Lab Assignment #7

Math 437 - Modern Data Analysis

Due March 24, 2023

Instructions

The purpose of this lab is to introduce several different classification strategies and variations on classification accuracy. In this lab we will work with another staple set of strategies: Naive Bayes and linear/quadratic discriminant analysis.

```
library(ISLR2)
library(ggplot2)
library(dplyr)
library(nycflights13)
library(e1071) # Naive Bayes
library(MASS) # LDA/QDA
library(yardstick) # only tidymodels package we'll need in this lab
```

This lab assignment is worth a total of **25 points**.

Problem 1: Naive Bayes

Part a (Code: 1 pt)

Run the code in ISLR Lab 4.7.5.

```
attach(Smarket)
train <- (Smarket$Year <2005)
Smarket.2005 <- Smarket[!train,]
Direction.2005 <- Direction[!train]
library(e1071)
```

```
nb.fit <- naiveBayes(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)


nb.fit
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     Down       Up
## 0.491984 0.508016
##
```

```
## Conditional probabilities:
##       Lag1
## Y             [,1]     [,2]
##   Down  0.04279022 1.227446
##   Up   -0.03954635 1.231668
##
##       Lag2
## Y             [,1]     [,2]
##   Down  0.03389409 1.239191
##   Up   -0.03132544 1.220765
```

```
mean(Lag1[train][Direction[train] == "Down"])
```

```
## [1] 0.04279022
```

```
sd(Lag1[train][Direction[train] == "Down"])
```

```
## [1] 1.227446
```

```
nb.class <- predict(nb.fit,Smarket.2005)

table(nb.class, Direction.2005)
```

```
##         Direction.2005
## nb.class Down  Up
##     Down   28  20
##     Up     83 121
```

```
mean(nb.class == Direction.2005)
```

```
## [1] 0.5912698
```

```
nb.preds <- predict(nb.fit, Smarket.2005, type = "raw")
nb.preds[1:5,]
```

```
##           Down        Up
## [1,] 0.4873164 0.5126836
## [2,] 0.4762492 0.5237508
## [3,] 0.4653377 0.5346623
## [4,] 0.4748652 0.5251348
## [5,] 0.4901890 0.5098110
```

Part b (Code: 1 pt)

Recall that in Lab 6 we filtered the flights to only the United, American, and Delta carriers. Here we add another variable, `arr_ontime`, to represent whether the flight arrived on time or not.

```
flights2 <- flights %>% filter(
  carrier %in% c("UA", "AA", "DL"),
  !is.na(dep_delay),
  !is.na(arr_delay)
) %>%
  mutate(
    arr_ontime = as.factor(if_else(arr_delay <= 0, "yes", "no"))
  )
```

Non-randomly divide the `flights2` dataset into `flights_training`, which contains all flights through October, and `flights_test`, which contains all flights in November and December. You should be able to use the `filter` function to do this.

```r
flights2$arr_ontime <- relevel(flights2$arr_ontime, ref = "yes")

flights_training <- flights2 %>%
  filter(month == 10)

flights_test <- flights2 %>%
  filter(month == 11 | month == 12)
```

Then, fit a Naive Bayes model on the training set predicting whether a flight will be delayed (`arr_ontime` = "no") based on the departure delay (`dep_delay`), `carrier`, `distance` traveled, and `origin`.

```r
nb.flit <- naiveBayes(arr_ontime ~ dep_delay + carrier + distance + origin,
                      data = flights_training)
```

Part c (Code: 1.5 pts)

Unfortunately, there is no easy way to use `augment` on this model, so we'll have to make the predictions ourselves.

First, make class predictions on the `flights_test` dataset using similar code to that done in Lab 4.7.5. Then, create the `flights_nb_predictions` data frame or tibble containing two columns: `predicted`, representing the predicted classes, and `actual`, representing the actual classes. Use `flights_nb_predictions` to obtain the confusion matrix for the model.

```r
nb.flass <- predict(nb.flit, flights_test)

flights_nb_predictions <- data.frame(
  predicted = nb.flass,
  actual = flights_test$arr_ontime
)

(cmtrx <- conf_mat(flights_nb_predictions, truth = actual,
         estimate = predicted))
```

```
##            Truth
## Prediction   yes    no
##        yes 12787  5043
##        no    662  4206
```

Part d (Code: 0.5 pts; Explanation: 2 pts)

Without running any additional code, use the confusion matrix from part (c) to estimate the sensitivity, specificity, positive predictive value, and negative predictive value for the model. Express all answers as fractions and then convert to decimals rounded to the thousandths place (3 decimal places).

