HW 4

Alyson Longworth

12/29/2023

This homework is designed to give you practice fitting a logistic regression and working with statistical/philosophical measures of fairness. We will work with the `titanic` dataset which we have previously seen in class in connection to decision trees.

Below I will preprocess the data precisely as we did in class. You can simply refer to `data_train` as your training data and `data_test` as your testing data.

```r
#this is all of the preprocessing done for the decision trees lecture.

path <- 'https://raw.githubusercontent.com/guru99-edu/R-Programming/master/titanic_data.csv'
titanic <-read.csv(path)
head(titanic)
```

```
##   x pclass survived                                            name    sex
## 1 1      1        1                   Allen, Miss. Elisabeth Walton female
## 2 2      1        1                  Allison, Master. Hudson Trevor   male
## 3 3      1        0                    Allison, Miss. Helen Loraine female
## 4 4      1        0            Allison, Mr. Hudson Joshua Creighton   male
## 5 5      1        0 Allison, Mrs. Hudson J C (Bessie Waldo Daniels) female
## 6 6      1        1                             Anderson, Mr. Harry   male
##      age sibsp parch ticket     fare   cabin embarked
## 1     29     0     0  24160 211.3375      B5        S
## 2 0.9167     1     2 113781   151.55 C22 C26        S
## 3      2     1     2 113781   151.55 C22 C26        S
## 4     30     1     2 113781   151.55 C22 C26        S
## 5     25     1     2 113781   151.55 C22 C26        S
## 6     48     0     0  19952    26.55     E12        S
##                         home.dest
## 1                    St Louis, MO
## 2 Montreal, PQ / Chesterville, ON
## 3 Montreal, PQ / Chesterville, ON
## 4 Montreal, PQ / Chesterville, ON
## 5 Montreal, PQ / Chesterville, ON
## 6                    New York, NY
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#replace ? with NA
replace_question_mark <- function(x) {
  if (is.character(x)) {
    x <- na_if(x, "?")
  }
  return(x)
}

titanic <- titanic %>%
  mutate_all(replace_question_mark)

set.seed(678)
shuffle_index <- sample(1:nrow(titanic))
head(shuffle_index)
```

```
## [1]   57  774  796 1044  681  920
```

```
titanic <- titanic[shuffle_index, ]
head(titanic)
```

```
##         x pclass survived                                         name
## 57     57      1        1                      Carter, Mr. William Ernest
## 774   774      3        0                               Dimic, Mr. Jovan
## 796   796      3        0                          Emir, Mr. Farred Chehab
## 1044 1044      3        1                     Murphy, Miss. Margaret Jane
## 681   681      3        0                              Boulos, Mr. Hanna
## 920   920      3        0 Katavelas, Mr. Vassilios ('Catavelas Vassilios')
##         sex  age sibsp parch ticket   fare   cabin embarked     home.dest
## 57     male   36     1     2 113760    120 B96 B98        S Bryn Mawr, PA
## 774    male   42     0     0 315088 8.6625    <NA>        S          <NA>
## 796    male <NA>     0     0   2631  7.225    <NA>        C          <NA>
## 1044 female <NA>     1     0 367230   15.5    <NA>        Q          <NA>
## 681    male <NA>     0     0   2664  7.225    <NA>        C         Syria
## 920    male 18.5     0     0   2682 7.2292    <NA>        C          <NA>
```

```
library(dplyr)
# Drop variables
clean_titanic <- titanic %>%
select(-c(home.dest, cabin, name, x, ticket)) %>%
#Convert to factor level
    mutate(pclass = factor(pclass, levels = c(1, 2, 3), labels = c('Upper', 'Middle', 'Lower')),
    survived = factor(survived, levels = c(0, 1), labels = c('No', 'Yes'))) %>%
na.omit()
#previously were characters
clean_titanic$age <- as.numeric(clean_titanic$age)
clean_titanic$fare <- as.numeric(clean_titanic$fare)
glimpse(clean_titanic)
```

```
## Rows: 1,043
## Columns: 8
## $ pclass   <fct> Upper, Lower, Lower, Middle, Lower, Middle, Lower, Lower, Upp~
## $ survived <fct> Yes, No, No, No, No, No, No, No, Yes, No, Yes, No, No, Yes, N~
## $ sex      <chr> "male", "male", "male", "male", "female", "female", "male", "~
## $ age      <dbl> 36.0, 42.0, 18.5, 44.0, 19.0, 26.0, 23.0, 28.5, 64.0, 36.5, 4~
## $ sibsp    <int> 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0~
## $ parch    <int> 2, 0, 0, 0, 0, 1, 0, 0, 2, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0~
## $ fare     <dbl> 120.0000, 8.6625, 7.2292, 13.0000, 16.1000, 26.0000, 7.8542, ~
## $ embarked <chr> "S", "S", "C", "S", "S", "S", "S", "S", "C", "S", "S", "S", "~
```

```r
create_train_test <- function(data, size = 0.8, train = TRUE) {
    n_row = nrow(data)
    total_row = size * n_row
    train_sample <- 1: total_row
    if (train == TRUE) {
        return (data[train_sample, ])
    } else {
        return (data[-train_sample, ])
    }
}
data_train <- create_train_test(clean_titanic, 0.8, train = TRUE)
data_test <- create_train_test(clean_titanic, 0.8, train = FALSE)
```

