Project_3

Laney Huang

May 5, 2017

Reading Data:

```
ti.train <- read.csv("train.csv", header=TRUE)
ti.train <- ti.train[, -c(1, 4, 9, 11)]
ti.train$Survived <- factor(ti.train$Survived, levels=c(0, 1)) #0 is no
ti.train$Pclass <- factor(ti.train$Pclass, levels=c(1, 2,3),
                          labels = c("upper","middle","lower"))
ti.train[is.na(ti.train$Age) & ti.train$Sex == "male", "Age"] <-
  median(ti.train[ti.train$Sex == "male","Age"], na.rm = TRUE)
ti.train[is.na(ti.train$Age) & ti.train$Sex == "female", "Age"] <-
  median(ti.train[ti.train$Sex == "female","Age"], na.rm = TRUE)
# ti.train <- ti.train[which(!is.na(ti.train$Age)), ]
ti.train <- ti.train[which(ti.train$Embarked != ""), ]
ti.train <- droplevels(ti.train)
```

After reading in the entire data and observing both the dataframe produced and the summary of it, I removed 4 columns. PassengerId and name are simply different for each passenger and just an identification term, not an explanatory variable. Then, I removed the ticket column, which seemed to contain many numbers that are not informative in any way. Finally, after some deliberation, I also removed the Cabin variable, because most of the entries are blank, and will likely have little effect on the prediction of survival. I also dropped the rows in which embarkation was not given, which was a total of 2 entries. For the values under Age which were N/A, I substitued in values for the median age of the particular gender. I did the same below when reading in the test set.

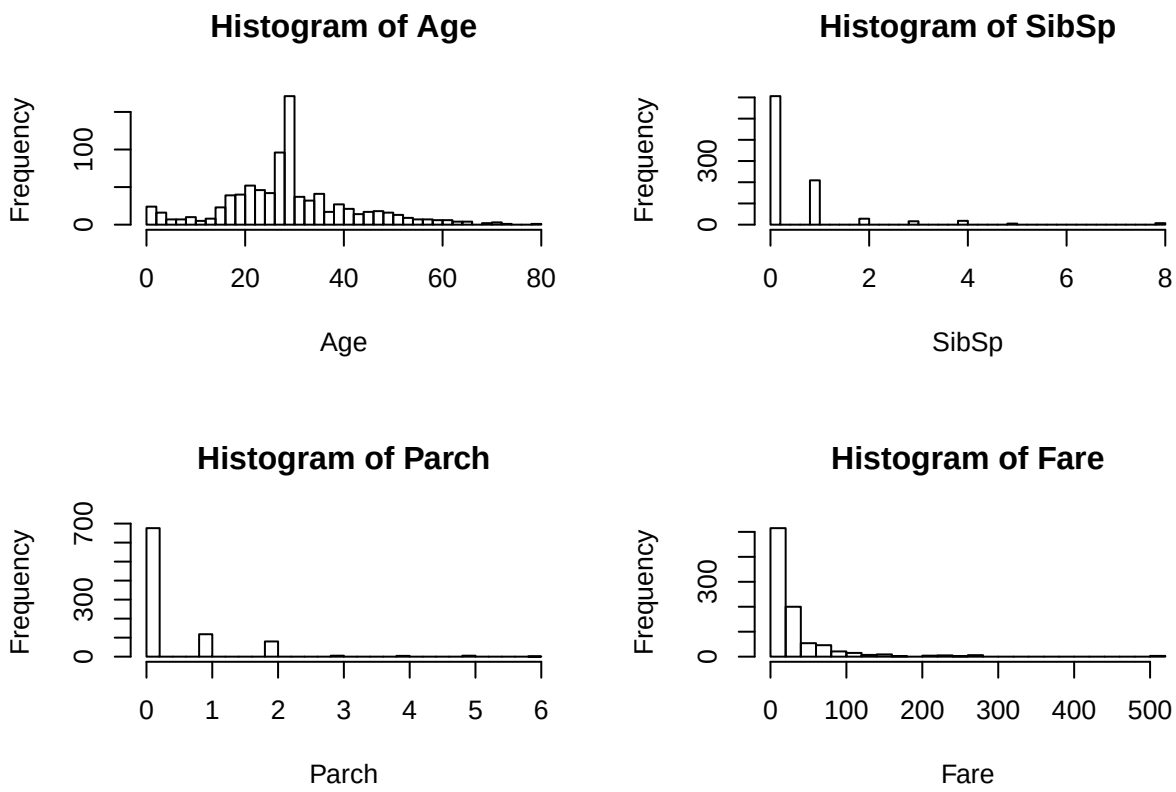
```
ti.test <- read.csv("test.csv", header=TRUE)
ti.test$Pclass <- factor(ti.test$Pclass, levels=c(1, 2,3),
                          labels = c("upper","middle","lower"))
ti.test[is.na(ti.test$Age) & ti.test$Sex == "male", "Age"] <-
  median(ti.train[ti.train$Sex == "male","Age"], na.rm = TRUE)
ti.test[is.na(ti.test$Age) & ti.test$Sex == "female", "Age"] <-
  median(ti.train[ti.train$Sex == "female","Age"], na.rm = TRUE)
ti.test[which(is.na(ti.test$Fare)), "Fare"] <- median(ti.train$Fare)
```

