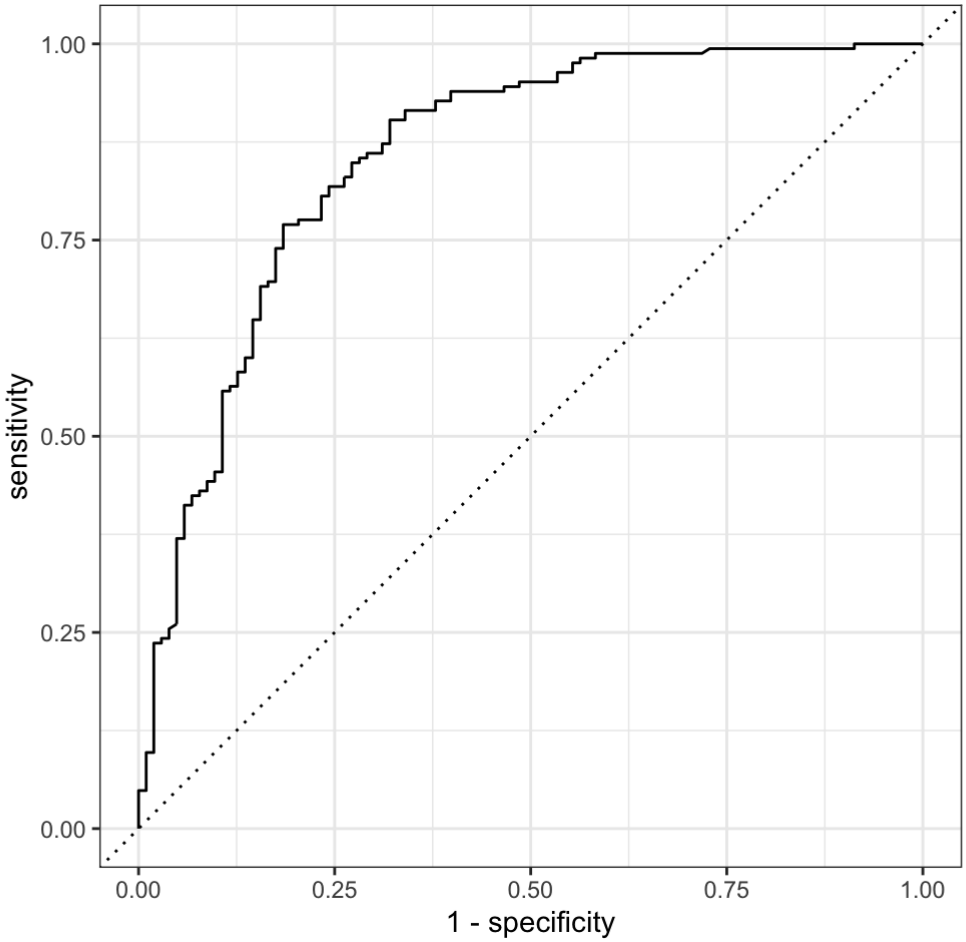
```
augment(log_fit, new_data = titanic_test) %>% conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction  No Yes
##        No  144  33
##        Yes  21  70
```

Using ROC Curve

Hide

```
augment(log_fit, new_data = titanic_test) %>%
  roc_curve(factor(survived), .pred_No) %>%
  autoplot()
```



Hide

```
CurveArea <- augment(log_fit, new_data = titanic_test) %>%
  roc_auc(factor(survived), .pred_No)

CurveArea
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.855
```

From the data, the accuracy seems to be greater on the test data, rather than the training data, thus we can say the model is not a good fit.