COMPSCIX 415.2 Homework 8

Bryan Hee

March 26, 2018

Exercise 1

There are 891 observations or people and 12 variables/columns in the Titanic training dataset.

```
Titanic_kaggle <- read.csv("C:/Users/BryanHee/OneDrive - stok LLC/Intro to Data Science/HW Assignments/
Titanic_kaggle <- transform(Titanic_kaggle, Survived = as.factor(Survived))
glimpse(Titanic_kaggle)
```

```
## Observations: 891
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <fct> 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        <fct> Braund, Mr. Owen Harris, Cumings, Mrs. John Bradle...
## $ Sex         <fct> male, female, female, female, male, male, male, ma...
## $ 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      <fct> A/5 21171, PC 17599, STON/O2. 3101282, 113803, 373...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <fct> , C85, , C123, , , E46, , , , G6, C103, , , , , , ...
## $ Embarked    <fct> S, C, S, S, S, Q, S, S, S, C, S, S, S, S, S, S, Q,...
```

```
Titanic_kaggle %>%
  group_by(Survived) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   Survived count
##   <fct>    <int>
## 1 0          549
## 2 1          342
```

Exercise 2

```
set.seed(29283)

Titanic_train <- Titanic_kaggle %>% sample_frac(0.7)
Titanic_test <- Titanic_kaggle %>% filter(!(Titanic_kaggle$PassengerId %in% Titanic_train$PassengerId))
```

Exercise 3

Using a logistic regression model, Survived is predicted based off of the attributes Pclass, Sex and Fare.

Based off of the broom function, tidy, we are able to see the estimates, std.error, statistics, and p.values of the model inputs. The estimate related to class tells us that as the Pclass value goes up by one (i.e. going from middle class to lower class), the odds of survival drops by ~0.88. The odds are related to the probability.

The coefficient associated with sex is intepretted as: males' odds of survival is 2.8 less than women. The fare estimate predicts that as the fare goes up by $1, the odds of survival go up by ~0.002.

Based off of a 0.5 p-value cutoff, all features are significant. Fare is extremely close to the cutoff though.

```
mod_1 <- glm(Survived ~ Pclass + Sex + Fare, data = Titanic_train, family = 'binomial')
tidy(mod_1)
```

```
##          term     estimate   std.error   statistic      p.value
## 1 (Intercept)  3.158129368 0.430541964   7.3352417 2.213217e-13
## 2      Pclass -0.875841345 0.145107088  -6.0358274 1.581502e-09
## 3     Sexmale -2.840421241 0.228239516 -12.4449144 1.490440e-35
## 4        Fare  0.001846965 0.002290049   0.8065179 4.199443e-01
```

