151AHW5

Jiyoon Clover Jeong

11/26/2017

```
train = read.csv("/Users/cloverjiyoon/2017Fall/Stat 151A/HW/HW5/train.csv")
test = read.csv("/Users/cloverjiyoon/2017Fall/Stat 151A/HW/HW5/test.csv")

testID <- test$PassengerId
train$PassengerId <- NULL
test$PassengerId <- NULL

train$data_type = "train"
test$data_type = "test"
test$Survived = ""
whole = rbind(train, test)

whole$Name <- NULL
whole$Cabin <- NULL
whole$Ticket <- NULL


train$data_type = as.factor(train$data_type)

whole$Pclass = as.factor(whole$Pclass)
whole$Survived = as.numeric(whole$Survived)

summary(whole)
```

```
##     Survived      Pclass      Sex           Age             SibSp
##  Min.   :0.0000   1:323   female:466   Min.   : 0.17   Min.   :0.0000
##  1st Qu.:0.0000   2:277   male  :843   1st Qu.:21.00   1st Qu.:0.0000
##  Median :0.0000   3:709                Median :28.00   Median :0.0000
##  Mean   :0.3838                        Mean   :29.88   Mean   :0.4989
##  3rd Qu.:1.0000                        3rd Qu.:39.00   3rd Qu.:1.0000
##  Max.   :1.0000                        Max.   :80.00   Max.   :8.0000
##  NA's   :418                           NA's   :263
##      Parch            Fare          Embarked  data_type
##  Min.   :0.000   Min.   :  0.000    :  2    Length:1309
##  1st Qu.:0.000   1st Qu.:  7.896   C:270    Class :character
##  Median :0.000   Median : 14.454   Q:123    Mode  :character
##  Mean   :0.385   Mean   : 33.295   S:914
##  3rd Qu.:0.000   3rd Qu.: 31.275
##  Max.   :9.000   Max.   :512.329
##                  NA's   :1
```

```
set.seed(100)

for(i in 2: dim(whole)[2]){
  cat("Number of missing/NA values of ", names(whole)[i], "  :  ",
      nrow(whole[whole[,i] == "", ]) , "\n" )
}
```

```
