INFX 573 Problem Set 8 - Prediction

Vighnesh Misal

Due: Tuesday, November 26, 2019

Collaborators: Ashish Anand

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset8.Rmd` file from Canvas. Open `problemset8.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset8.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps7.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```r
# Load standard libraries
library(tidyverse)
library(gridExtra)
library(MASS)
library(pROC)
library(arm)
library(randomForest)
library(dplyr)
library(Metrics)
library(ROCR)
```

In this problem set we will use the and datasets used previously in class. The flights dataset (via the the library) contains information on flight delays and weather. Titanic text file contains data about the survival of passengers aboard the Titanic. Table contains a description of this data.

As part of this assignment, we will evaluate the performance of several statistical learning methods. We will fit our learning models using a set of observations and measure its performance on a set of observations.

Discuss the advantages of using a training/test split when evaluating statistical models.

Training/test split ensures that predictions are on data that is unknown. This ensures that the model can be improved upon later.

Predictions with a continuous output variable

Load in the flights dataset. Join the flights data to the weather data based on the departure location, date, and hour of the flight. Exclude data entries which cannot be joined to weather data. Copy the joined data so we can refer to it later.

```
# Load data
library(nycflights13)

merged_columns <- c("origin","year", "month", "day", "time_hour")
dataset1 <- merge(weather,flights, by = merged_columns)
```

From the joined data, keep only the following columns as we build our first model: departure delay, origin, departure time, temperature, wind speed, precipitation, and visibility. Omit observations that do not have all of these variables present.

```
subset1 <- subset(dataset1,select = c("dep_delay", "dep_time","origin",
"temp", "wind_speed", "precip", "visib"))

subset1<-subset1[complete.cases(subset1),]
```

Split your data into a and set based on an 80-20 split. In other words, 80% of the observations will be in the training set and 20% will be in the test set. Remember to set the random seed.

```
sample1 <- floor(0.80 * nrow(subset1))

set.seed(123)
train_independent <- sample(seq_len(nrow(subset1)), size = sample1)

train_model <- subset1[train_independent, ]
test_model <- subset1[-train_independent, ]
```

Build a linear regression model to predict departure delay using the subset of variables indicated in (3.). What is the RMSE on the training set? What is the RMSE on the test set? Which is higher and is this expected?

```
training_model1 <- lm( dep_delay ~ dep_time + precip + wind_speed,
train_model)
summary(training_model1)

##
## Call:
```

```
## lm(formula = dep_delay ~ dep_time + precip + wind_speed, data =
train_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -159.83  -18.49   -7.87    1.44 1304.61
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.768e+01  2.544e-01 -69.518  < 2e-16 ***
## dep_time     2.119e-02  1.555e-04 136.252  < 2e-16 ***
## precip       1.203e+02  2.538e+00  47.411  < 2e-16 ***
## wind_speed   1.057e-01  1.371e-02   7.708 1.28e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.49 on 261514 degrees of freedom
## Multiple R-squared:  0.07636,    Adjusted R-squared:  0.07635
## F-statistic:  7207 on 3 and 261514 DF,  p-value: < 2.2e-16
```

```
training_model1_y <- predict(training_model1)

RMSE_Test_set <- sqrt(mean((training_model1_y - train_model$dep_delay)^2))
RMSE_Test_set
```

```
## [1] 38.48656
```

```
testing_model1 <- lm( dep_delay ~ dep_time + precip + wind_speed, test_model)
summary(testing_model1)
```

```
##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + wind_speed, data =
test_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.88  -18.97   -8.24    1.32 1118.60
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.710e+01  5.229e-01  -32.70   <2e-16 ***
## dep_time     2.155e-02  3.179e-04   67.80   <2e-16 ***
## precip       1.060e+02  5.222e+00   20.29   <2e-16 ***
## wind_speed   4.239e-02  2.808e-02    1.51    0.131
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.44 on 65376 degrees of freedom
## Multiple R-squared:  0.07301,    Adjusted R-squared:  0.07296
## F-statistic:  1716 on 3 and 65376 DF,  p-value: < 2.2e-16
```

Since RMSE_test(39.44) is higher than RMSE_training(38.48656), an assumption of model overfitting can be made.

Now, improve upon these prediction results by including additional variables in your model. Make sure you keep at least 95% of original data (i.e. about 320K observations across both the training and test datasets). Do not include the arrival time, scheduled arrival time, or the arrival delay in your model. Use the same observations as above for the training and test sets (i.e. keep the same rows but add different variables/columns at your discretion). Can you improve upon the training RMSE? Once you have a model that you feel adequately improves the training RMSE, does your model improve the test RMSE? Which variables did you include in your model?

```
training_model2 <- lm( dep_delay ~ dep_time + precip + wind_speed + temp,
train_model)
summary(training_model2)

##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + wind_speed + temp,
##     data = train_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -160.03  -18.68   -8.00    1.82 1305.79
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.303e+01  3.532e-01  -65.21   <2e-16 ***
## dep_time     2.085e-02  1.562e-04  133.44   <2e-16 ***
## precip       1.197e+02  2.536e+00   47.20   <2e-16 ***
## wind_speed   1.523e-01  1.387e-02   10.99   <2e-16 ***
## temp         9.304e-02  4.266e-03   21.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.45 on 261513 degrees of freedom
## Multiple R-squared:  0.07804,    Adjusted R-squared:  0.07802
## F-statistic:  5534 on 4 and 261513 DF,  p-value: < 2.2e-16

training_model2_y <- predict(training_model2)

sqrt(mean((training_model2_y - train_model$dep_delay)^2))

## [1] 38.45161

testing_model2 <- lm( dep_delay ~ dep_time + precip + wind_speed + temp,
test_model)
summary(testing_model2)
```

```
##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + wind_speed + temp,
##     data = test_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.31  -19.19   -8.36    1.68 1119.65
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.272e+01  7.215e-01 -31.497  < 2e-16 ***
## dep_time     2.115e-02  3.196e-04  66.181  < 2e-16 ***
## precip       1.046e+02  5.218e+00  20.054  < 2e-16 ***
## wind_speed   9.088e-02  2.838e-02   3.202  0.00136 **
## temp         9.875e-02  8.738e-03  11.301  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.4 on 65375 degrees of freedom
## Multiple R-squared:  0.07481,    Adjusted R-squared:  0.07476
## F-statistic:  1322 on 4 and 65375 DF,  p-value: < 2.2e-16
```