Exercise 4

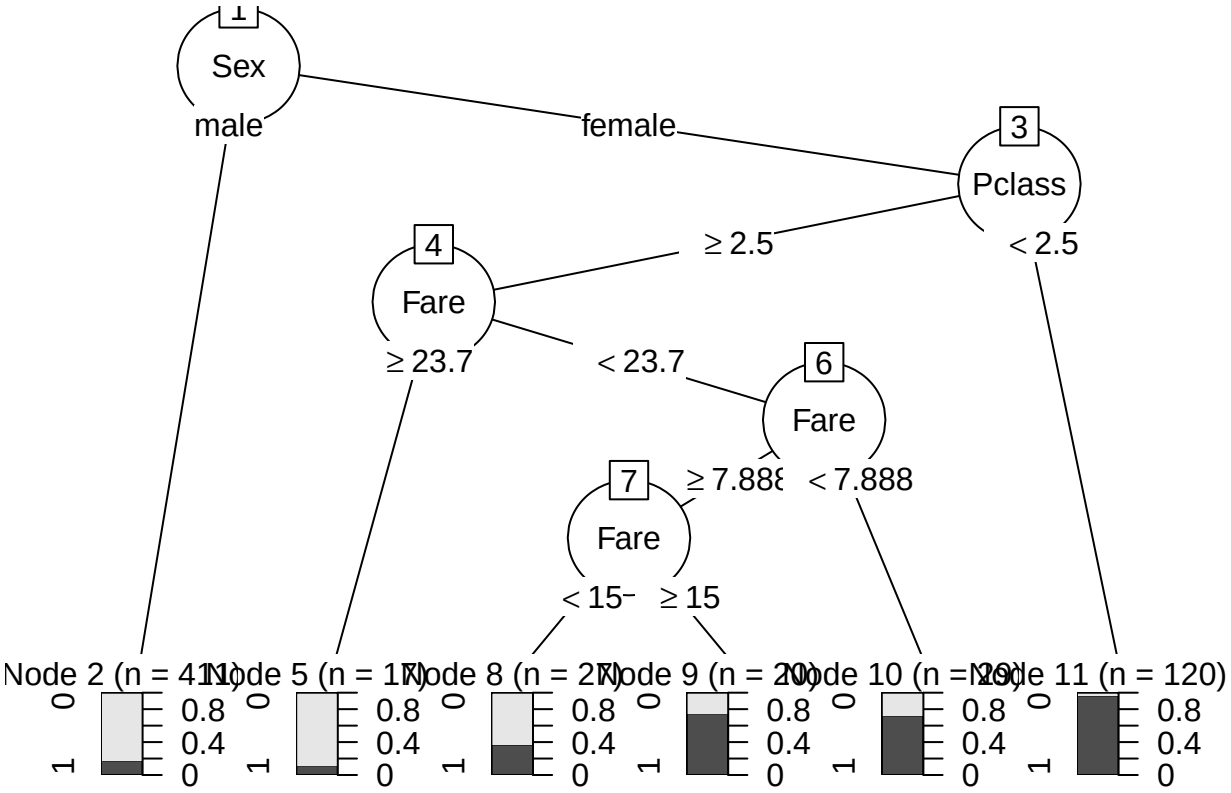
One path a Titanic passenger might take down the tree: A female passenger with a class proxy number greater than or equal to 2.5, and a fare value less than $23.7 but greater than or equal to 15 dollars has a 75% probability of survival based off of the training dataset.

Something that surprises me about the results of the tree, was that the node with the second lowest probability of survival was lower class women who had paid a lot for their fare. Also, most lower class women who paid less than 7.90 dollars survived. However, most lower class women who paid a little more (>$7.90 but < 15 dollars) died.

```
tree_mod <- rpart(Survived ~ Pclass + Sex + Fare, data = Titanic_train)
tree_mod
```

```
## n= 624
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 624 237 0 (0.62019231 0.37980769)
##    2) Sex=male 411  73 0 (0.82238443 0.17761557) *
##    3) Sex=female 213  49 1 (0.23004695 0.76995305)
##      6) Pclass>=2.5 93  45 1 (0.48387097 0.51612903)
##       12) Fare>=23.7 17   2 0 (0.88235294 0.11764706) *
##       13) Fare< 23.7 76  30 1 (0.39473684 0.60526316)
##         26) Fare>=7.8875 47  22 1 (0.46808511 0.53191489)
##           52) Fare< 15 27  10 0 (0.62962963 0.37037037) *
##           53) Fare>=15 20   5 1 (0.25000000 0.75000000) *
##         27) Fare< 7.8875 29   8 1 (0.27586207 0.72413793) *
##      7) Pclass< 2.5 120   4 1 (0.03333333 0.96666667) *
```

```
plot(as.party(tree_mod))
```



Exercise 5

For the following code, i chose to use the train dataset to then inform the predictions on the test dataset. This code differed slightly from what was provided on the homework handout.