## Number of missing/NA values of  Pclass   :   0
## Number of missing/NA values of  Sex   :   0
## Number of missing/NA values of  Age   :   263
## Number of missing/NA values of  SibSp   :   0
## Number of missing/NA values of  Parch   :   0
## Number of missing/NA values of  Fare   :   1
## Number of missing/NA values of  Embarked   :   2
## Number of missing/NA values of  data_type   :   0
```

```
# Drop variable "Survived", "Name", "Ticket", "Cabin", "data_type"
lmage <- lm(Age ~ ., data = whole[, -c(1, 9)])
summary(lmage)
```

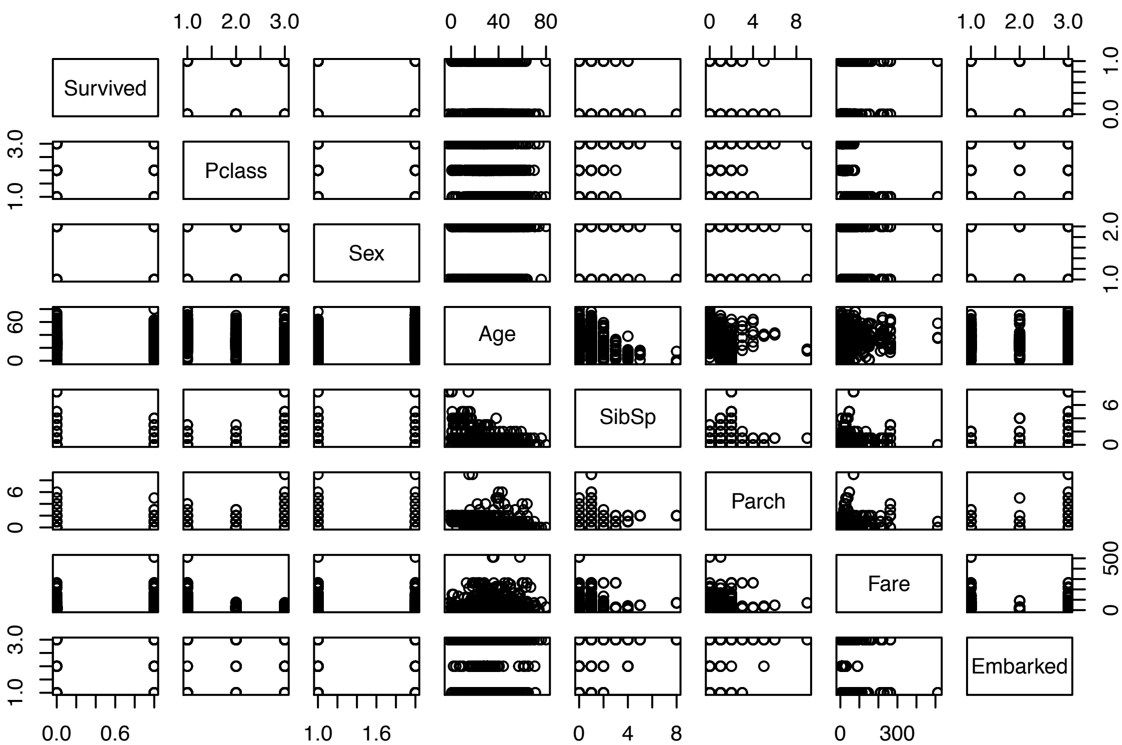
```
##
## Call:
## lm(formula = Age ~ ., data = whole[, -c(1, 9)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.830  -7.810  -1.496   7.341  46.237
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  50.437930   8.911786   5.660 1.96e-08 ***
## Pclass2     -11.217664   1.290719  -8.691  < 2e-16 ***
## Pclass3     -15.880277   1.227040 -12.942  < 2e-16 ***
## Sexmale       2.730088   0.843007   3.239  0.00124 **
## SibSp        -3.180196   0.463970  -6.854 1.23e-11 ***
## Parch        -0.703905   0.522641  -1.347  0.17833
## Fare         -0.005474   0.009366  -0.584  0.55902
## EmbarkedC   -11.778842   8.946107  -1.317  0.18825
## EmbarkedQ    -6.856667   9.111830  -0.753  0.45192
## EmbarkedS    -9.482555   8.944117  -1.060  0.28930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.56 on 1035 degrees of freedom
##   (264 observations deleted due to missingness)
## Multiple R-squared:  0.2448, Adjusted R-squared:  0.2383
## F-statistic: 37.28 on 9 and 1035 DF,  p-value: < 2.2e-16
```

```
# Age NA
predictedage <- predict(lmage, whole[is.na(whole$Age), -c(1, 9)])
whole$Age[is.na(whole$Age)] <- predictedage


# Fare NA - median since
whole$Fare[is.na(whole$Fare)] <- median(whole$Fare, na.rm=TRUE)