```
testing_model2_y <- predict(testing_model2)
sqrt(mean((testing_model2_y - test_model$dep_delay)^2))
```

```
## [1] 39.39768
```

```
training_model3 <- lm( dep_delay ~ dep_time + precip +  temp, train_model)
summary(training_model3)
```

```
##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + temp, data = train_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -162.73  -18.57   -8.00    1.71 1304.89
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.126e+01  3.141e-01  -67.67   <2e-16 ***
## dep_time     2.108e-02  1.548e-04  136.20   <2e-16 ***
## precip       1.206e+02  2.535e+00   47.57   <2e-16 ***
## temp         8.582e-02  4.216e-03   20.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.46 on 261514 degrees of freedom
## Multiple R-squared:  0.07761,    Adjusted R-squared:  0.0776
## F-statistic:  7335 on 3 and 261514 DF,  p-value: < 2.2e-16
```

```
training_model3_y <- predict(training_model3)

sqrt(mean((training_model3_y - train_model$dep_delay)^2))

## [1] 38.46048

testing_model3 <- lm( dep_delay ~ dep_time + precip + wind_speed + temp,
test_model)
summary(testing_model3)

##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + wind_speed + temp,
##     data = test_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.31  -19.19   -8.36    1.68 1119.65
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.272e+01  7.215e-01 -31.497  < 2e-16 ***
## dep_time     2.115e-02  3.196e-04  66.181  < 2e-16 ***
## precip       1.046e+02  5.218e+00  20.054  < 2e-16 ***
## wind_speed   9.088e-02  2.838e-02   3.202  0.00136 **
## temp         9.875e-02  8.738e-03  11.301  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.4 on 65375 degrees of freedom
## Multiple R-squared:  0.07481,    Adjusted R-squared:  0.07476
## F-statistic:  1322 on 4 and 65375 DF,  p-value: < 2.2e-16

testing_model3_y <- predict(testing_model3)
sqrt(mean((testing_model3_y - test_model$dep_delay)^2))

## [1] 39.39768

training_model4 <- lm( dep_delay ~ dep_time + visib + origin, train_model)
summary(training_model4)

##
## Call:
## lm(formula = dep_delay ~ dep_time + visib + origin, data = train_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -67.60  -18.41   -7.97    2.24 1302.81
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  6.3585302  0.4160346   15.28   <2e-16 ***
## dep_time     0.0220997  0.0001545  143.01   <2e-16 ***
## visib       -2.2541856  0.0379704  -59.37   <2e-16 ***
## originJFK   -4.3007881  0.1808588  -23.78   <2e-16 ***
## originLGA   -3.9245684  0.1842090  -21.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.35 on 261513 degrees of freedom
## Multiple R-squared:  0.08295,    Adjusted R-squared:  0.08294
## F-statistic:  5914 on 4 and 261513 DF,  p-value: < 2.2e-16
```

```
training_model4_y <- predict(training_model4)

sqrt(mean((training_model4_y - train_model$dep_delay)^2))
```

```
## [1] 38.34902
```

```
testing_model4 <- lm( dep_delay ~ dep_time + precip + wind_speed + temp,
test_model)
summary(testing_model4)
```

```
##
## Call:
## lm(formula = dep_delay ~ dep_time + precip + wind_speed + temp,
##     data = test_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.31  -19.19   -8.36    1.68 1119.65
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.272e+01  7.215e-01 -31.497  < 2e-16 ***
## dep_time     2.115e-02  3.196e-04  66.181  < 2e-16 ***
## precip       1.046e+02  5.218e+00  20.054  < 2e-16 ***
## wind_speed   9.088e-02  2.838e-02   3.202  0.00136 **
## temp         9.875e-02  8.738e-03  11.301  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.4 on 65375 degrees of freedom
## Multiple R-squared:  0.07481,    Adjusted R-squared:  0.07476
## F-statistic:  1322 on 4 and 65375 DF,  p-value: < 2.2e-16
```

```
testing_model4_y <- predict(testing_model4)
sqrt(mean((testing_model4_y - test_model$dep_delay)^2))
```

```
## [1] 39.39768
```

The model with the lowest observed RMSE is the best model i.e. the model with variables dep_time + visib + origin

Predictions with a categorical output (classification)

Load in the titanic data. Split your data into a and set based on an 80-20 split. In other words, 80% of the observations will be in the training set and 20% will be in the test set. Remember to set the random seed.

```
titanic_data <- read.csv('titanic.csv')

sample_size <- floor(0.80 * nrow(titanic_data))

set.seed(101)

titanic_training_data <- sample(seq_len(nrow(titanic_data)), size =
sample_size)

titanic_training <- titanic_data[titanic_training_data, ]
titanic_testing <- titanic_data[-titanic_training_data, ]
```

In this problem set our goal is to predict the survival of passengers. First, let's train a logistic regression model for survival that controls for the socioeconomic status of the passenger.

Fit the model described above (i.e. one that only takes into account socioeconomic status) using the function in R.

```
titanic_training1<- glm(survived ~ pclass + fare, family = binomial, data =
titanic_training)

summary(titanic_training1)

##
## Call:
## glm(formula = survived ~ pclass + fare, family = binomial, data =
titanic_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6178  -0.7798  -0.7563   1.0568   1.6786
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.871989   0.260952   3.342 0.000833 ***
## pclass      -0.666864   0.096932  -6.880    6e-12 ***
## fare         0.002998   0.001737   1.725 0.084445 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1379.6  on 1045  degrees of freedom
## Residual deviance: 1279.5  on 1043  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1285.5
##
## Number of Fisher Scoring iterations: 4
```

```r
exp(coef(titanic_training1)[2])
```

```
##    pclass
## 0.5133159
```

What might you conclude based on this model about the probability of survival for lower class passengers?

It's evident from the model that lower class passengers had a lower chance of survival owing to their socio-economic status. This means that people from highe class were given a preference during the evacuation.

Next, let's consider the performance of this model.

Predict the survival of passengers for each observation in your test set using the model fit in Problem 2. Save these predictions as .