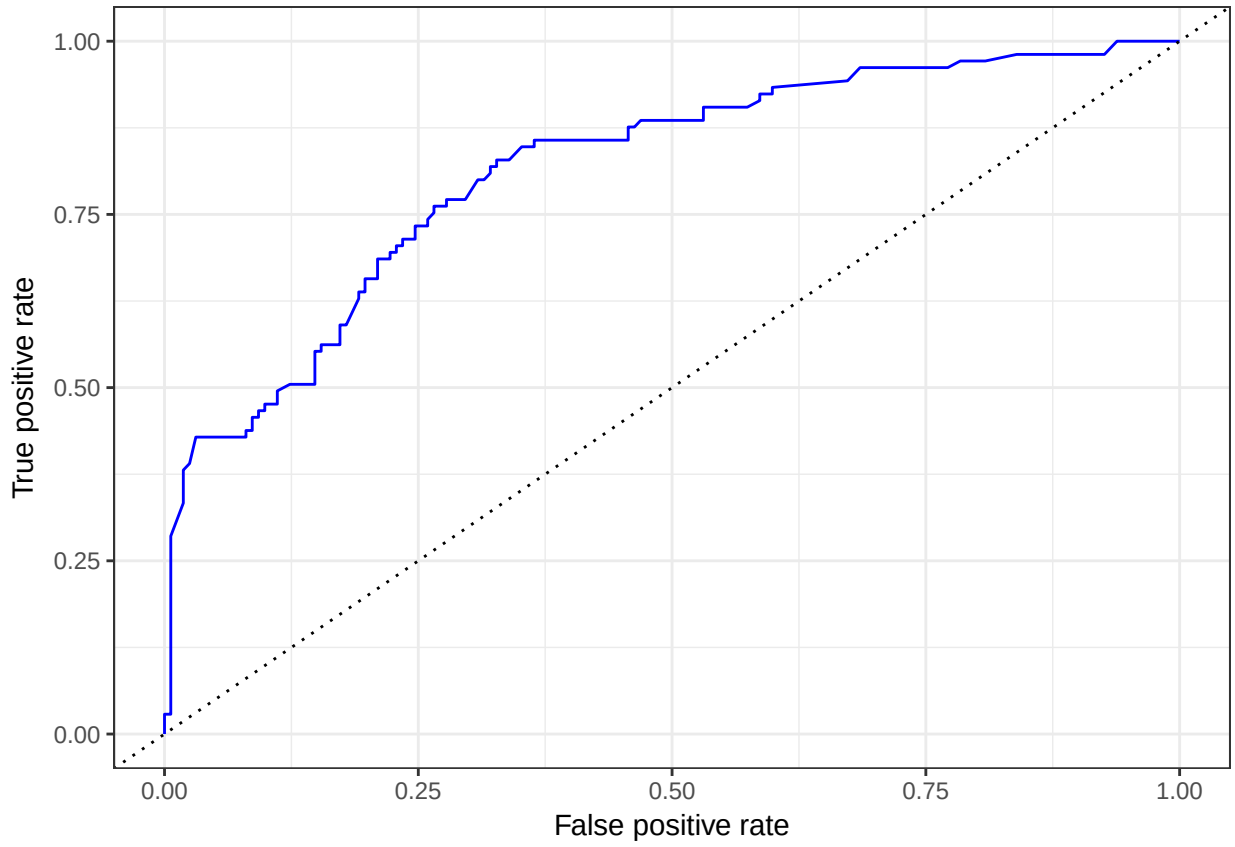
```
mod_2 <- glm(Survived ~ Pclass + Sex + Fare, data = Titanic_train, family = 'binomial')
tidy(mod_2)
```

```
##          term     estimate   std.error   statistic      p.value
## 1 (Intercept)  3.158129368 0.430541964   7.3352417 2.213217e-13
## 2      Pclass -0.875841345 0.145107088  -6.0358274 1.581502e-09
## 3     Sexmale -2.840421241 0.228239516 -12.4449144 1.490440e-35
## 4        Fare  0.001846965 0.002290049   0.8065179 4.199443e-01
```

