COMPSCIX 415.2 Homework 8

Vishnu Vardhan

4/3/2018

Contents

Q1 1
Q2 2
Q3 2
Q4 3
Q5-A 4
Q5-B 4
Q5-C 5
Q5-C 5

Q1

Load the train.csv dataset into R. How many observations and columns are there? Convert the target variable to a factor because it will be loaded into R as an integer by default. ### Answer

```
d <- read_csv("train.csv")
```

```
## Parsed with column specification:
## cols(
##   PassengerId = col_integer(),
##   Survived = col_integer(),
##   Pclass = col_integer(),
##   Name = col_character(),
##   Sex = col_character(),
##   Age = col_double(),
##   SibSp = col_integer(),
##   Parch = col_integer(),
##   Ticket = col_character(),
##   Fare = col_double(),
##   Cabin = col_character(),
##   Embarked = col_character()
## )
```

```
glimpse(d)
```

```
## Observations: 891
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3,...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bra...
## $ Sex         <chr> "male", "female", "female", "female", "male", "mal...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, ...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4,...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1,...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1138...
```

```
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, NA, ...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", ...
```

```
d$Survived <- factor(d$Survived)
```

Q2

Our first step is to randomly split the data into train and test datasets. We will use a 70/30 split, and use the random seed of 29283 so that we all should get the same training and test set.

Answer

```
library (caTools)

set.seed(29283)
sample = sample.split(d, SplitRatio = 0.7)
train  = subset(d, sample == TRUE)
test   = subset(d, sample == FALSE)
```

Q3

Our target is called Survived. First, fit a logistic regression model using Pclass, Sex, Fare as your three features. Fit the model using the glm() function. Ask yourself these questions before fitting the model: • What kind of relationship will these features have with the probability of survival? • Are these good features, given the problem we are trying to solve?

After fitting the model, output the coefficients using the broom package and answer these questions: • How would you interpret the coefficients? • Are the features significant? Use the code below and fill in the blanks.

Answer

Having explored the data before, i would expect pClass to be strongly correlated to surival. The same applies for gender.

Yes, these are good features, since this would either reflect a bias, or potential seating situations that could affect survival.

The estimates indicate that being “male”, or being in a low “class” is strongly correlated to not surving.

Apart from fare all the features are significant.

```
library(broom)
# Fit a model with intercept only
mod_1 <- glm(Survived ~ Pclass + Sex + Fare, data = train, family = 'binomial')
# take a look at the features and coefficients
tidy(mod_1)
```

```
##          term      estimate   std.error   statistic      p.value
## 1 (Intercept)  3.0419518912 0.445497005   6.8282207 8.597423e-12
## 2      Pclass -0.9119952774 0.148868564  -6.1261777 9.001528e-10
## 3     Sexmale -2.5511219484 0.223834027 -11.3973822 4.308812e-30
## 4        Fare  0.0008583218 0.002651979   0.3236534 7.462005e-01
```

Q4

Now, let's fit a model using a classification tree, using the same features and plot the final decision tree. Use the code below for help. Answer these questions: • Describe in words one path a Titanic passenger might take down the tree. (Hint: look at your tree, choose a path from the top to a terminal node, and describe the path like this - a male passenger who paid a fare > 30 and was in first class has a high probability of survival) • Does anything surprise you about the fitted tree?

Answer

A female passenger in a class greater than 2.5 (essentially - class 3), and with fare >= 22.904 had a less than 20% chance of surviving.

I was really surprised that fare played such a major part of the decision tree.