```r
titanic_testing1 <-  glm(survived ~ pclass + fare, family = binomial, data =
titanic_testing)
summary(titanic_testing1)
```

```
##
## Call:
## glm(formula = survived ~ pclass + fare, family = binomial, data =
titanic_testing)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6011  -0.8930  -0.8049   1.1173   1.6090
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.115042   0.644348  -0.179  0.85830
## pclass      -0.328167   0.217429  -1.509  0.13122
## fare         0.017926   0.006654   2.694  0.00705 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 357.68  on 261  degrees of freedom
## Residual deviance: 314.53  on 259  degrees of freedom
## AIC: 320.53
##
## Number of Fisher Scoring iterations: 5

yhat<- predict(titanic_testing1, titanic_testing, type="response")
```

Use a threshold of 0.5 to classify predictions. What is the number of false positives on the test data? Interpret this in your own words.

```
classify_prediction <- data.frame(titanic_testing$survived, yhat)

classify_prediction$classification <- classify_prediction$yhat > 0.5

classify_prediction

##      titanic_testing.survived      yhat classification
## 5                           0 0.9066621           TRUE
## 12                          1 0.9743069           TRUE
## 21                          1 0.6222010           TRUE
## 27                          1 0.7666547           TRUE
## 35                          0 0.5081838           TRUE
## 41                          0 0.5662772           TRUE
## 44                          1 0.7159438           TRUE
## 45                          1 0.8773851           TRUE
## 46                          0 0.5081838           TRUE
## 50                          1 0.9998401           TRUE
## 62                          1 0.7251775           TRUE
## 63                          0 0.6577855           TRUE
## 65                          1 0.6244982           TRUE
## 66                          1 0.6244982           TRUE
## 70                          1 0.6324508           TRUE
## 75                          0 0.6198628           TRUE
## 85                          0 0.6973366           TRUE
## 96                          1 0.7357936           TRUE
## 99                          1 0.8122372           TRUE
## 100                         1 0.5662772           TRUE
## 112                         1 0.9862307           TRUE
## 114                         1 0.9862307           TRUE
## 118                         1 0.6404526           TRUE
## 120                         1 0.8757364           TRUE
## 121                         1 0.5134284           TRUE
## 122                         1 0.8757364           TRUE
## 128                         1 0.6244982           TRUE
## 131                         1 0.6505873           TRUE
## 132                         1 0.6505873           TRUE
## 136                         0 0.5443872           TRUE
## 151                         0 0.3909766          FALSE
## 152                         1 0.7175361           TRUE
## 160                         1 0.6447754           TRUE
```

```
## 165                                1 0.5081838           TRUE
## 166                                1 0.7631759           TRUE
## 168                                1 0.7631759           TRUE
## 188                                1 0.5653964           TRUE
## 190                                0 0.5236285           TRUE
## 191                                1 0.7219803           TRUE
## 198                                0 0.6244982           TRUE
## 202                                0 0.6192819           TRUE
## 205                                1 0.7368812           TRUE
## 208                                1 0.7631759           TRUE
## 213                                0 0.5222868           TRUE
## 221                                1 0.6610233           TRUE
## 225                                0 0.5169172           TRUE
## 227                                0 0.6793295           TRUE
## 228                                1 0.6793295           TRUE
## 230                                1 0.8189101           TRUE
## 231                                1 0.7743281           TRUE
## 233                                0 0.6198628           TRUE
## 236                                1 0.5662772           TRUE
## 245                                0 0.5685873           TRUE
## 250                                1 0.9860777           TRUE
## 251                                1 0.9860777           TRUE
## 254                                1 0.9860777           TRUE
## 257                                1 0.5057194           TRUE
## 274                                1 0.8773851           TRUE
## 279                                1 0.5258638           TRUE
## 282                                1 0.6342895           TRUE
## 285                                1 0.7292669           TRUE
## 289                                1 0.5054021           TRUE
## 292                                1 0.7280264           TRUE
## 293                                1 0.6198628           TRUE
## 294                                1 0.6198628           TRUE
## 300                                0 0.5134284           TRUE
## 303                                1 0.9998401           TRUE
## 307                                0 0.7195602           TRUE
## 308                                0 0.7195602           TRUE
## 314                                0 0.9660469           TRUE
## 320                                1 0.8773851           TRUE
## 325                                1 0.4155324          FALSE
## 329                                0 0.4242653          FALSE
## 349                                0 0.3685757          FALSE
## 350                                1 0.3685757          FALSE
## 361                                1 0.4374522          FALSE
## 363                                1 0.4025334          FALSE
## 368                                0 0.3706642          FALSE
## 369                                0 0.4242653          FALSE
## 384                                0 0.3623402          FALSE
## 388                                1 0.4718850          FALSE
## 403                                1 0.3721637          FALSE
## 415                                0 0.4025334          FALSE
```

```
## 420                              0 0.3623402          FALSE
## 424                              0 0.3685757          FALSE
## 429                              1 0.3748555          FALSE
## 430                              0 0.3685757          FALSE
## 441                              1 0.5972075           TRUE
## 459                              1 0.3582086          FALSE
## 464                              0 0.4485100          FALSE
## 471                              0 0.3658681          FALSE
## 480                              1 0.4934983          FALSE
## 482                              1 0.4934983          FALSE
## 484                              1 0.4551700          FALSE
## 492                              0 0.3771511          FALSE
## 495                              1 0.4730208          FALSE
## 496                              0 0.3794000          FALSE
## 505                              0 0.3685757          FALSE
## 512                              0 0.3548671          FALSE
## 513                              0 0.4421815          FALSE
## 518                              0 0.3685757          FALSE
## 519                              0 0.4718850          FALSE
## 520                              0 0.3706642          FALSE
## 526                              0 0.3582086          FALSE
## 531                              0 0.3582086          FALSE
## 533                              0 0.3685757          FALSE
## 534                              1 0.4025334          FALSE
## 538                              0 0.3685757          FALSE
## 541                              1 0.4242653          FALSE
## 544                              0 0.3582086          FALSE
## 547                              1 0.3685757          FALSE
## 553                              0 0.3582086          FALSE
## 554                              1 0.3582086          FALSE
## 561                              1 0.3685757          FALSE
## 567                              0 0.3771511          FALSE
## 570                              0 0.5972075           TRUE
## 580                              0 0.3582086          FALSE
## 583                              1 0.3801211          FALSE
## 584                              1 0.3801211          FALSE
## 586                              0 0.4242653          FALSE
## 593                              0 0.4319458          FALSE
## 596                              0 0.3680544          FALSE
## 614                              1 0.3180497          FALSE
## 618                              0 0.2742510          FALSE
## 643                              0 0.3689146          FALSE
## 644                              1 0.3689146          FALSE
## 645                              0 0.3689146          FALSE
## 646                              1 0.2769200          FALSE
## 650                              0 0.2742510          FALSE
## 652                              0 0.2748758          FALSE
## 659                              1 0.3198830          FALSE
## 660                              1 0.3198830          FALSE
## 661                              1 0.3198830          FALSE
```