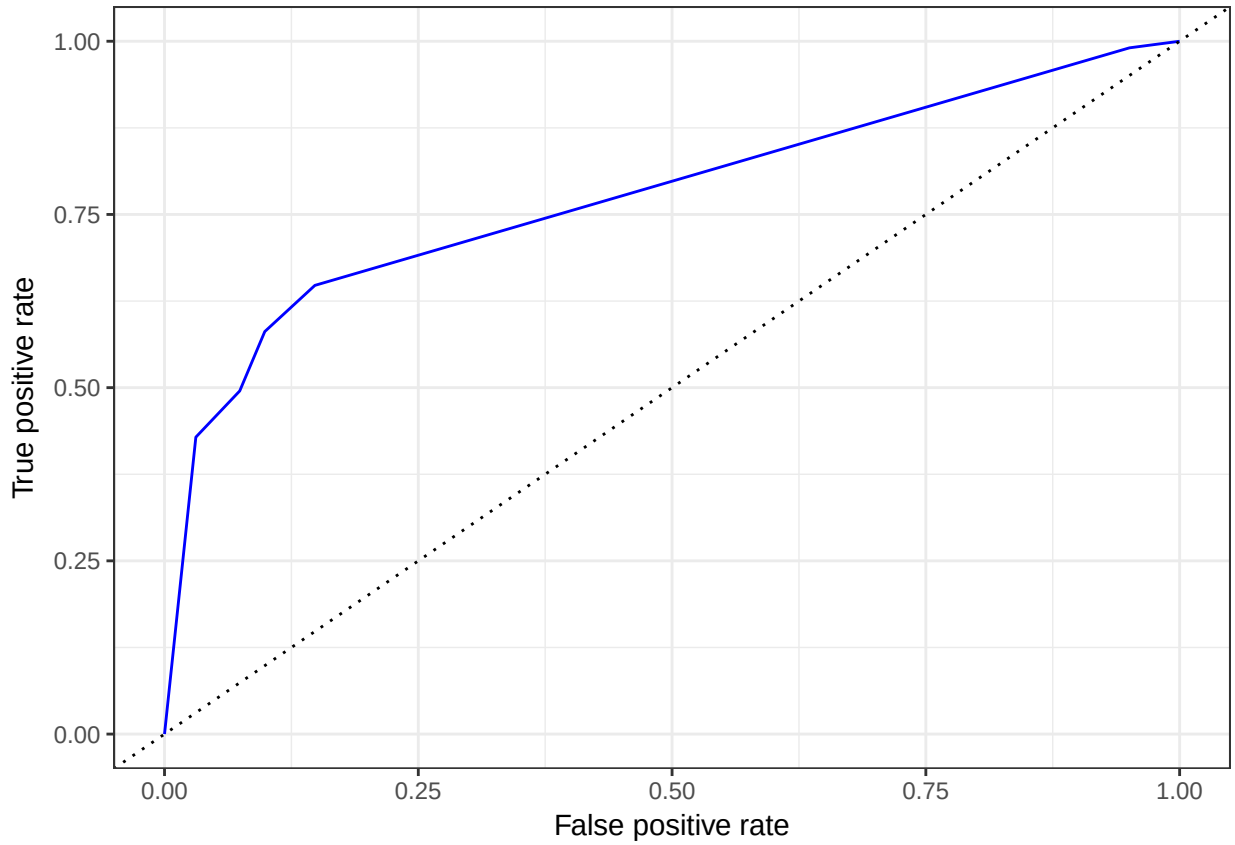
```
test_logit <- predict(mod_2, newdata = Titanic_test, type = 'response')

tree_mod2 <- rpart(Survived ~ Pclass + Sex + Fare, data = Titanic_train)
test_tree <- predict(tree_mod2, newdata = Titanic_test)[,2]
```

```
pred_logit <- prediction(predictions = test_logit, labels = Titanic_test$Survived)
pred_tree <- prediction(predictions = test_tree, labels = Titanic_test$Survived)

perf_logit <- performance(pred_logit, measure = 'tpr', x.measure = 'fpr')
perf_logit_tbl <- tibble(perf_logit@x.values[[1]], perf_logit@y.values[[1]])
names(perf_logit_tbl) <- c('fpr', 'tpr')
```

```r
perf_tree <- performance(pred_tree, measure = 'tpr', x.measure = 'fpr')
perf_tree_tbl <- tibble(perf_tree@x.values[[1]], perf_tree@y.values[[1]])
names(perf_tree_tbl) <- c('fpr', 'tpr')

plot_roc <- function(perf_tbl) {
  p <- ggplot(data = perf_tbl, aes(x = fpr, y = tpr)) +
  geom_line(color = 'blue') +
  geom_abline(intercept = 0, slope = 1, lty = 3) +
  labs(x = 'False positive rate', y = 'True positive rate') +
  theme_bw()
  return(p)
}

plot_roc(perf_logit_tbl)
```




```r
plot_roc(perf_tree_tbl)
```