Basic Summary/Analysis:

```
summary(ti.train)
```

```
##  Survived    Pclass         Sex           Age            SibSp
##  0:549    upper :214   female:312   Min.   : 0.42   Min.   :0.0000
##  1:340    middle:184   male  :577   1st Qu.:22.00   1st Qu.:0.0000
##           lower :491                Median :29.00   Median :0.0000
##                                     Mean   :29.40   Mean   :0.5242
##                                     3rd Qu.:35.00   3rd Qu.:1.0000
##                                     Max.   :80.00   Max.   :8.0000
##      Parch             Fare          Embarked
##  Min.   :0.0000   Min.   :  0.000   C:168
##  1st Qu.:0.0000   1st Qu.:  7.896   Q: 77
##  Median :0.0000   Median : 14.454   S:644
##  Mean   :0.3825   Mean   : 32.097
##  3rd Qu.:0.0000   3rd Qu.: 31.000
##  Max.   :6.0000   Max.   :512.329
```

```
par(mfrow=c(2, 2))
cols = colnames(ti.train)
for(i in 1:8) {
  if(is.numeric(ti.train[, i])) {
    hist(ti.train[, i], main = paste("Histogram of", cols[i]), xlab = cols[i], breaks=30)
  }
}
```



Above, I have plotted the histograms for each of the numeric variables. Age seems to have a wide distribution

that is spread relatively symmetrically, likely due to how I replaced the N/A values. SibSp and Parch counts are heavily distributioned on the left end, with lower values. Fares are also concentrated towards lower tickets costs, but since the range extends to 500, there may be extreme, outlier values around that tail.

Below is the pairs plot of all the variables, colored separately by the Survived variable. Pink indicates no survival, blue indicates survival.

```
ggpairs(data= ti.train,
        aes(color = Survived),
        columns = c(2:8),
        lower=list(combo=wrap("facethist", binwidth=5)),
        diag=list(continuous=no_fill))
```

Since survival is the variable of interest, I will focus mostly on its interaction with each of the other variables. Ratio of survivals does seem to differ between passenger classes and sex, and is much more noticeable in the sex category, with males having a higher percentage of deaths. Age doesn't seem to have as much of an effect on survival, as the distributions for both are similar, though there is a slight discrepancy of low ages and their survival. This likely is from the increased survival of children. Sibsp, Parch, and Fare have no noticeable differences between distributions of survival and nonsurvival. For embarked, it appears that those departing from Southampton have a higher low fraction of surviving passengers, which could possibly just be a side effect of its higher count.