```
## 664                                   0 0.2772791          FALSE
## 667                                   0 0.3014443          FALSE
## 669                                   0 0.2778334          FALSE
## 675                                   1 0.4783202          FALSE
## 677                                   0 0.2767556          FALSE
## 680                                   0 0.3044409          FALSE
## 685                                   0 0.3054067          FALSE
## 686                                   0 0.3076931          FALSE
## 687                                   1 0.2766659          FALSE
## 689                                   0 0.2742360          FALSE
## 691                                   0 0.2769200          FALSE
## 695                                   0 0.2731818          FALSE
## 697                                   0 0.2800418          FALSE
## 714                                   0 0.2766957          FALSE
## 724                                   0 0.2766957          FALSE
## 741                                   0 0.2800418          FALSE
## 745                                   0 0.2855255          FALSE
## 747                                   1 0.2767556          FALSE
## 749                                   0 0.3012397          FALSE
## 751                                   0 0.2772791          FALSE
## 755                                   0 0.3392544          FALSE
## 768                                   0 0.2772791          FALSE
## 771                                   0 0.2749651          FALSE
## 773                                   0 0.2772791          FALSE
## 775                                   0 0.2772791          FALSE
## 779                                   1 0.2940261          FALSE
## 780                                   0 0.2767556          FALSE
## 781                                   1 0.2778334          FALSE
## 790                                   0 0.2748758          FALSE
## 792                                   0 0.2748908          FALSE
## 807                                   0 0.3814683          FALSE
## 808                                   0 0.3814683          FALSE
## 811                                   0 0.3814683          FALSE
## 813                                   0 0.2767556          FALSE
## 815                                   0 0.2767259          FALSE
## 818                                   0 0.2778334          FALSE
## 830                                   0 0.4356607          FALSE
## 835                                   0 0.2789588          FALSE
## 836                                   0 0.2778334          FALSE
## 837                                   0 0.2843385          FALSE
## 844                                   0 0.3226521          FALSE
## 846                                   1 0.3067393          FALSE
## 854                                   0 0.2749651          FALSE
## 859                                   1 0.4783202          FALSE
## 860                                   0 0.2731818          FALSE
## 862                                   0 0.2773840          FALSE
## 871                                   1 0.2773840          FALSE
## 875                                   1 0.2772493          FALSE
## 884                                   0 0.2771297          FALSE
## 885                                   0 0.2771297          FALSE
```

```
## 888                              1 0.2780582          FALSE
## 889                              0 0.2722780          FALSE
## 896                              1 0.2890584          FALSE
## 898                              0 0.2768453          FALSE
## 901                              0 0.3364472          FALSE
## 918                              1 0.2975424          FALSE
## 922                              0 0.2749651          FALSE
## 928                              0 0.3014443          FALSE
## 939                              0 0.2771297          FALSE
## 940                              0 0.2929418          FALSE
## 943                              0 0.2748758          FALSE
## 953                              0 0.2768453          FALSE
## 960                              0 0.2773840          FALSE
## 968                              0 0.2767556          FALSE
## 975                              0 0.3076931          FALSE
## 976                              0 0.3076931          FALSE
## 991                              0 0.2773840          FALSE
## 994                              1 0.2767108          FALSE
## 998                              1 0.2748758          FALSE
## 1002                             1 0.3356473          FALSE
## 1004                             1 0.3356473          FALSE
## 1015                             0 0.2778334          FALSE
## 1019                             0 0.2778334          FALSE
## 1026                             1 0.2940261          FALSE
## 1028                             0 0.2778334          FALSE
## 1032                             0 0.2778334          FALSE
## 1034                             1 0.2768453          FALSE
## 1046                             0 0.2767556          FALSE
## 1048                             1 0.2748758          FALSE
## 1049                             1 0.3063266          FALSE
## 1051                             1 0.3063266          FALSE
## 1053                             0 0.2772791          FALSE
## 1058                             1 0.2894579          FALSE
## 1069                             0 0.2713615          FALSE
## 1071                             0 0.2770399          FALSE
## 1074                             0 0.2767556          FALSE
## 1079                             1 0.2772195          FALSE
## 1080                             1 0.2768453          FALSE
## 1085                             0 0.2791092          FALSE
## 1095                             1 0.2801170          FALSE
## 1100                             0 0.3270094          FALSE
## 1118                             0 0.2773840          FALSE
## 1122                             1 0.3320922          FALSE
## 1129                             0 0.2772791          FALSE
## 1137                             0 0.2780582          FALSE
## 1145                             0 0.3595231          FALSE
## 1147                             0 0.3595231          FALSE
## 1158                             0 0.3236159          FALSE
## 1160                             1 0.2778334          FALSE
## 1166                             0 0.2748758          FALSE
```

```
## 1177                              0 0.5367441          TRUE
## 1184                              0 0.2773840         FALSE
## 1186                              0 0.3293975         FALSE
## 1187                              0 0.3293975         FALSE
## 1190                              1 0.3099890         FALSE
## 1191                              1 0.2830787         FALSE
## 1194                              0 0.2766659         FALSE
## 1200                              0 0.2778334         FALSE
## 1207                              0 0.3544823         FALSE
## 1215                              0 0.2800418         FALSE
## 1224                              1 0.2760386         FALSE
## 1228                              0 0.2843082         FALSE
## 1230                              0 0.2800418         FALSE
## 1235                              0 0.2742510         FALSE
## 1238                              0 0.2769200         FALSE
## 1241                              1 0.2795151         FALSE
## 1245                              1 0.2795151         FALSE
## 1248                              1 0.3076931         FALSE
## 1250                              0 0.2767556         FALSE
## 1256                              0 0.2748908         FALSE
## 1258                              1 0.3044409         FALSE
## 1259                              1 0.3044409         FALSE
## 1265                              0 0.3016172         FALSE
## 1276                              0 0.3149957         FALSE
## 1279                              0 0.2771297         FALSE
## 1288                              0 0.2767556         FALSE
## 1291                              1 0.2740726         FALSE
## 1297                              0 0.2800418         FALSE
## 1302                              0 0.2748758         FALSE
```