Based off of the area under the curve, AUC, values both models seem to be performing decently well. Anything greater than 0.5 is better than a chance guess. The Logistic model is actually performing better than the classification tree, which was surprising to me. These models could probably be improved by swapping out a feature for fare.

```
auc_logit <- performance(prediction.obj = pred_logit, measure = "auc")
auc_tree <- performance(prediction.obj = pred_tree, measure = "auc")

# extract the AUC value
auc_logit@y.values[[1]]
```

```
## [1] 0.8147854
```

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

```
## [1] 0.776602
```

```
Titanic_test %>%
  mutate(pred_logit = test_logit,
         pred_tree = test_tree) -> Titanic_test

cutoff = 0.3
Titanic_test %>%
  mutate(logit = case_when(pred_logit >= cutoff ~ "yes",
                    pred_logit < cutoff ~ "no"),
         tree =  case_when(pred_tree >= cutoff ~ "yes",
                           pred_tree < cutoff ~ "no")) -> Titanic_test
```

```
Titanic_test %>% count(logit, Survived) %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##   logit   `0`   `1`
## * <chr> <int> <int>
## 1 no      112    21
## 2 yes      50    84
```

```
Titanic_test %>% count(tree, Survived) %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##   tree    `0`   `1`
## * <chr> <int> <int>
## 1 no      138    37
## 2 yes      24    68
```

Exercise 6

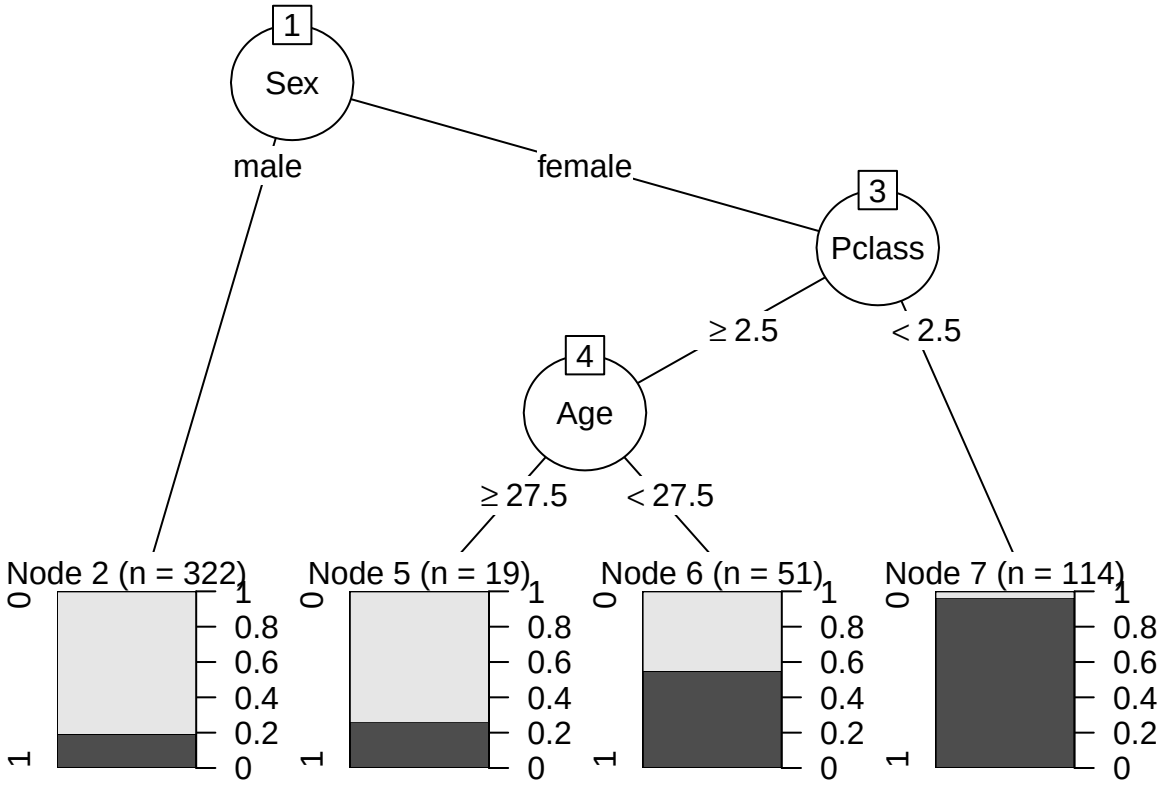
```
Titanic_train %>%
  filter(!is.na(Titanic_train$Age)) -> Titanic_train2
Titanic_test %>%
  filter(!is.na(Titanic_test$Age)) -> Titanic_test2


logit_mod6 <- glm(Survived ~ Pclass + Sex + Age, data = Titanic_train2, family = 'binomial')
tidy(logit_mod6)
```

```
##          term    estimate   std.error  statistic      p.value
## 1 (Intercept)  4.69299343 0.602182541   7.793307 6.527774e-15
## 2      Pclass -1.15981402 0.165626890  -7.002571 2.513078e-12
## 3     Sexmale -2.66313941 0.247843014 -10.745267 6.238101e-27
## 4         Age -0.03199468 0.009636951  -3.320001 9.001728e-04
```