Create a table reporting the proportion of people in the training set surviving the Titanic. Do the same for the testing set. Comment on whether the current training-testing partition looks suitable.

```r
summary(data_train)
```

```
##     pclass    survived      sex                  age             sibsp
##  Upper :220   No :502   Length:834         Min.   : 0.1667   Min.   :0.0000
##  Middle:209   Yes:332   Class :character   1st Qu.:21.0000   1st Qu.:0.0000
##  Lower :405             Mode  :character   Median :28.0000   Median :0.0000
##                                            Mean   :29.6103   Mean   :0.5144
##                                            3rd Qu.:38.8750   3rd Qu.:1.0000
##                                            Max.   :76.0000   Max.   :8.0000
##      parch             fare          embarked
##  Min.   :0.0000   Min.   :  0.00   Length:834
##  1st Qu.:0.0000   1st Qu.:  8.05   Class :character
##  Median :0.0000   Median : 15.25   Mode  :character
##  Mean   :0.4149   Mean   : 34.65
##  3rd Qu.:1.0000   3rd Qu.: 32.16
##  Max.   :6.0000   Max.   :512.33
```

```r
summary(data_test)
```

```
##     pclass   survived      sex                  age             sibsp
##  Upper :62   No :116   Length:209         Min.   : 0.6667   Min.   :0.0000
##  Middle:52   Yes: 93   Class :character   1st Qu.:21.0000   1st Qu.:0.0000
```

```
##  Lower :95            Mode  :character   Median :27.0000   Median :0.0000
##                                          Mean   :30.6228   Mean   :0.4641
##                                          3rd Qu.:39.0000   3rd Qu.:1.0000
##                                          Max.   :80.0000   Max.   :4.0000
##      parch             fare          embarked
##  Min.   :0.0000   Min.   :  0.00   Length:209
##  1st Qu.:0.0000   1st Qu.:  8.05   Class :character
##  Median :0.0000   Median : 19.50   Mode  :character
##  Mean   :0.4498   Mean   : 44.41
##  3rd Qu.:1.0000   3rd Qu.: 50.50
##  Max.   :4.0000   Max.   :512.33
```

```
prop_surv_train <- prop.table(table(data_train$survived))
prop_surv_test <- prop.table(table(data_test$survived))
survival_table <- data.frame(Dataset = c("Training", "Testing"), Survived = c(prop_surv_train["Yes"], pr
print(survival_table)
```

```
##    Dataset  Survived
## 1 Training 0.3980815
## 2  Testing 0.4449761
```

In the training dataset, 39.8% of individuals survived and in the testing dataset, 44.5% of individuals survived. Comparing these proportions, I notice that the proportion of survivors in the testing dataset was about 4.7% higher than in the training dataset. However, because these proportions are relatively close, I would say that the current training-testing partition looks suitable.