```
False_pos <- nrow(subset(classify_prediction, titanic_testing$survived ==0 &
classification ==1))
False_neg <- nrow(subset(classify_prediction, titanic_testing$survived ==1 &
classification ==0))
True_pos <- nrow(subset(classify_prediction, titanic_testing$survived ==1 &
classification ==1))
True_neg <- nrow(subset(classify_prediction, titanic_testing$survived ==0 &
classification ==0))

false_positive_rate = False_pos/(False_pos+True_neg)
false_positive_rate
```

```
## [1] 0.1466667
```

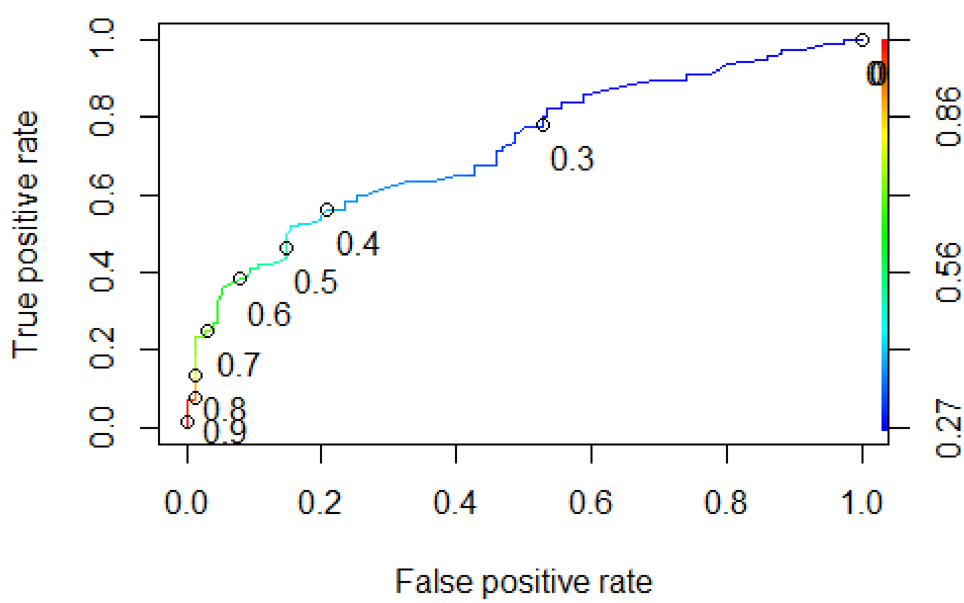
The model is not highly susceptible to providing false positives as it currently contains 22 false_pos. The false positive rate is about 14.67%.

Using the function, plot the ROC curve for this model. Discuss what you find.

```
predictions_titanic <- predict(titanic_testing1, newdata=titanic_testing,
type="response")

ROCR_pred_titanic  <- prediction(predictions_titanic ,
titanic_testing$survived)
ROCR_perf_titanic  <- performance(ROCR_pred_titanic , measure = "tpr",
x.measure = "fpr")

plot(ROCR_perf_titanic , colorize = TRUE, text.adj = c(-0.2,1.7),
print.cutoffs.at = seq(0,1,0.1))
```



A much lower threshold value would be more suitable as the graph with the current threshold of 0.5 doesn't give us much information.

Suppose we use the data to construct a new predictor variable based on a passenger's listed title (i.e. Mr., Mrs., Miss., Master). Why might this be an interesting variable to help predict passenger survival?

Use the following custom function to add this predictor to your dataset.

```
# Making a feature that includes more titles
getTitles <- function(name) {
  for (title in c("Master", "Miss", "Mrs.", "Mr.")) {
    if (grepl(title, name)) {
      return(title)
    }
  }
  return("Nothing")
}

result1 <- numeric(length(titanic_training$name))
for (i in seq_along(titanic_training$name)) {
  result1[i] <- getTitles(titanic_training$name[i])
}

title1 <- as.data.frame(result1)
title1_training <- cbind(titanic_training,title1)

result2 <- numeric(length(titanic_testing$name))
for (i in seq_along(titanic_testing$name)) {
  result2[i] <- getTitles(titanic_testing$name[i])
}

title2 <- as.data.frame(result2)
title2_testing <- cbind(titanic_testing,title2)
```

The predictor takes into consideration the gender, age and marital status of the passengers. Generally women, elderly and children are given preference during an evacuation which can cause the survival rate to be skewed. By performing an analysis of the passenger title we can determine key charecteristics of the passenger related to survival.

Fit a second logistic regression model including this new feature. Use the function to look at the model. Did this new feature improve the model?

```
title.train<- glm(survived ~ pclass + result1 +fare, family = binomial, data
= title1_training)
summary(title.train)
```

```
##
## Call:
## glm(formula = survived ~ pclass + result1 + fare, family = binomial,
##     data = title1_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0931  -0.6572  -0.4132   0.6588   2.2780
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.75655    0.46235   5.962 2.49e-09 ***
## pclass         -1.00033    0.12042  -8.307  < 2e-16 ***
## result1Miss     0.37452    0.32544   1.151  0.24981
## result1Mr.     -2.15689    0.32119  -6.715 1.88e-11 ***
## result1Mrs.     0.71478    0.35475   2.015  0.04392 *
## result1Nothing -1.70259    0.54513  -3.123  0.00179 **
## fare           -0.00205    0.00194  -1.057  0.29058
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1379.57  on 1045  degrees of freedom
## Residual deviance:  986.35  on 1039  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1000.3
##
## Number of Fisher Scoring iterations: 4
```

Comment on the overall fit of this model. For example, you might consider exploring when misclassification occurs.