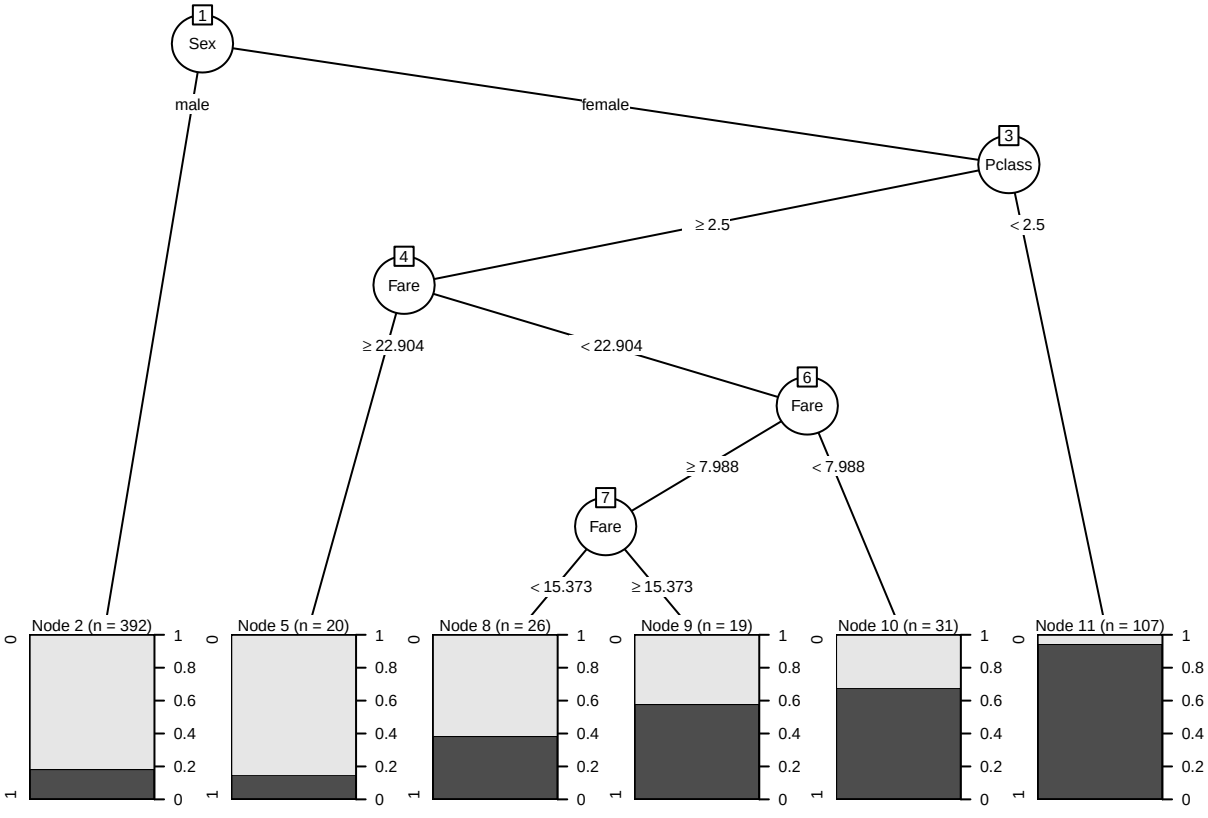
```
library(rpart)
library(partykit)
```

```
## Loading required package: grid
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
tree_mod <- rpart(Survived ~ Pclass + Sex + Fare, data = train)
plot(as.party(tree_mod), gp = gpar(fontsize = 6))
```



Q5-A

Evaluate both the logistic regression model and classification tree on the test_set. First, use the predict() function to get the model predictions for the testing set. Use the code below for help.

Answer

```
test_logit <- predict(mod_1, newdata = test, type = 'response')
test_tree <- predict(tree_mod, newdata = test)[,2]
```

Q5-B

Next, we will plot the ROC curves from both models using the code below. Don't just copy and paste the code. Go through it line by line and see what it is doing. Recall that predictions from your decision tree are given as a two column matrix.