Use the `glm` command to build a logistic regression on the training partition. `survived` should be your response variable and `pclass`, `sex`, `age`, `sibsp`, and `parch` should be your response variables.

```
model <- glm(survived ~ pclass + sex + age + sibsp + parch,family = binomial(link = 'logit'),data = data
summary(model)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp + parch,
##     family = binomial(link = "logit"), data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6566  -0.6647  -0.3999   0.6465   2.4394
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.903165   0.409280   9.537  < 2e-16 ***
## pclassMiddle -1.291506   0.257421  -5.017 5.25e-07 ***
## pclassLower  -2.404084   0.262022  -9.175  < 2e-16 ***
## sexmale      -2.684206   0.200130 -13.412  < 2e-16 ***
## age          -0.036776   0.007494  -4.907 9.24e-07 ***
## sibsp        -0.395584   0.118587  -3.336  0.00085 ***
```

```
## parch          0.032494   0.111916   0.290  0.77155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1121.27  on 833  degrees of freedom
## Residual deviance:  757.87  on 827  degrees of freedom
## AIC: 771.87
##
## Number of Fisher Scoring iterations: 5
```

We would now like to test whether this classifier is *fair* across the sex subgroups. It was reported that women and children were prioritized on the life-boats and as a result survived the incident at a much higher rate. Let us see if our model is able to capture this fact.

Subset your test data into a male group and a female group. Then, use the `predict` function on the male testing group to come up with predicted probabilities of surviving the Titanic for each male in the testing set. Do the same for the female testing group.

```r
male_test <- data_test[data_test$sex == "male", ]
female_test <- data_test[data_test$sex == "female", ]
male_test_predicted_prob <- predict(model, newdata = male_test, type = 'response')
summary(male_test_predicted_prob)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02236 0.09884 0.12779 0.19792 0.27053 0.75710
```

```r
female_test_predicted_prob <- predict(model, newdata = female_test, type = 'response')
summary(female_test_predicted_prob)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3626  0.6660  0.8132  0.7710  0.8973  0.9581
```

Now recall that for this logistic *regression* to be a true classifier, we need to pair it with a decision boundary. Use an `if-else` statement to translate any predicted probability in the male group greater than 0.5 into `Yes` (as in Yes this individual is predicted to have survived). Likewise an predicted probability less than 0.5 should be translated into a `No`.

Do this for the female testing group as well, and then create a confusion matrix for each of the male and female test set predictions. You can use the `confusionMatrix` command as seen in class to expidite this process as well as provide you necessary metrics for the following questions.

```r
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
male_predicted <- ifelse(male_test_predicted_prob > 0.5, "Yes", "No")
female_predicted <- ifelse(female_test_predicted_prob > 0.5, "Yes", "No")
cm_male <- confusionMatrix(as.factor(male_predicted), male_test$survived, positive = "Yes")
cm_male
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  93  28
##        Yes  4   4
##
##                Accuracy : 0.7519
##                  95% CI : (0.6682, 0.8237)
##     No Information Rate : 0.7519
##     P-Value [Acc > NIR] : 0.5473
##
##                   Kappa : 0.1119
##
##  Mcnemar's Test P-Value : 4.785e-05
##
##             Sensitivity : 0.12500
##             Specificity : 0.95876
##          Pos Pred Value : 0.50000
##          Neg Pred Value : 0.76860
##              Prevalence : 0.24806
##          Detection Rate : 0.03101
##    Detection Prevalence : 0.06202
##       Balanced Accuracy : 0.54188
##
##        'Positive' Class : Yes
##
```

```
cm_female <- confusionMatrix(as.factor(female_predicted), female_test$survived, positive = "Yes")
cm_female
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No   4   2
##        Yes 15  59
##
##                Accuracy : 0.7875
##                  95% CI : (0.6817, 0.8711)
##     No Information Rate : 0.7625
##     P-Value [Acc > NIR] : 0.354209
##
##                   Kappa : 0.2325
##
##  Mcnemar's Test P-Value : 0.003609
```

```
##
##             Sensitivity : 0.9672
##             Specificity : 0.2105
##          Pos Pred Value : 0.7973
##          Neg Pred Value : 0.6667
##              Prevalence : 0.7625
##          Detection Rate : 0.7375
##    Detection Prevalence : 0.9250
##       Balanced Accuracy : 0.5889
##
##        'Positive' Class : Yes
##
```