# Embarked NA
whole$Embarked[whole$Embarked == ""] <- sample(c("C", "Q", "S"), size = 2,
                                              replace = T)
whole$Embarked = as.factor(as.character(whole$Embarked))
```

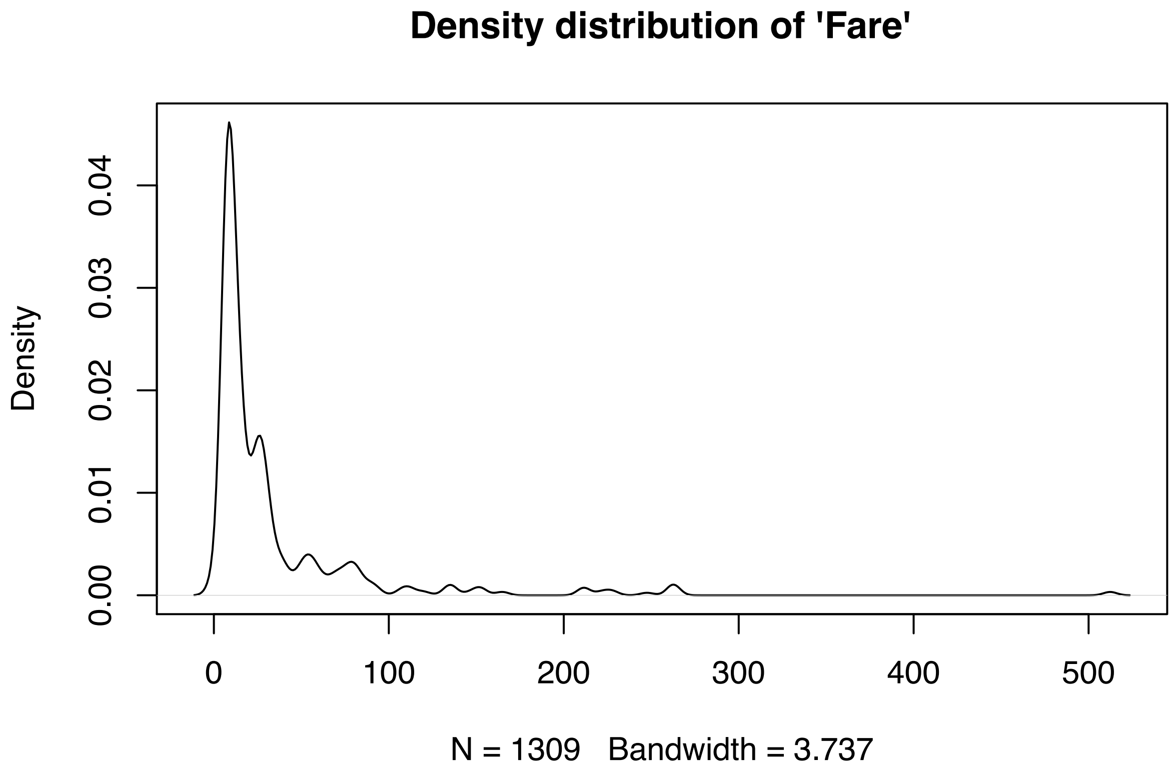
```r
# Check

for(i in 2: dim(whole)[2]){
  cat("Number of missing/NA values of ", names(whole)[i], "  :  ",
      nrow(whole[whole[,i] == "", ]) , "\n" )
}
```

```
## Number of missing/NA values of  Pclass   :   0
## Number of missing/NA values of  Sex   :   0
## Number of missing/NA values of  Age   :   0
## Number of missing/NA values of  SibSp   :   0
## Number of missing/NA values of  Parch   :   0
## Number of missing/NA values of  Fare   :   0
## Number of missing/NA values of  Embarked   :   0
## Number of missing/NA values of  data_type   :   0
```

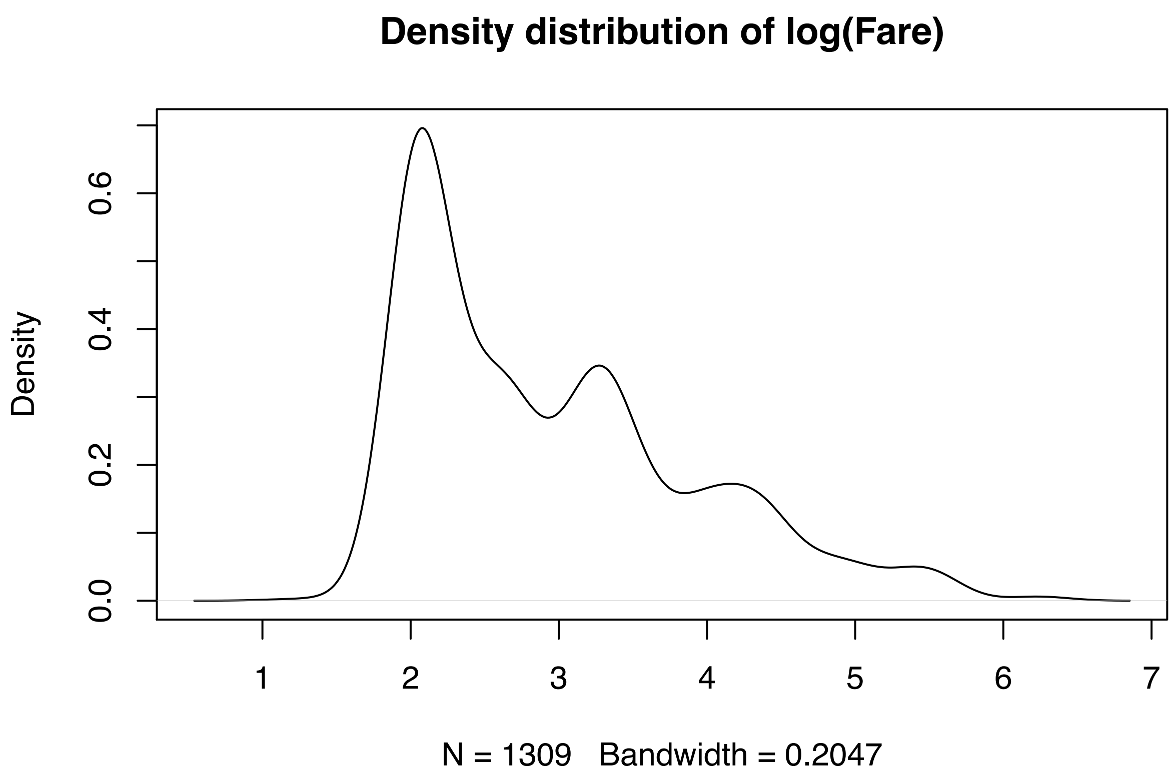
```r
pairs(whole[,-9])
```




```r
plot(density(whole$Fare), main = "Density distribution of 'Fare'")
```




```
plot(density(log(whole$Fare)), main = "Density distribution of log(Fare)")
```




```
# before we transform log(Fare), change 0 value to median

whole$Fare[whole$Fare == 0] <- median(whole$Fare, na.rm=TRUE)
```

```r
train = whole[whole$data_type == 'train',]
test = whole[whole$data_type == 'test',]
train$data_type = NULL
test$data_type = NULL
```

```r
# FUll model

fit1 = glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch + log(Fare) + Embarked , family = bi
summary(fit1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     log(Fare) + Embarked, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7042  -0.6079  -0.4124   0.6139   2.4907
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.189023   0.934183   3.414 0.000641 ***
