HW 4

Logan Schmitt

03/18/2024

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.

```
#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
```

```
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", "~
```

```
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.

```
#Overall proportion of people who survived / did not survive the Titanic
round(prop.table(table(clean_titanic$survived)),2)
```

```
##
##   No  Yes
## 0.59 0.41
```

```
#Training set table
round(prop.table(table(data_train$survived)),2)
```

```
##
##  No Yes
## 0.6 0.4
```

```
#Testing set table
round(prop.table(table(data_test$survived)),2)
```

```
##
##   No  Yes
## 0.56 0.44
```

Logan Input: In the training set, roughly 40% of the passengers survived. In the testing set, roughly 44% of the passengers survived. Since the testing set proportion is similar to the training set proportion (which is representative of the overall dataset), 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.

```
head(data_train)
```

```
##      pclass survived    sex  age sibsp parch     fare embarked
```

```
## 57    Upper      Yes   male 36.0     1     2 120.0000        S
## 774   Lower       No   male 42.0     0     0   8.6625        S
## 920   Lower       No   male 18.5     0     0   7.2292        C
## 430  Middle       No   male 44.0     0     0  13.0000        S
## 1012  Lower       No female 19.0     1     0  16.1000        S
## 476  Middle       No female 26.0     1     1  26.0000        S
```

```
model = glm(survived ~ pclass+sex+age+sibsp+parch, family = binomial(link="logit"), data = data_train)
model
```

```
##
## Call:  glm(formula = survived ~ pclass + sex + age + sibsp + parch,
##     family = binomial(link = "logit"), data = data_train)
##
## Coefficients:
##  (Intercept)  pclassMiddle   pclassLower       sexmale           age
##      3.90316      -1.29151      -2.40408      -2.68421      -0.03678
##        sibsp         parch
##     -0.39558       0.03249
##
## Degrees of Freedom: 833 Total (i.e. Null);  827 Residual
## Null Deviance:       1121
## Residual Deviance: 757.9     AIC: 771.9
```

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.

```
#Subsetting testing data into a male / female group:
male_data_test = data_test %>%
  filter(sex=="male")
female_data_test = data_test %>%
  filter(sex=="female")

#Predicting male testing group survival probabilities
male_results = predict(model, newdata = male_data_test, type = 'response')

#Predicting female testing group survival probabilities
female_results = predict(model, newdata = female_data_test, type = 'response')
```

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.

```
library(caret)
```

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

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

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

```
#Creating decision boundary for male survival results
male_results = ifelse(male_results > 0.5,"Yes","No")
male_confusionMatrix = confusionMatrix(
  as.factor(male_results), male_data_test$survive, positive = "Yes")
male_confusionMatrix
```

```
## 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
##
```

```
#Creating decision boundary for female survival results
female_results = ifelse(female_results > 0.5,"Yes","No")
female_confusionMatrix = confusionMatrix(
  as.factor(female_results), female_data_test$survive, positive = "Yes")
female_confusionMatrix
```

```
## 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.

```
# Male survival proportions in the testing set
round(prop.table(table(male_data_test$survived)),3)
```

```
##
##    No   Yes
## 0.752 0.248
```

```
# Female survival proportions in the testing set
round(prop.table(table(female_data_test$survived)),3)
```

```
##
##    No   Yes
## 0.238 0.762
```

```
# Summary of the model
summary(model)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp + parch,
##     family = binomial(link = "logit"), data = data_train)
##
## 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
```

Logan Input: Interpreting pclassLower: for passengers with lower-class tickets, their log odds of surviving decreases by -2.404084 compared to passengers with upper-class tickets, holding all other variables constant.

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.

```
epsilon = 0.2
# Measure 1: the Overall accuracy rate ratio between females and males
OverallAccuracyRatioMale = (93+4) / (93+4+4+28)
OverallAccuracyRatioMale
```

```
## [1] 0.751938
```

```
OverallAccuracyRatioFemale = (4+59) / (4+59+15+2)
OverallAccuracyRatioFemale
```

```
## [1] 0.7875
```

```
# Measure 2: the disparate impact between females and males
DInumerator = (28+4) / (28+4+93+4) # Males
DIdenominator = (2+59) / (2+59+4+15) # Females
DI = DInumerator / DIdenominator
DI
```

```
## [1] 0.3253272
```

```
# Measure 3: the statistical parity between females and males
StatParity = abs(DInumerator-DIdenominator)
StatParity
```

```
## [1] 0.514438
```

```
# Measure 4: the predictive equality between females and males
PEfirst = 28 / (28+93)
PEsecond = 2 / (2+4)
PredEquality = abs(PEfirst - PEsecond) # False Positive Rate
PredEquality
```

```
## [1] 0.1019284
```

```r
# Measure 5: equal opportunity between females and males
EOfirst = 4 / (4+4)
EOsecond = 59 / (59+15)
EqualOpp = abs(EOfirst - EOsecond) # True Positive Rate
EqualOpp
```

```
## [1] 0.2972973
```

```r
# Testing disparate impact
DI < (1-epsilon)
```

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

```r
# Testing statistical parity
StatParity > epsilon
```

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

```r
# Testing predictive equality
PredEquality > epsilon
```

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

```r
# Testing equal opportunity
EqualOpp > epsilon
```

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

Logan Input: The overall accuracy ratio is above 70% for both classes, so this criteria indicates a relatively high overall accuracy. The disparate impact between females and males exists when the male survival rate divided by the female survival rate is less than 1 - epsilon. Since 0.75 is less than 0.8, disparate impact exists among the classes. The statistical parity between females and males exists when the absolute difference between the male and female survival rates is greater than epsilon. Since 0.79 is greater than 0.2, statistical parity exists among the classes. The predictive equality between females and males exists when the false positive rate is greater than epsilon. Since 0.1 is not greater than 0.2, predictive equality is not violated. The equal opportunity between females and males when the true positive rate is greater than epsilon. Since 0.3 is greater than 0.2, equal opportunity is violated Here, we see overall accuracy and some criteria met, but a few statistical fairness conditions remain violated. NOTE: In the Titanic dataset, the protected class is the female sex.

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?

Logan Input: John Rawls emphasized the idea of "justice as fairness." His concept, essentially an extension of "justice as need," has two components. Firstly it states that some amount of difference is inherent to the world and wherever these differences exist, we as a society should allocate resources to protect the most vulnerable. Secondly, we should allocate resources under a "veil of ignorance," such that we are impartial to our own characteristics in society and don't favor one group over another in this distribution.
Relating this philosophical notion of fairness to the Titanic, those on board determined women and children

most vulnerable (one reason possibly being due to the fact that men and / or adults can survive longer without extra help). As a result, crew and passengers prioritized the lifeboats and life vests for these two groups, as it was the most “fair” way to allocate these limited resources.