Other interesting things to note is that embarkation changes very little relatie to the other variables, so dropping it from the regression may be a consideration. The general range of ages has a higher mean in lower passenger classes. Somehow, also surprisingly, Fare ranges have a higher average for lower passenger classes, which is likely contributed to by outlier entries, but higher passenger classes do not show presence of these outliers. Parch and SibSp are also quite correlated.

Prediction via Logistic Regression

```
ti.fit <- glm(Survived~., data=ti.train, family=binomial)
summary(ti.fit) #783, 803
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = ti.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6213  -0.6100  -0.4207   0.6149   2.4534
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.080577   0.473117   8.625  < 2e-16 ***
## Pclassmiddle -0.919380   0.297500  -3.090  0.00200 **
## Pclasslower  -2.153017   0.298053  -7.224 5.06e-13 ***
## Sexmale      -2.697176   0.201098 -13.412  < 2e-16 ***
## Age          -0.039309   0.007895  -4.979 6.39e-07 ***
## SibSp        -0.323696   0.109147  -2.966  0.00302 **
## Parch        -0.090958   0.118951  -0.765  0.44447
## Fare          0.002279   0.002461   0.926  0.35439
## EmbarkedQ    -0.063883   0.382415  -0.167  0.86733
## EmbarkedS    -0.438869   0.239739  -1.831  0.06716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.82  on 888  degrees of freedom
## Residual deviance:  783.83  on 879  degrees of freedom
## AIC: 803.83
##
## Number of Fisher Scoring iterations: 5
```

First, I performed simple logistic regression on all explanatory variables, and the output of the fit is shown above. However, interaction terms ought to be investigated, and so I refit a model that includes interaction terms between all the variables, which was reasonable in this instance because the number of additional variables is not too high and still computationally viable.

```
ti.inter.fit <- glm(Survived~.*., data=ti.train, family=binomial)
summary(ti.inter.fit) #672, 760
```

```
##
## Call:
## glm(formula = Survived ~ . * ., family = binomial, data = ti.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5029  -0.5646  -0.3432   0.3925   2.9461
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             3.966e+00  1.875e+00   2.115  0.03444 *
## Pclassmiddle           -3.199e-01  2.022e+00  -0.158  0.87433
```

```
## Pclasslower            -3.407e+00  1.637e+00  -2.081  0.03743 *
## Sexmale                -1.475e+00  1.595e+00  -0.925  0.35502
## Age                    -4.948e-02  4.032e-02  -1.227  0.21966
## SibSp                   1.508e+00  1.048e+00   1.438  0.15035
## Parch                   1.791e+00  9.280e-01   1.929  0.05368 .
## Fare                   -1.901e-02  1.596e-02  -1.191  0.23355
## EmbarkedQ              -2.610e+00  1.215e+01  -0.215  0.82992
## EmbarkedS              -7.014e-01  1.239e+00  -0.566  0.57136
## Pclassmiddle:Sexmale   -1.207e+00  1.297e+00  -0.931  0.35187
## Pclasslower:Sexmale     1.265e+00  1.288e+00   0.982  0.32619
## Pclassmiddle:Age       -1.501e-02  3.516e-02  -0.427  0.66938
## Pclasslower:Age         1.706e-02  3.211e-02   0.531  0.59516
## Pclassmiddle:SibSp     -1.016e+00  8.699e-01  -1.168  0.24290
## Pclasslower:SibSp      -1.624e+00  7.989e-01  -2.033  0.04206 *
## Pclassmiddle:Parch      4.799e-01  8.034e-01   0.597  0.55027
## Pclasslower:Parch      -4.980e-01  7.112e-01  -0.700  0.48382
## Pclassmiddle:Fare      -6.430e-03  2.911e-02  -0.221  0.82520
## Pclasslower:Fare        4.703e-02  1.710e-02   2.750  0.00595 **
## Pclassmiddle:EmbarkedQ  6.190e+00  1.190e+01   0.520  0.60296
## Pclasslower:EmbarkedQ   3.890e+00  1.055e+01   0.369  0.71238
## Pclassmiddle:EmbarkedS  1.885e-01  1.113e+00   0.169  0.86558
## Pclasslower:EmbarkedS  -4.780e-01  7.776e-01  -0.615  0.53877
## Sexmale:Age            -4.794e-02  2.385e-02  -2.010  0.04440 *
## Sexmale:SibSp          -8.652e-02  3.335e-01  -0.259  0.79532
## Sexmale:Parch           4.651e-01  4.069e-01   1.143  0.25308
## Sexmale:Fare           -1.158e-02  1.323e-02  -0.875  0.38155
## Sexmale:EmbarkedQ      -2.407e+00  1.146e+00  -2.100  0.03571 *
## Sexmale:EmbarkedS      -2.794e-01  6.294e-01  -0.444  0.65711
## Age:SibSp              -2.033e-03  1.546e-02  -0.131  0.89541
## Age:Parch              -4.736e-02  1.474e-02  -3.214  0.00131 **
## Age:Fare                1.035e-03  3.949e-04   2.622  0.00874 **
## Age:EmbarkedQ          -1.927e-03  1.001e-01  -0.019  0.98463
## Age:EmbarkedS           3.632e-02  2.552e-02   1.423  0.15472
## SibSp:Parch            -7.853e-03  2.334e-01  -0.034  0.97315
## SibSp:Fare             -3.465e-03  5.680e-03  -0.610  0.54183
## SibSp:EmbarkedQ         1.839e+00  1.484e+00   1.239  0.21520
## SibSp:EmbarkedS        -5.445e-01  4.870e-01  -1.118  0.26354
## Parch:Fare             -4.888e-03  5.044e-03  -0.969  0.33244
## Parch:EmbarkedQ        -2.122e+01  5.455e+02  -0.039  0.96897
## Parch:EmbarkedS        -2.835e-02  4.183e-01  -0.068  0.94597
## Fare:EmbarkedQ         -2.312e-02  1.440e-01  -0.161  0.87247
## Fare:EmbarkedS         -6.180e-03  8.511e-03  -0.726  0.46777
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.82  on 888  degrees of freedom
## Residual deviance:  672.02  on 845  degrees of freedom
## AIC: 760.02
##
## Number of Fisher Scoring iterations: 16
```