## Pclass2     -0.824833   0.353736  -2.332 0.019712 *
## Pclass3     -1.946750   0.428556  -4.543 5.56e-06 ***
## Sexmale     -2.684398   0.202467 -13.258  < 2e-16 ***
## Age         -0.043170   0.008313  -5.193 2.07e-07 ***
## SibSp       -0.460657   0.120215  -3.832 0.000127 ***
## Parch       -0.157423   0.129450  -1.216 0.223952
## log(Fare)    0.331804   0.204369   1.624 0.104471
## EmbarkedQ    0.049022   0.386576   0.127 0.899091
## EmbarkedS   -0.363009   0.242250  -1.498 0.134006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  779.26  on 881  degrees of freedom
## AIC: 799.26
##
## Number of Fisher Scoring iterations: 5
```

```r
# Obtain newtrain for bestglm object
newtrain <- train[, c(2:8,1)]
newtrain$Fare <- log(newtrain$Fare)


# AIC criteria selection
AIC.fit1 = bestglm(newtrain, family = binomial, IC = "AIC", method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
AIC.fit1$BestModels
```

```
##   Pclass  Sex  Age SibSp Parch  Fare Embarked Criterion
## 1   TRUE TRUE TRUE  TRUE FALSE FALSE     TRUE  796.2444
## 2   TRUE TRUE TRUE  TRUE FALSE FALSE    FALSE  796.5294
## 3   TRUE TRUE TRUE  TRUE FALSE  TRUE    FALSE  796.5718
## 4   TRUE TRUE TRUE  TRUE  TRUE  TRUE    FALSE  796.5741
## 5   TRUE TRUE TRUE  TRUE FALSE  TRUE     TRUE  796.7673
```

```r
# BIC criteria selection
BIC.fit1 = bestglm(newtrain, family = binomial, IC = "BIC", method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
BIC.fit1$BestModels
```

```
##   Pclass  Sex  Age SibSp Parch  Fare Embarked Criterion
## 1   TRUE TRUE TRUE  TRUE FALSE FALSE    FALSE  820.4911
## 2   TRUE TRUE TRUE  TRUE FALSE  TRUE    FALSE  825.3258
## 3   TRUE TRUE TRUE  TRUE  TRUE FALSE    FALSE  826.8248
## 4   TRUE TRUE TRUE  TRUE FALSE FALSE     TRUE  829.7908
## 5   TRUE TRUE TRUE  TRUE  TRUE  TRUE    FALSE  830.1206
```