Answer

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

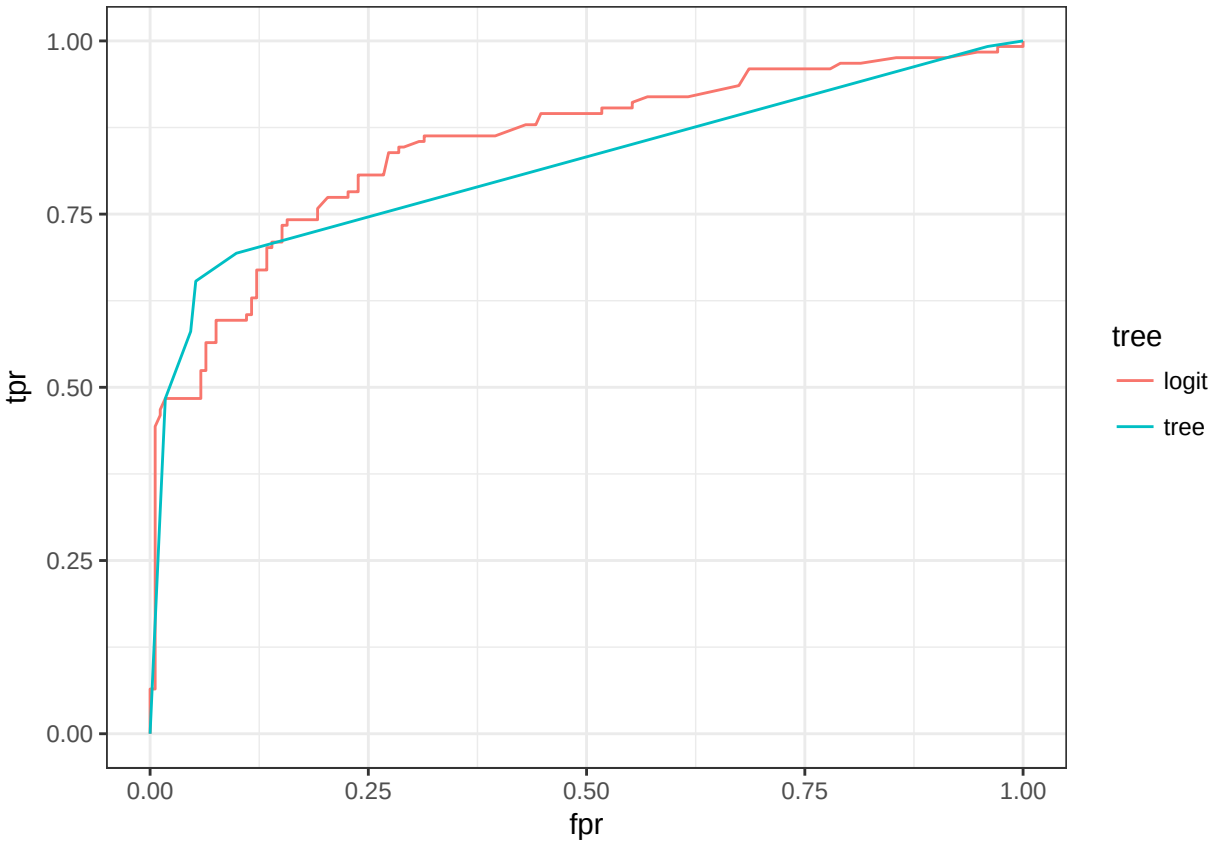
```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
perf_logit = performance (
  prediction(predictions = test_logit, labels = test$Survived),
  measure = "tpr",
  x.measure = "fpr")

perf_tree = performance (
  prediction(predictions = test_tree, labels = test$Survived),
  measure = "tpr",
  x.measure = "fpr")

roc_data <- bind_rows(
  tibble(fpr = perf_logit@x.values[[1]],tpr = perf_logit@y.values[[1]], tree = "logit"),
  tibble(fpr = perf_tree@x.values[[1]],tpr = perf_tree@y.values[[1]], tree = "tree")
  )

ggplot(data = roc_data, aes(fpr,tpr)) + geom_line(aes(colour = tree)) + theme_bw()
```



Q5-C

Now, use the performance() function to calculate the area under the curve (AUC) for both ROC curves. Check ?performance for help on plugging in the right measure argument.

Answer

```
# calculate the AUC
auc_logit <-  performance (
  prediction(predictions = test_logit, labels = test$Survived), measure = "auc")
auc_tree <-  performance (
  prediction(predictions = test_tree, labels = test$Survived), measure = "auc")
# extract the AUC value
auc_logit@y.values[[1]]
```

```
## [1] 0.8508299
```

```
auc_tree@y.values[[1]]
```

```
## [1] 0.8202832
```

Q5-C

Lastly, pick a probability cutoff by looking at the ROC curves. You pick, there's no right answer (but there is a wrong answer - make sure to pick something between 0 and 1). Using that probability cutoff, create the

confusion matrix for each model by following these steps:

Answer

```
cut_off = 0.25

test_set <- test

test_set$predicted_logit <- test_logit
test_set$predicted_tree  <- test_tree

test_set %>% mutate(s = case_when(
  predicted_logit > cut_off ~ "Yes",
  predicted_logit <= cut_off ~ "No")) %>%  count(s, Survived)  %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##       s   `0`   `1`
## * <chr> <int> <int>
## 1    No   124    20
## 2   Yes    48   104
```

```
test_set %>% mutate(s = case_when(
  predicted_tree > cut_off ~ "Yes",
  predicted_tree <= cut_off ~ "No")) %>%  count(s, Survived)  %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##       s   `0`   `1`
## * <chr> <int> <int>
## 1    No   155    38
## 2   Yes    17    86
```