As can be seen by both the residual deviance and AIC score of the two models, adding interaction terms

greatly increased the accuracy of the model, though the number of terms in the new model is still much greater than before. Using the step function, I attempt to find the best model based on AIC criterion by both adding and subtracting variables.

*Trace of steps is suppressed for convenience

```
step(ti.inter.fit, direction="both", trace = FALSE)
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked + Pclass:Sex + Pclass:SibSp + Pclass:Parch +
##     Pclass:Fare + Sex:Age + Sex:Embarked + Age:Parch + Age:Fare +
##     SibSp:Embarked + Parch:Embarked + Fare:Embarked, family = binomial,
##     data = ti.train)
##
## Coefficients:
##          (Intercept)          Pclassmiddle           Pclasslower
##            3.994e+00            -1.100e+00            -3.935e+00
##              Sexmale                   Age                 SibSp
##           -2.263e+00            -8.741e-03             1.040e+00
##                Parch                  Fare             EmbarkedQ
##            1.489e+00            -2.887e-02             1.719e+00
##            EmbarkedS  Pclassmiddle:Sexmale   Pclasslower:Sexmale
##           -2.211e-01            -5.819e-01             1.877e+00
##   Pclassmiddle:SibSp     Pclasslower:SibSp    Pclassmiddle:Parch
##           -3.831e-01            -1.110e+00             1.045e+00
##    Pclasslower:Parch     Pclassmiddle:Fare      Pclasslower:Fare
##           -6.175e-02            -1.341e-02             4.350e-02
##          Sexmale:Age     Sexmale:EmbarkedQ     Sexmale:EmbarkedS
##           -5.522e-02            -2.215e+00             6.390e-02
##            Age:Parch              Age:Fare       SibSp:EmbarkedQ
##           -4.856e-02             9.009e-04             1.980e+00
##      SibSp:EmbarkedS       Parch:EmbarkedQ       Parch:EmbarkedS
##           -7.923e-01            -2.096e+01            -1.584e-01
##       Fare:EmbarkedQ        Fare:EmbarkedS
##           -7.585e-02            -1.009e-03
##
## Degrees of Freedom: 888 Total (i.e. Null);  860 Residual
## Null Deviance:       1183
## Residual Deviance: 680.7     AIC: 738.7
```