We can see that indeed, at least within the testing groups, women did seem to survive at a higher proportion than men (24.8% to 76.3% in the testing set). Print a summary of your trained model and interpret one of the fitted coefficients in light of the above disparity.

```r
summary(model)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp + parch,
##     family = binomial(link = "logit"), data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6566  -0.6647  -0.3999   0.6465   2.4394
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.903165   0.409280   9.537  < 2e-16 ***
## pclassMiddle -1.291506   0.257421  -5.017 5.25e-07 ***
## pclassLower  -2.404084   0.262022  -9.175  < 2e-16 ***
## sexmale      -2.684206   0.200130 -13.412  < 2e-16 ***
## age          -0.036776   0.007494  -4.907 9.24e-07 ***
## sibsp        -0.395584   0.118587  -3.336  0.00085 ***
## parch         0.032494   0.111916   0.290  0.77155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1121.27  on 833  degrees of freedom
## Residual deviance:  757.87  on 827  degrees of freedom
## AIC: 771.87
##
## Number of Fisher Scoring iterations: 5
```

In light of the disparity above, I chose to interpret the "sexmale" coefficient. For this coefficient, the estimate is -2.684206 and because it is the largest negative value of all coefficients, it can be interpreted that males are associated with the lowest odds of survival compared to females.

Now let's see if our model is *fair* across this explanatory variable. Calculate five measures (as defined in class) in this question: the Overall accuracy rate ratio between females and males, the disparate impact between females and males, the statistical parity between females and males, and the predictive equality as well as equal opportunity between females and males (collectively these last two comprise equalized odds). Set a reasonable ϵ each time and then comment on which (if any) of these five criteria are met.

```
#overall accuracy rate ratio between females and males
accuracy_male <- (93+4)/(93+4+28+4)
accuracy_female <-(4+59)/(59+4+15+2)
overall_accuracy_ratio <- accuracy_female / accuracy_male

#disparate impact between females and males
disparate_impact <- (59+2)/(4+28)

#statistical parity between females and males
statistical_parity <- (59/(59+2)) - (4/(4+28))

#predictive equality between females and males
predictive_equality <- (15/(15+4)) - (4/(4+93))

#equal opportunity between females and males
equal_opportunity <- (59/(59+2)) - (4/(4+28))

overall_accuracy_ratio_fair <- abs(overall_accuracy_ratio - 1) > 0.2
overall_accuracy_ratio_fair
```

```
## [1] FALSE
```

```
disparate_impact_fair <- disparate_impact < 1-0.1
disparate_impact_fair
```

```
## [1] FALSE
```

```
statistical_parity_fair <- abs(statistical_parity) > 0.2
statistical_parity_fair
```

```
## [1] TRUE
```

```
predictive_equality_fair <- abs(predictive_equality) > 0.2
predictive_equality_fair
```

```
## [1] TRUE
```

```
equal_opportunity_fair <- abs(equal_opportunity) > 0.2
equal_opportunity_fair
```

```
## [1] TRUE
```

Only the statistical parity, predictive equality, and equal opportunity fairness criteria are met because the differences between males and females in each of these measures is within the threshold of 0.2.

It is always important for us to interpret our results in light of the original data and the context of the analysis. In this case, it is relevant that we are analyzing a historical event post-facto and any disparities across demographics identified are unlikely to be replicated. So even though our model fails numerous of the statistical fairness criteria, I would argue we need not worry that our model could be misused to perpetuate discrimination in the future. After all, this model is likely not being used to prescribe a preferred method of treatment in the future.

Even so, provide a *philosophical* notion of justice or fairness that may have motivated the Titanic survivors to act as they did. Spell out what this philosophical notion or principle entails?

The Titanic survivors may have been motivated to act as they did to prioritize the women and children on the life-boats because they could have been viewing the situation with a virtue ethics perspective or a utilitarian perspective. At the time, the Titanic survivors chose to prioritize the safety of the most vulnerable groups who may not have been able to protect themselves as well as the others on the ship could have. With this in mind, these survivors could have chosen to act with bravery, generosity, and empathy to protect the more vulnerable groups. On the other hand, this idea could have been viewed with a utilitarian perspective because if these people were able to prioritize the most vulnerable groups with the lowest survival potential, considering there would be more of a chance that the stronger group of people on the ship would have survived with less assistance, this would result in overall more survivors.