```
title.test2 <-  glm(survived ~ pclass + result2 +fare, family = binomial,
data = title2_testing)
summary(title.test2)
```

```
##
## Call:
## glm(formula = survived ~ pclass + result2 + fare, family = binomial,
##     data = title2_testing)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5594  -0.5985  -0.3964   0.5780   2.2727
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.171814   1.138679   1.907  0.05648 .
## pclass         -0.841822   0.267071  -3.152  0.00162 **
## result2Miss     1.128585   0.818425   1.379  0.16790
```

```
## result2Mr.     -2.201954   0.806780  -2.729  0.00635 **
## result2Mrs.     0.896288   0.852245   1.052  0.29295
## result2Nothing -1.811910   1.127206  -1.607  0.10796
## fare            0.006667   0.005296   1.259  0.20810
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 357.68  on 261  degrees of freedom
## Residual deviance: 217.92  on 255  degrees of freedom
## AIC: 231.92
##
## Number of Fisher Scoring iterations: 5
```

```
yhat2<- predict(title.test2, title2_testing, type="response")

classify_prediction2 <- data.frame(title2_testing$survived, yhat2)
classify_prediction2$classification <- classify_prediction2$yhat2 > 0.5
classify_prediction2
```

```
##      title2_testing.survived      yhat2 classification
## 5                          0 0.96219153           TRUE
## 12                         1 0.97686979           TRUE
## 21                         1 0.37248368          FALSE
## 27                         1 0.43420226          FALSE
## 35                         0 0.33293300          FALSE
## 41                         0 0.44573963          FALSE
## 44                         1 0.93905632           TRUE
## 45                         1 0.96627807           TRUE
## 46                         0 0.42435878          FALSE
## 50                         1 0.92715538           TRUE
## 62                         1 0.94002517           TRUE
## 63                         0 0.38601348          FALSE
## 65                         1 0.37333463          FALSE
## 66                         1 0.92958316           TRUE
## 70                         1 0.93040785           TRUE
## 75                         0 0.37162044          FALSE
## 85                         0 0.40210364          FALSE
## 96                         1 0.94114600           TRUE
## 99                         1 0.94958935           TRUE
## 100                        1 0.92345886           TRUE
## 112                        1 0.98540001           TRUE
## 114                        1 0.98540001           TRUE
## 118                        1 0.94470188           TRUE
## 120                        1 0.60088208           TRUE
## 121                        1 0.33466884          FALSE
## 122                        1 0.95760233           TRUE
## 128                        1 0.92958316           TRUE
## 131                        1 0.94555616           TRUE
```

```
## 132                              1 0.93228347          TRUE
## 136                              0 0.34503913         FALSE
## 151                              0 0.29484644         FALSE
## 152                              1 0.41086294         FALSE
## 160                              1 0.94506645          TRUE
## 165                              1 0.33293300         FALSE
## 166                              1 0.43243547         FALSE
## 168                              1 0.94408165          TRUE
## 188                              1 0.93826964          TRUE
## 190                              0 0.33806085         FALSE
## 191                              1 0.95158567          TRUE
## 198                              0 0.37333463         FALSE
## 202                              0 0.37140639         FALSE
## 205                              1 0.94126129          TRUE
## 208                              1 0.95515310          TRUE
## 213                              0 0.33761341         FALSE
## 221                              1 0.94643465          TRUE
## 225                              0 0.33582658         FALSE
## 227                              0 0.39462035         FALSE
## 228                              1 0.93525179          TRUE
## 230                              1 0.95037342          TRUE
## 231                              1 0.95614211          TRUE
## 233                              0 0.37162044         FALSE
## 236                              1 0.35252846         FALSE
## 245                              0 0.35332780         FALSE
## 250                              1 0.95602971          TRUE
## 251                              1 0.98533994          TRUE
## 254                              1 0.98157703          TRUE
## 257                              1 0.33211912         FALSE
## 274                              1 0.90262331          TRUE
## 279                              1 0.43080433         FALSE
## 282                              1 0.93059829          TRUE
## 285                              1 0.94045598          TRUE
## 289                              1 0.91676270          TRUE
## 292                              1 0.94032517          TRUE
## 293                              1 0.37162044         FALSE
## 294                              1 0.92910158          TRUE
## 300                              0 0.42626669         FALSE
## 303                              1 0.99719720          TRUE
## 307                              0 0.41176423         FALSE
## 308                              0 0.41176423         FALSE
## 314                              0 0.63137779          TRUE
## 320                              1 0.96627807          TRUE
## 325                              1 0.82411348          TRUE
## 329                              0 0.17647162         FALSE
## 349                              0 0.16422661         FALSE
## 350                              1 0.84598194          TRUE
## 361                              1 0.17939724         FALSE
## 363                              1 0.85280404          TRUE
## 368                              0 0.16468467         FALSE
```

```
## 369                              0 0.17647162          FALSE
## 384                              0 0.16285856          FALSE
## 388                              1 0.83609768           TRUE
## 403                              1 0.84672607           TRUE
## 415                              0 0.17167912          FALSE
## 420                              0 0.16285856          FALSE
## 424                              0 0.16422661          FALSE
## 429                              1 0.81474346           TRUE
## 430                              0 0.16422661          FALSE
## 441                              1 0.86030441           TRUE
## 459                              1 0.84379764           TRUE
## 464                              0 0.18186409          FALSE
## 471                              0 0.16363266          FALSE
## 480                              1 0.86920738           TRUE
## 482                              1 0.84046227           TRUE
## 484                              1 0.86256629           TRUE
## 492                              0 0.16610731          FALSE
## 495                              1 0.83632980           TRUE
## 496                              0 0.16660049          FALSE
## 505                              0 0.16422661          FALSE
## 512                              0 0.16121777          FALSE
## 513                              0 0.18045062          FALSE
## 518                              0 0.16422661          FALSE
## 519                              0 0.18713017          FALSE
## 520                              0 0.16468467          FALSE
## 526                              0 0.22205441          FALSE
## 531                              0 0.16195164          FALSE
## 533                              0 0.22494704          FALSE
## 534                              1 0.85280404           TRUE
## 538                              0 0.16422661          FALSE
## 541                              1 0.85693960           TRUE
## 544                              0 0.16195164          FALSE
## 547                              1 0.22494704          FALSE
## 553                              0 0.16195164          FALSE
## 554                              1 0.84379764           TRUE
## 561                              1 0.84598194           TRUE
## 567                              0 0.16610731          FALSE
## 570                              0 0.21747861          FALSE
## 580                              0 0.16195164          FALSE
## 583                              1 0.84835574           TRUE
## 584                              1 0.81599805           TRUE
## 586                              0 0.17647162          FALSE
## 593                              0 0.17817365          FALSE
## 596                              0 0.16411225          FALSE
## 614                              1 0.08088773          FALSE
## 618                              0 0.07525739          FALSE
## 643                              0 0.46396582          FALSE
## 644                              1 0.72794036           TRUE
## 645                              0 0.08735708          FALSE
## 646                              1 0.07560416          FALSE
```