The final model that is output by the step function seems to be nearly the same as the previous logistic model including interactions. The terms that have been taken out are: Pclass:Age, Pclass:Embarked, Sex:SibSp, Sex:Parch, Sex:Fare, Age:SibSp, Age:Embarked, SibSp:Parch, SibSp:Fare, Parch:Fare.

I've created a new variable to hold this new fit model below. Though this model's residual deviance is greater than the full model's, its AIC is much lower.

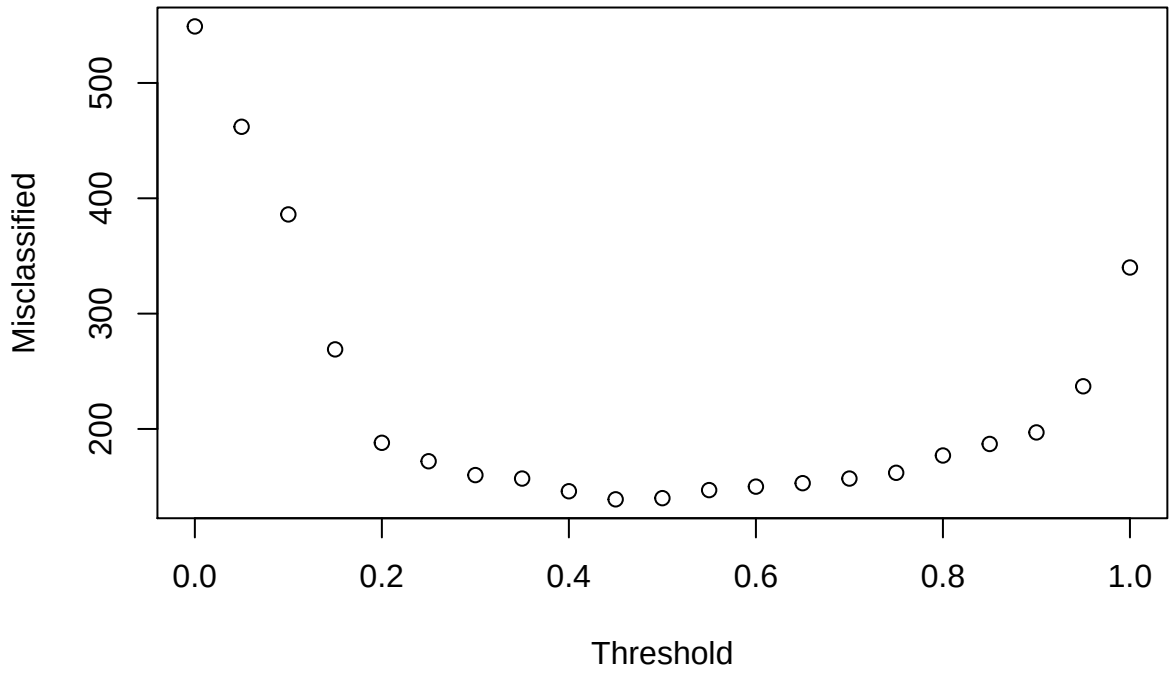
```
step.ti.fit <- glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
    Fare + Embarked + Pclass:Sex + Pclass:SibSp + Pclass:Parch +
    Pclass:Fare + Sex:Age + Sex:Embarked + Age:Parch + Age:Fare +
    SibSp:Embarked + Parch:Embarked + Fare:Embarked, family = binomial,
    data = ti.train)
summary(step.ti.fit) #680, 738
```

```
##
## Call:
```

```
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked + Pclass:Sex + Pclass:SibSp + Pclass:Parch +
##     Pclass:Fare + Sex:Age + Sex:Embarked + Age:Parch + Age:Fare +
##     SibSp:Embarked + Parch:Embarked + Fare:Embarked, family = binomial,
##     data = ti.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8421  -0.5928  -0.3410   0.3858   2.8149
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.994e+00  9.933e-01   4.020 5.81e-05 ***
## Pclassmiddle         -1.100e+00  1.003e+00  -1.097 0.272645
## Pclasslower          -3.935e+00  8.178e-01  -4.812 1.50e-06 ***
## Sexmale              -2.263e+00  9.855e-01  -2.297 0.021645 *
## Age                  -8.741e-03  1.889e-02  -0.463 0.643504
## SibSp                 1.040e+00  4.999e-01   2.080 0.037529 *
## Parch                 1.489e+00  5.729e-01   2.599 0.009347 **
## Fare                 -2.887e-02  8.407e-03  -3.434 0.000594 ***
## EmbarkedQ             1.719e+00  7.657e-01   2.245 0.024767 *
## EmbarkedS            -2.211e-01  5.743e-01  -0.385 0.700210
## Pclassmiddle:Sexmale -5.819e-01  8.934e-01  -0.651 0.514832
## Pclasslower:Sexmale   1.877e+00  7.639e-01   2.457 0.014010 *
## Pclassmiddle:SibSp   -3.831e-01  6.507e-01  -0.589 0.556074
## Pclasslower:SibSp    -1.110e+00  4.416e-01  -2.514 0.011939 *
## Pclassmiddle:Parch    1.045e+00  6.184e-01   1.689 0.091164 .
