M445 HW 7

Erick Castillo

4/2/2021

The following chunk of code displays how I prepared the Titanic dataset for this assignment.

```
df = Titanic <- read.csv("~/M445_HW/Titanic.csv")

df = subset(df, select = c('Pclass', 'Sex', 'Age',
                           'SibSp', 'Parch', 'Fare',
                           'Embarked', 'Survived'))
summary(df) #177 missing values come from Age

df$Sex <- ifelse(df$Sex == 'male', 1, 0)
df$Age <- ifelse(is.na(df$Age), round(mean(df$Age, na.rm = TRUE)), round(df$Age))
df = df[!(df$Embarked==""), ] #drop empty strings
df$Embarked <- as.factor(df$Embarked)

#train-test split of the data:
smp_size <- floor(0.70 * nrow(df))
set.seed(505)
train_ind <- sample(seq_len(nrow(df)), size = smp_size)

train <- df[train_ind, ]
test <- df[-train_ind, ]
```

A. Build a classification model using LDA.

```
lda.mod = lda((Survived~.), data = train)
lda.pred <- lda.mod %>% predict(test)
compare1 = data.frame(test$Survived, lda.pred$class)
confusionMatrix(as.factor(lda.pred$class), as.factor(test$Survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 139  27
##          1  24  77
##
##                Accuracy : 0.809
##                  95% CI : (0.7566, 0.8543)
##     No Information Rate : 0.6105
##     P-Value [Acc > NIR] : 2.3e-12
##
##                   Kappa : 0.5963
##
##  Mcnemar's Test P-Value : 0.7794
##
##             Sensitivity : 0.8528
##             Specificity : 0.7404
##          Pos Pred Value : 0.8373
##          Neg Pred Value : 0.7624
##              Prevalence : 0.6105
##          Detection Rate : 0.5206
##    Detection Prevalence : 0.6217
##       Balanced Accuracy : 0.7966
##
##        'Positive' Class : 0
##
```

The above is the output for the confusion matrix associated with the LDA model. It has a relatively high accuracy.

B. Create a classification model with QDA. Compare these results with the LDA confusion matrix output.

```
qda.mod = qda((Survived~.), data = train)
qda.pred <- predict(qda.mod, test)
compare2 = data.frame(test$Survived, qda.pred$class)
confusionMatrix(as.factor(qda.pred$class), as.factor(test$Survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 137  28
##          1  26  76
##
```

```
##                Accuracy : 0.7978
##                  95% CI : (0.7445, 0.8443)
##     No Information Rate : 0.6105
##     P-Value [Acc > NIR] : 4.162e-11
##
##                   Kappa : 0.5733
##
##  Mcnemar's Test P-Value : 0.8918
##
##             Sensitivity : 0.8405
##             Specificity : 0.7308
##          Pos Pred Value : 0.8303
##          Neg Pred Value : 0.7451
##              Prevalence : 0.6105
##          Detection Rate : 0.5131
##    Detection Prevalence : 0.6180
##       Balanced Accuracy : 0.7856
##
##        'Positive' Class : 0
##
```

Utilizing a QDA model lowers the accuracy when compared to the LDA model. The p-value of the McNemar test increased in this case compared to the LDA model. The same conclusion would be reached regardless, but the strength of the evidence in favor of the H_0 increased.

C. Create a classification model using logistic regression analysis.

```
log.mod <- glm(Survived~., family=binomial(link = 'cloglog'), data = train)
pred1 <- predict.glm(log.mod, newdata = test, type = 'response')
pred1 <- ifelse(pred1 < 0.57, 0, 1)
confusionMatrix(as.factor(pred1), as.factor(test$Survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 151  35
##          1  12  69
##
##                Accuracy : 0.824
##                  95% CI : (0.7729, 0.8677)
##     No Information Rate : 0.6105
##     P-Value [Acc > NIR] : 3.466e-14
##
##                   Kappa : 0.6144
##
##  Mcnemar's Test P-Value : 0.001332
##
##             Sensitivity : 0.9264
##             Specificity : 0.6635
##          Pos Pred Value : 0.8118
##          Neg Pred Value : 0.8519
##              Prevalence : 0.6105
```

```
##          Detection Rate : 0.5655
##    Detection Prevalence : 0.6966
##       Balanced Accuracy : 0.7949
##
##        'Positive' Class : 0
##
```

The accuracy in this case is slightly higher than both those seen in LDA and QDA. This indicates this is a slightly better model at predicting the survival of passengers.

D. Create predictions through the use of a random forest.

```
rf_classifier = randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, da

test_set_predictors = select(test, -Survived)
prediction_for_table <- predict(rf_classifier, test_set_predictors)
confusionMatrix(as.factor(test$Survived), as.factor(prediction_for_table))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 146  17
##          1  27  77
##
##                Accuracy : 0.8352
##                  95% CI : (0.7852, 0.8776)
##     No Information Rate : 0.6479
##     P-Value [Acc > NIR] : 8.447e-12
##
##                   Kappa : 0.6474
##
##  Mcnemar's Test P-Value : 0.1748
##
##             Sensitivity : 0.8439
##             Specificity : 0.8191
##          Pos Pred Value : 0.8957
##          Neg Pred Value : 0.7404
##              Prevalence : 0.6479
##          Detection Rate : 0.5468
##    Detection Prevalence : 0.6105
##       Balanced Accuracy : 0.8315
##
##        'Positive' Class : 0
##
```

The above output states that the random forest model has an accuracy of 83.52%. This is slightly better than the logistic model and both LDA and QDA. This would be my preferred method of classification.