```r
fit2 <- glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch + log(Fare) + Embarked + Sex:Age, f
summary(fit2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     log(Fare) + Embarked + Sex:Age, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5520  -0.5938  -0.4066   0.6113   2.6578
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.836914   0.953862   2.974 0.002938 **
## Pclass2     -1.166533   0.381974  -3.054 0.002258 **
## Pclass3     -2.239953   0.448517  -4.994 5.91e-07 ***
## Sexmale     -1.060355   0.445182  -2.382 0.017226 *
## Age         -0.008559   0.012223  -0.700 0.483767
## SibSp       -0.387202   0.120456  -3.214 0.001307 **
## Parch       -0.182926   0.129628  -1.411 0.158196
## log(Fare)    0.240685   0.210226   1.145 0.252256
## EmbarkedQ   -0.021070   0.385094  -0.055 0.956367
## EmbarkedS   -0.428968   0.244995  -1.751 0.079959 .
## Sexmale:Age -0.059979   0.015428  -3.888 0.000101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  763.23  on 880  degrees of freedom
```

```
## AIC: 785.23
##
## Number of Fisher Scoring iterations: 5
```

```r
AgeSex <- ifelse(whole$Sex == "male", whole$Age, 0)

whole$AgeSex <- AgeSex
train = whole[whole$data_type == 'train',]
test = whole[whole$data_type == 'test',]

train$data_type = NULL
test$data_type = NULL

# Xy for bestglm function
newtrain = train[, c(2:9,1)]
newtrain$Fare <- log(newtrain$Fare)

AIC.fit2 = bestglm(newtrain, family = binomial, IC = "AIC")
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
BIC.fit2 = bestglm(newtrain, family = binomial, IC = "BIC")
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
AIC.fit2$BestModels
```

```