## Pclasslower:Parch    -6.175e-02  4.405e-01  -0.140 0.888517
## Pclassmiddle:Fare    -1.341e-02  2.883e-02  -0.465 0.641827
## Pclasslower:Fare      4.350e-02  1.596e-02   2.725 0.006424 **
## Sexmale:Age          -5.522e-02  2.065e-02  -2.674 0.007499 **
## Sexmale:EmbarkedQ    -2.215e+00  1.066e+00  -2.079 0.037626 *
## Sexmale:EmbarkedS     6.390e-02  5.907e-01   0.108 0.913857
## Age:Parch            -4.856e-02  1.223e-02  -3.970 7.18e-05 ***
## Age:Fare              9.009e-04  2.372e-04   3.798 0.000146 ***
## SibSp:EmbarkedQ       1.980e+00  1.103e+00   1.795 0.072678 .
## SibSp:EmbarkedS      -7.923e-01  4.506e-01  -1.759 0.078650 .
## Parch:EmbarkedQ      -2.096e+01  5.279e+02  -0.040 0.968327
## Parch:EmbarkedS      -1.584e-01  4.036e-01  -0.392 0.694820
## Fare:EmbarkedQ       -7.585e-02  3.214e-02  -2.360 0.018294 *
## Fare:EmbarkedS       -1.009e-03  5.868e-03  -0.172 0.863421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.82  on 888  degrees of freedom
## Residual deviance:  680.68  on 860  degrees of freedom
## AIC: 738.68
##
## Number of Fisher Scoring iterations: 16
```

Threshold determination to minimize misclassification:

```r
conf <- matrix(0, nrow = 21, ncol = 5)
colnames(conf) <- c("thr", "a", "b", "c", "d")
conf[, 1] <- seq(0, 1, by = 0.05)
y <- as.numeric(ti.train$Survived) - 1
y.pred <- step.ti.fit$fitted.values
for (i in 1:21) {
  a <- sum((!y) & (y.pred <= conf[i, 1]))
  b <- sum((!y) & (y.pred > conf[i, 1]))
  c <- sum((y) & (y.pred <= conf[i, 1]))
  d <- sum((y) & (y.pred > conf[i, 1]))
  conf[i, 2:5] <- c(a, b, c, d)
}
plot(conf[, 1], conf[, 3] + conf[, 4], xlab = "Threshold", ylab = "Misclassified") #0.5
```



With the model, I have created a confusion matrix for the prediction and actual results, and plotted the errors (predict no, actual yes and predict yes, actual no) over .05 increments in probability. Sicne the plot reaches a minimum at approximately 0.5, I will use that value as the threshold to determine the category assigned to a specific entry based on its predicted probability.