```
## 650                              0 0.07525739          FALSE
## 652                              0 0.07533863          FALSE
## 659                              1 0.71163357           TRUE
## 660                              1 0.71163357           TRUE
## 661                              1 0.66173466           TRUE
## 664                              0 0.07565077          FALSE
## 667                              0 0.65452841           TRUE
## 669                              0 0.07572269          FALSE
## 675                              1 0.10165761          FALSE
## 677                              0 0.07558283          FALSE
## 680                              0 0.70611288           TRUE
## 685                              0 0.65610333           TRUE
## 686                              0 0.07956559          FALSE
## 687                              1 0.69560111           TRUE
## 689                              0 0.07525544          FALSE
## 691                              0 0.07560416          FALSE
## 695                              0 0.07511831          FALSE
## 697                              0 0.69692296           TRUE
## 714                              0 0.07557505          FALSE
## 724                              0 0.07557505          FALSE
## 741                              0 0.07600900          FALSE
## 745                              0 0.07671826          FALSE
## 747                              1 0.07558283          FALSE
## 749                              0 0.07873949          FALSE
## 751                              0 0.07565077          FALSE
## 755                              0 0.08358587          FALSE
## 768                              0 0.07565077          FALSE
## 771                              0 0.07535024          FALSE
## 773                              0 0.07565077          FALSE
## 775                              0 0.07565077          FALSE
## 779                              1 0.70226483           TRUE
## 780                              0 0.69563640           TRUE
## 781                              1 0.69605972           TRUE
## 790                              0 0.07533863          FALSE
## 792                              0 0.07534058          FALSE
## 807                              0 0.73186702           TRUE
## 808                              0 0.73186702           TRUE
## 811                              0 0.08895816          FALSE
## 813                              0 0.07558283          FALSE
## 815                              0 0.07557896          FALSE
## 818                              0 0.07572269          FALSE
## 830                              0 0.74793353           TRUE
## 835                              0 0.07586864          FALSE
## 836                              0 0.07572269          FALSE
## 837                              0 0.07656492          FALSE
## 844                              0 0.08147414          FALSE
## 846                              1 0.65662964           TRUE
## 854                              0 0.07535024          FALSE
## 859                              1 0.10165761          FALSE
## 860                              0 0.69422296           TRUE
```

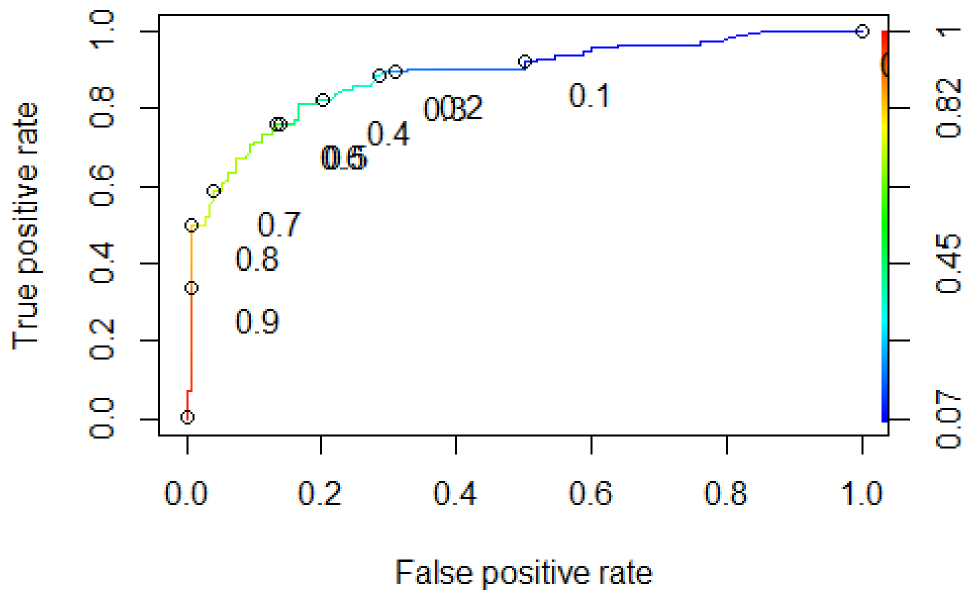
```
## 862                              0 0.69588338          TRUE
## 871                              1 0.69588338          TRUE
## 875                              1 0.07564690         FALSE
## 884                              0 0.07563138         FALSE
## 885                              0 0.07563138         FALSE
## 888                              1 0.07575186         FALSE
## 889                              0 0.07500065         FALSE
## 896                              1 0.70039109          TRUE
## 898                              0 0.07559447         FALSE
## 901                              0 0.45083261         FALSE
## 918                              1 0.07826524         FALSE
## 922                              0 0.07535024         FALSE
## 928                              0 0.07876570         FALSE
## 939                              0 0.07563138         FALSE
## 940                              0 0.65109690          TRUE
## 943                              0 0.07533863         FALSE
## 953                              0 0.07559447         FALSE
## 960                              0 0.07566439         FALSE
## 968                              0 0.69563640          TRUE
## 975                              0 0.07956559         FALSE
## 976                              0 0.65700535          TRUE
## 991                              0 0.07566439         FALSE
## 994                              1 0.69561876          TRUE
## 998                              1 0.64355023          TRUE
## 1002                             1 0.71706397          TRUE
## 1004                             1 0.08312740         FALSE
## 1015                             0 0.64481098          TRUE
## 1019                             0 0.07572269         FALSE
## 1026                             1 0.43278958         FALSE
## 1028                             0 0.07572269         FALSE
## 1032                             0 0.07572269         FALSE
## 1034                             1 0.07559447         FALSE
## 1046                             0 0.07558283         FALSE
## 1048                             1 0.69489481          TRUE
## 1049                             1 0.70679851          TRUE
## 1051                             1 0.65646683          TRUE
## 1053                             0 0.07565077         FALSE
## 1058                             1 0.70054273          TRUE
## 1069                             0 0.07488127         FALSE
## 1071                             0 0.07561973         FALSE
## 1074                             0 0.07558283         FALSE
## 1079                             1 0.69581875          TRUE
## 1080                             1 0.69567169          TRUE
## 1085                             0 0.07588814         FALSE
## 1095                             1 0.64577741          TRUE
## 1100                             0 0.71411273          TRUE
## 1118                             0 0.07566439         FALSE
## 1122                             1 0.44903124         FALSE
## 1129                             0 0.07565077         FALSE
## 1137                             0 0.64490641          TRUE
```