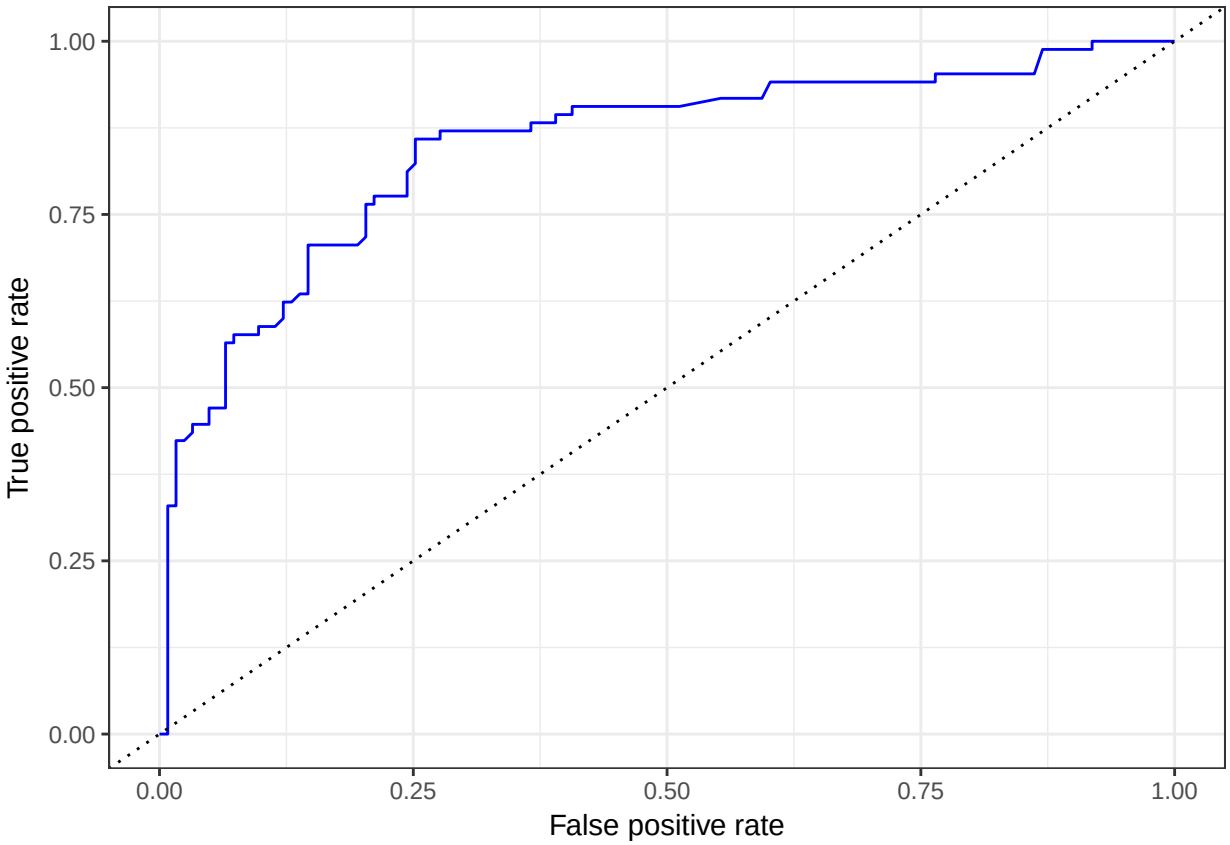
```
test_logit6 <- predict(logit_mod6, newdata = Titanic_test2, type = 'response')

tree_mod6 <- rpart(Survived ~ Pclass + Sex + Age, data = Titanic_train2)
tree_mod6
```

```
## n= 506
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 506 205 0 (0.59486166 0.40513834)
##    2) Sex=male 322  62 0 (0.80745342 0.19254658) *
##    3) Sex=female 184  41 1 (0.22282609 0.77717391)
##      6) Pclass>=2.5 70  33 0 (0.52857143 0.47142857)
##       12) Age>=27.5 19   5 0 (0.73684211 0.26315789) *
##       13) Age< 27.5 51  23 1 (0.45098039 0.54901961) *
##      7) Pclass< 2.5 114   4 1 (0.03508772 0.96491228) *
```