This function below simply helps convert the probability by determining if it is above or below threshold.

```r
conv.Surv <- function(x) {
  if(x >= 0.50) return(1)
  else return(0)
}
```

Using the predict() function with “response” as its type parameter, I return a vector of probabilities based on

the logistic regression equation. I then determine the predicted category for each entry using the threshold I obtained above, and then write its output into a csv file.

```
pred <- predict(step.ti.fit, ti.test, type="response")
pred <- sapply(pred, conv.Surv)
subm = data.frame(PassengerId = ti.test$PassengerId, Survived = pred)
write.csv(subm, file = "Subm.csv", row.names = FALSE)
```

Kaggle Score: 0.76077

Through some simple algebra, it can be seen that the Kaggle Score is calculated by the number of correct predictions of the Survive category. My accuracy score is not that high compared to others on the scoreboard.

Classification Tree:

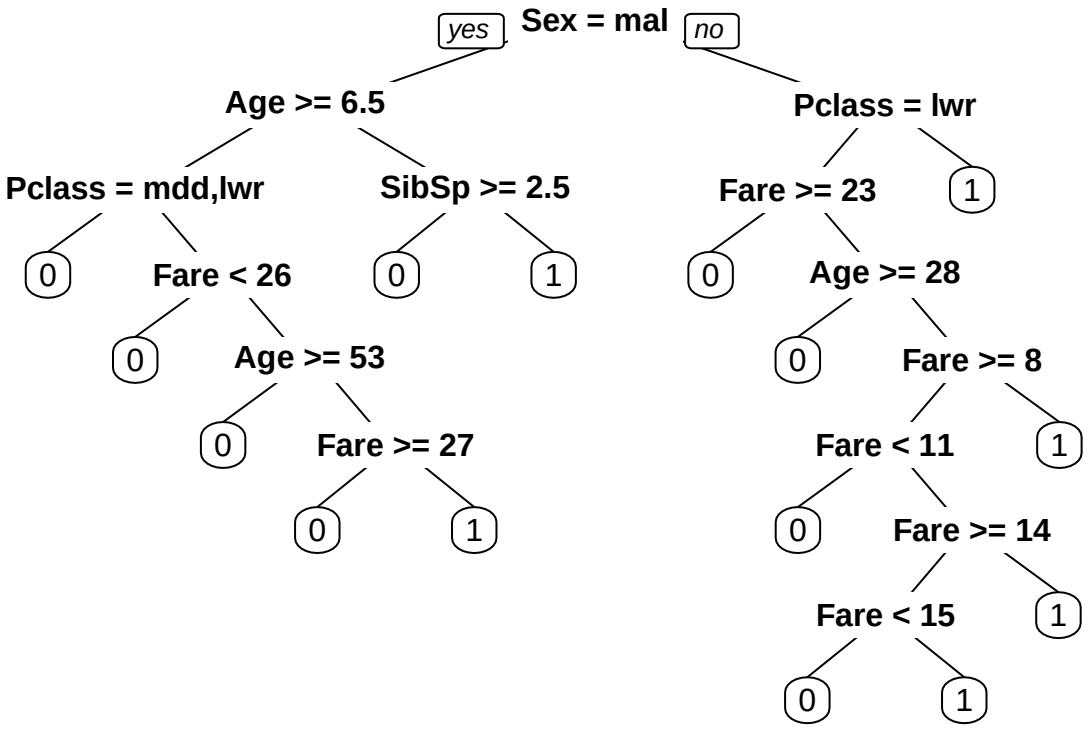
```
rt = rpart(Survived~., data = ti.train, method="class", cp=0.0001)
printcp(rt)
```

```
##
## Classification tree:
## rpart(formula = Survived ~ ., data = ti.train, method = "class",
##     cp = 1e-04)
##
## Variables actually used in tree construction:
## [1] Age    Fare   Pclass Sex    SibSp
##
## Root node error: 340/889 = 0.38245
##
## n= 889
##
##           CP nsplit rel error  xerror     xstd
## 1  0.4411765      0   1.00000 1.00000 0.042618
## 2  0.0308824      1   0.55882 0.55882 0.035949
## 3  0.0235294      3   0.49706 0.52059 0.035018
## 4  0.0205882      4   0.47353 0.51176 0.034793
## 5  0.0117647      6   0.43235 0.50294 0.034565
## 6  0.0088235      8   0.40882 0.49118 0.034253
## 7  0.0066176     10   0.39118 0.48235 0.034014
## 8  0.0029412     14   0.36471 0.49118 0.034253
## 9  0.0014706     17   0.35588 0.50294 0.034565
## 10 0.0001000     19   0.35294 0.50294 0.034565
```