```
## 1145                       0 0.46021644          FALSE
## 1147                       0 0.67629929           TRUE
## 1158                       0 0.08159687          FALSE
## 1160                       1 0.69605972           TRUE
## 1166                       0 0.07533863          FALSE
## 1177                       0 0.10988589          FALSE
## 1184                       0 0.07566439          FALSE
## 1186                       0 0.08233267          FALSE
## 1187                       0 0.08233267          FALSE
## 1190                       1 0.70812080           TRUE
## 1191                       1 0.07640208          FALSE
## 1194                       0 0.07557118          FALSE
## 1200                       0 0.07572269          FALSE
## 1207                       0 0.45818823          FALSE
## 1215                       0 0.07600900          FALSE
## 1224                       1 0.69535401           TRUE
## 1228                       0 0.69857508           TRUE
## 1230                       0 0.07600900          FALSE
## 1235                       0 0.07525739          FALSE
## 1238                       0 0.07560416          FALSE
## 1241                       1 0.42632304          FALSE
## 1245                       1 0.64552329           TRUE
## 1248                       1 0.65700535           TRUE
## 1250                       0 0.07558283          FALSE
## 1256                       0 0.07534058          FALSE
## 1258                       1 0.70611288           TRUE
## 1259                       1 0.65572083           TRUE
## 1265                       0 0.07878786          FALSE
## 1276                       0 0.08049825          FALSE
## 1279                       0 0.07563138          FALSE
## 1288                       0 0.07558283          FALSE
## 1291                       1 0.64320604           TRUE
## 1297                       0 0.07600900          FALSE
## 1302                       0 0.07533863          FALSE
```

```
False_pos2 <- nrow(subset(classify_prediction2, title2_testing$survived ==0 &
classification ==1))
False_neg2 <- nrow(subset(classify_prediction2, title2_testing$survived ==1 &
classification ==0))
True_pos2 <- nrow(subset(classify_prediction2, title2_testing$survived ==1 &
classification ==1))
True_neg2 <- nrow(subset(classify_prediction2, title2_testing$survived ==0 &
classification ==0))

predictions2 <- predict(title.test2, newdata=title2_testing, type="response")
ROCR_pred_2 <- prediction(predictions2, title2_testing$survived)
ROCR_perf_2 <- performance(ROCR_pred_2, measure = "tpr", x.measure = "fpr")

plot(ROCR_perf_2, colorize = TRUE, text.adj = c(-1,1.7), print.cutoffs.at =
seq(0,1,0.1))
```



Improved AUC signifies the overall fit for the model has improved

Predict the survival of passengers for each observation in your test data using the new model. Save these predictions as .

```
title.test <-  glm(survived ~ pclass + result2 +fare, family = binomial, data
= title2_testing)
summary(title.test)

##
## Call:
## glm(formula = survived ~ pclass + result2 + fare, family = binomial,
##     data = title2_testing)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5594  -0.5985  -0.3964   0.5780   2.2727
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.171814   1.138679   1.907  0.05648 .
## pclass         -0.841822   0.267071  -3.152  0.00162 **
## result2Miss     1.128585   0.818425   1.379  0.16790
## result2Mr.     -2.201954   0.806780  -2.729  0.00635 **
## result2Mrs.     0.896288   0.852245   1.052  0.29295
## result2Nothing -1.811910   1.127206  -1.607  0.10796
```

```
## fare            0.006667   0.005296   1.259  0.20810
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 357.68  on 261  degrees of freedom
## Residual deviance: 217.92  on 255  degrees of freedom
## AIC: 231.92
##
## Number of Fisher Scoring iterations: 5
```

```
yhat2<- predict(title.test, title2_testing, type="response")
```

Random forests

Another very popular classifier used in data science is called a .

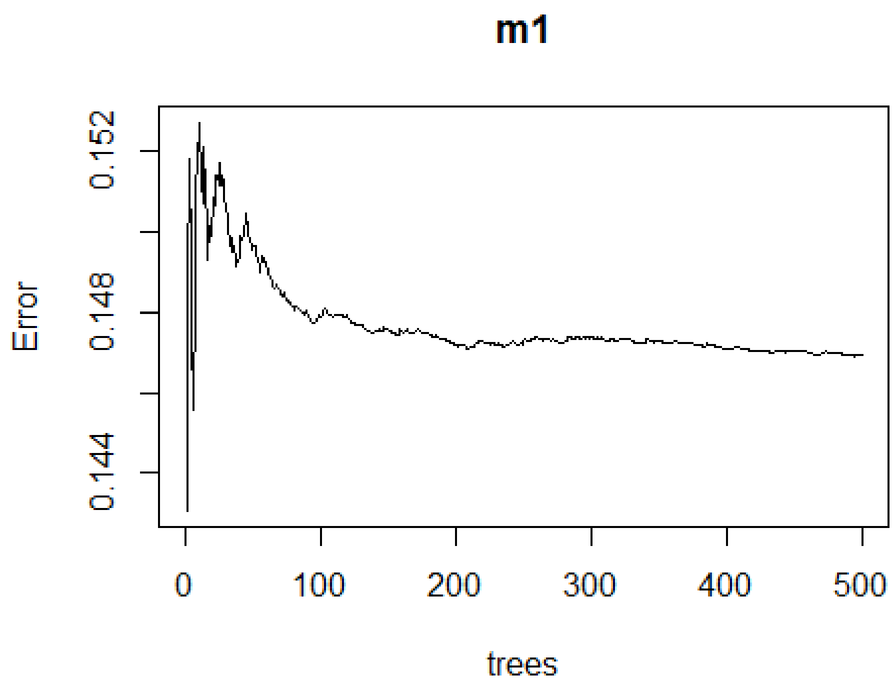
Use the function to fit a random forest model with passenger class and title as predictors. Make predictions for the test set using the random forest model. Save these predictions as .

```
set.seed(171)

m1 <- randomForest(formula =  survived ~ pclass + result1,data =
title1_training)

## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?

plot(m1)
```




```
m1_test <- randomForest(formula =  survived ~ pclass + result2,  data =
title2_testing)

## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

Develop your own random forest model (i.e. add/remove variables at your discretion), attempting to improve the model performance. Make predictions for the test set using your new random forest model. Save these predictions as .

Compare the accuracy of each of the classification models from this problem set using ROC curves. Comment on which statistical learning method works best for predicting survival of the Titanic passengers.