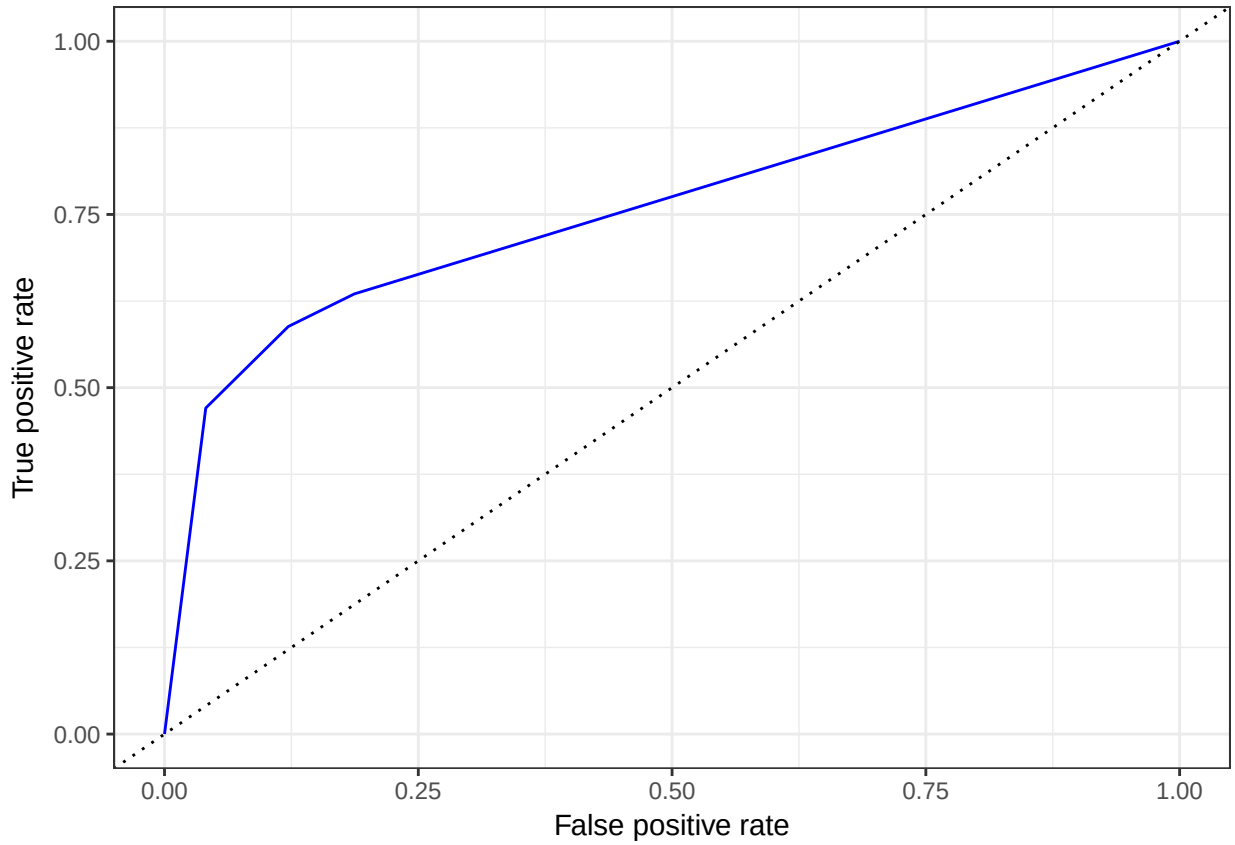
```
plot(as.party(tree_mod6))
```




```
test_tree6 <- predict(tree_mod6, newdata = Titanic_test2)[,2]



pred_logit6 <- prediction(predictions = test_logit6, labels = Titanic_test2$Survived)
pred_tree6 <- prediction(predictions = test_tree6, labels = Titanic_test2$Survived)

perf_logit6 <- performance(pred_logit6, measure = 'tpr', x.measure = 'fpr')
perf_logit6_tbl <- tibble(perf_logit6@x.values[[1]], perf_logit6@y.values[[1]])
names(perf_logit6_tbl) <- c('fpr', 'tpr')

perf_tree6 <- performance(pred_tree6, measure = 'tpr', x.measure = 'fpr')
perf_tree6_tbl <- tibble(perf_tree6@x.values[[1]], perf_tree6@y.values[[1]])
names(perf_tree6_tbl) <- c('fpr', 'tpr')

plot_roc <- function(perf_tbl) {
  p <- ggplot(data = perf_tbl, aes(x = fpr, y = tpr)) +
  geom_line(color = 'blue') +
  geom_abline(intercept = 0, slope = 1, lty = 3) +
  labs(x = 'False positive rate', y = 'True positive rate') +
  theme_bw()
  return(p)
}

plot_roc(perf_logit6_tbl)
```




```
plot_roc(perf_tree6_tbl)
```




```