Next, I use rpart() to fit the data into a classification tree. I initially set the cp value to be 0.0001, and then output the cp values along with their errors. Since the xerror changes each time the code is run, I loop through finding the min xerror 1000 times and then take the mode of the resulting vector of cp values. Below is the new, pruned tree that uses this optimal cp value.

```
lowcp <- c()
for(i in 1:1000) {
  rtree <- rpart(Survived~., data = ti.train, method="class", cp=0.0001)
  lowcp <- c(lowcp, rtree$cptable[which.min(rtree$cptable[,"xerror"]),"CP"])
}
```

```
rt.prune <- prune(rt, cp = Mode(lowcp))
prp(rt.prune)
```



This tree that is grown uses the variables Age, Fare, Pclass, Sex, and SibSp. The cp used is 0.0029412, resulting in 14 splits. Using this tree, I once again try to predict the values in the test set.

```
tree.pred <- predict(rt.prune, ti.test, type="class")
tree.pred <- as.numeric(tree.pred) - 1
subm = data.frame(PassengerId = ti.test$PassengerId, Survived = tree.pred)
write.csv(subm, file = "Subm.tree.csv", row.names = FALSE)
```

Kaggle Tree Score: 0.76077

There is no change in my Kaggle score compared to before.

Prediction via Random Forests:

```
rf <- randomForest(Survived~., data=ti.train, importance = TRUE)
rf
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = ti.train, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.65%
## Confusion matrix:
##     0   1 class.error
## 0 504  45  0.08196721
## 1 103 237  0.30294118
```

Now, random forests are used instead for classification. This method uses bootstrapping techniques to sample trees multiple times from the training set with random variables used each time at the splits. The object created by this function outputs a confusion matrix that gives a rough estimate of how accurate the averaged predictions are.

Once again, predictions on the test data set are made based on this classification method.

```
forest.pred <- predict(rf, ti.test)
forest.pred <- as.numeric(forest.pred) - 1
subm = data.frame(PassengerId = ti.test$PassengerId, Survived = forest.pred)
write.csv(subm, file = "Subm.forest.csv", row.names = FALSE)
```

Kaggle Score: 0.77990

This time, it appears that the score has increased slightly. Perhaps this large number of random samples drawn, then averaged, allow for a less biased estimation of the true classification method.

Comparison:

Overall, my scores were quite far from the top scoring entries, many of which seemed to even have perfect prediction. Through the three methods attempted, there was very little difference in the accuracy of my model on the test set.

Potential issues or improvements could be in how I handle the data originally. As there were many N/A values in the age category, I chose to substitute them with the median to include them all into my training data set for modeling. This could very well alter the accuracy of my data, as other methods of handling the N/A could be to use the mean, or even drop those values all together. Furthermore, since I use the same values to substitute in the test set, it could be an issue where the test set, by chance, actually has a different age distribution that will also decrease the accuracy of my modeling. Investigations into correlated explanatory variables and overlapping effects may also help.