Sensitivity = 12787/13449 0.951

Specificity = 4206/9249 0.455

Positive Predictive Value = 12787/17830 0.717

Negative Preidctive Value = 4206/4868 0.864

Then, using the `summary` function on your confusion matrix, check your answers. Remember that we are trying to predict that a flight will be delayed (`arr_ontime` = "no").

```r
summary(cmtrx)
```

```
## # A tibble: 13 x 3
##    .metric              .estimator .estimate
##    <chr>                <chr>          <dbl>
##  1 accuracy             binary         0.749
##  2 kap                  binary         0.438
##  3 sens                 binary         0.951
##  4 spec                 binary         0.455
##  5 ppv                  binary         0.717
##  6 npv                  binary         0.864
##  7 mcc                  binary         0.485
##  8 j_index              binary         0.406
##  9 bal_accuracy         binary         0.703
## 10 detection_prevalence binary         0.786
## 11 precision            binary         0.717
## 12 recall               binary         0.951
## 13 f_meas               binary         0.818
```

Problem 2: Discriminant Analysis

Part a (Code: 1 pt)

Run the code in ISLR Lab 4.7.3.

Part b (Code: 1 pt)

Run the code in ISLR Lab 4.7.4.

Part c (Code: 1 pt)

Fit a LDA model on the training set predicting whether a flight will be delayed (`arr_ontime` = "no") based on the departure delay (`dep_delay`), `carrier`, `distance` traveled, and `origin`.

Part d (Code: 1.5 pts)

Unfortunately, there is no easy way to use `augment` on this model, so we'll have to make the predictions ourselves.

First, make class predictions on the `flights_test` dataset using similar code to that done in Lab 4.7.3. Then, create the `flights_lda_predictions` data frame or tibble containing two columns: `predicted`, representing the predicted classes, and `actual`, representing the actual classes. Use `flights_lda_predictions` to obtain the confusion matrix for the model.

Part e (Code: 0.5 pts; Explanation: 2 pts)

Without running any additional code, use the confusion matrix from part (d) to estimate the sensitivity, specificity, positive predictive value, and negative predictive value for the model. Express all answers as fractions and then convert to decimals rounded to the thousandths place (3 decimal places).

Then, using the `summary` function on your confusion matrix, check your answers. Remember that we are trying to predict that a flight will be delayed (`arr_ontime` = "no").

Part f (Code: 3 pts; Explanation: 2 pts)

Repeat parts (c) through (e) for the QDA model. (Obviously, call your new data frame/tibble `flights_qda_predictions` instead.)

Problem 3: Model Selection

Part a (Code: 2 pts)

Add a column to the `flights_nb_predictions`, `flights_lda_predictions`, and `flights_qda_predictions` indicating the probability of not arriving on time (`arr_ontime == "no"`). It may be easiest to first obtain the predicted probabilities of being in each class, then add the column to the relevant data frame using `cbind` or `mutate`.

Then, compute the Brier scores for each model. As shown in the class activity, it is easiest to use the `mutate` function to obtain the "squared error" for each observation and then average the squared error.

Part b (Code: 1 pt)

Using the `mn_log_loss` function, obtain the cross-entropy/log loss for each of the three models.

Part c (Code: 1 pt)

Using the `mcc` function, obtain the Matthews Correlation Coefficient for each of the three models.

Part d (Explanation: 1.5 pts)

Compare the Brier score, log loss, and Matthews correlation coefficient for the three models by filling in the table below. Round all numbers to 3 decimal places.

Model	Brier	log loss	MCC
Naive Bayes			
LDA			
QDA			

Which of the three models performs the best on this test set by each measure?

Part e (Explanation: 1.5 pts)

If you had to recommend one of the three models to use to predict whether a flight would be delayed, would you use the Naive Bayes, LDA, or QDA model? Explain.