auc_logit6 <- performance(prediction.obj = pred_logit6, measure = "auc")
auc_tree6 <- performance(prediction.obj = pred_tree6, measure = "auc")

# extract the AUC value
auc_logit6@y.values[[1]]
```

```
## [1] 0.8472023
```

```
auc_tree6@y.values[[1]]
```

```
## [1] 0.7571497
```

```
Titanic_test2 %>%
  mutate(pred_logit6 = test_logit6,
         pred_tree6 = test_tree6) -> Titanic_test2

cutoff = 0.2
Titanic_test2 %>%
  mutate(logit6 = case_when(pred_logit6 >= cutoff ~ "yes",
                    pred_logit6 < cutoff ~ "no"),
         tree6 =  case_when(pred_tree6 >= cutoff ~ "yes",
                            pred_tree6 < cutoff ~ "no")) -> Titanic_test2


Titanic_test2 %>% count(logit6, Survived) %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##   logit6   `0`   `1`
```

```
## * <chr>  <int> <int>
## 1 no        76    10
## 2 yes       47    75
```

```
Titanic_test2 %>% count(tree6, Survived) %>% spread(Survived, n)
```

```
## # A tibble: 2 x 3
##   tree6   `0`   `1`
## * <chr> <int> <int>
## 1 no      100    31
## 2 yes      23    54
```