##   Pclass  Sex   Age SibSp Parch  Fare Embarked AgeSex Criterion
## 1   TRUE TRUE FALSE  TRUE FALSE FALSE     TRUE   TRUE  780.1064
## 2   TRUE TRUE FALSE  TRUE  TRUE FALSE     TRUE   TRUE  780.9543
## 3   TRUE TRUE FALSE  TRUE FALSE FALSE    FALSE   TRUE  781.3080
## 4   TRUE TRUE  TRUE  TRUE FALSE FALSE     TRUE   TRUE  781.6001
## 5   TRUE TRUE FALSE  TRUE  TRUE  TRUE     TRUE   TRUE  781.7128
```

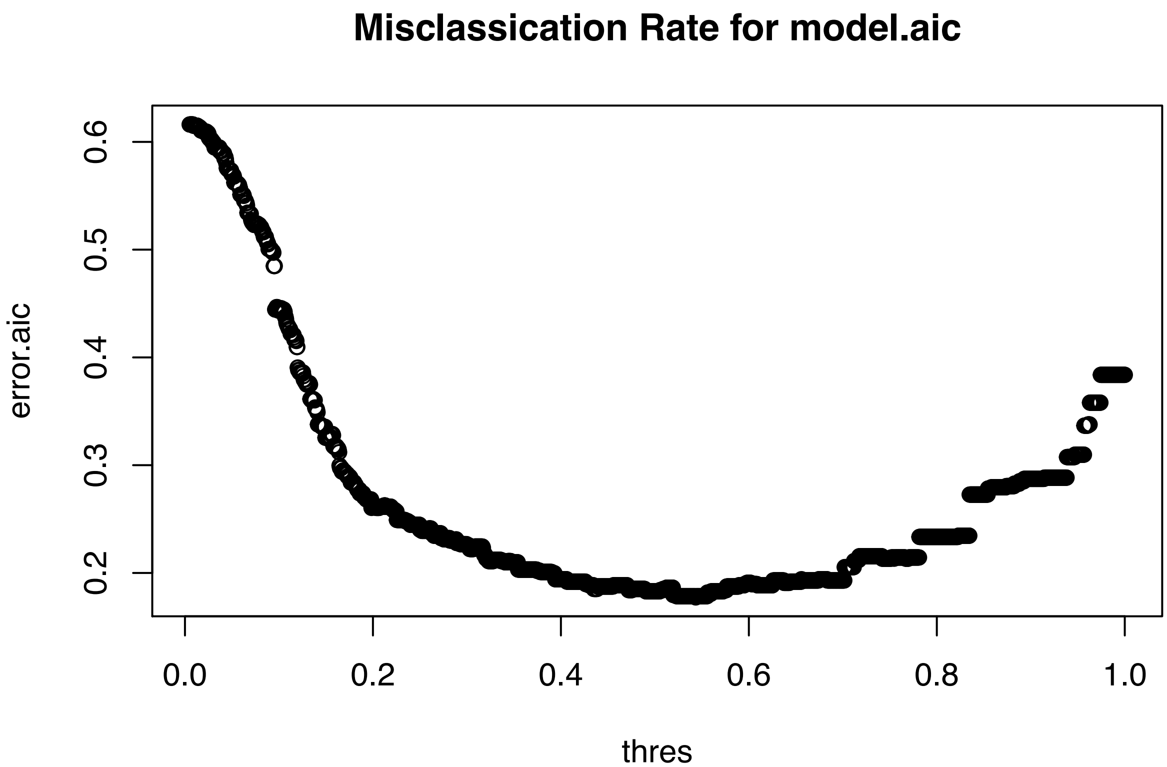
```r
BIC.fit2$BestModels
```

```
##   Pclass   Sex   Age SibSp Parch  Fare Embarked AgeSex Criterion
## 1   TRUE FALSE FALSE  TRUE FALSE FALSE    FALSE   TRUE  804.8516
## 2   TRUE  TRUE FALSE  TRUE FALSE FALSE    FALSE   TRUE  805.2698
## 3   TRUE FALSE FALSE  TRUE  TRUE FALSE    FALSE   TRUE  810.4898
## 4   TRUE  TRUE FALSE  TRUE  TRUE FALSE    FALSE   TRUE  810.6999
## 5   TRUE FALSE  TRUE  TRUE FALSE FALSE    FALSE   TRUE  810.8335
```

AIC

```r
# best model from aic
model.aic = glm(Survived ~ Pclass + Sex + SibSp + Embarked + AgeSex,
family = binomial, data = train)


# Find the minimum of misclassificaton rates of model.aic

thres <- seq(from=0.005, to=1, by = 0.0005)
predicted.aic <- predict(model.aic, newdata = train, type = "response")
```

```r
predclass.aic <- list()
error.aic <- c()

for (i in 1:length(thres)) {
  predclass.aic[[i]] <- ifelse(predicted.aic < i*0.0005, 0, 1)

  # Misclassification rate
  error.aic[i] <- mean(predclass.aic[[i]] != train$Survived)
}


plot(y = error.aic, x = thres, main = "Misclassication Rate for model.aic")
```




```r
# Minimum of the Misclassication rate
min1 <- min(error.aic)
min1
```

```
## [1] 0.1773288
```

```r
optim1 <- which(error.aic == min(error.aic))
cat("Optimal threshold for model.aic :", optim1 * 0.0005)
```

```
## Optimal threshold for model.aic : 0.539
```

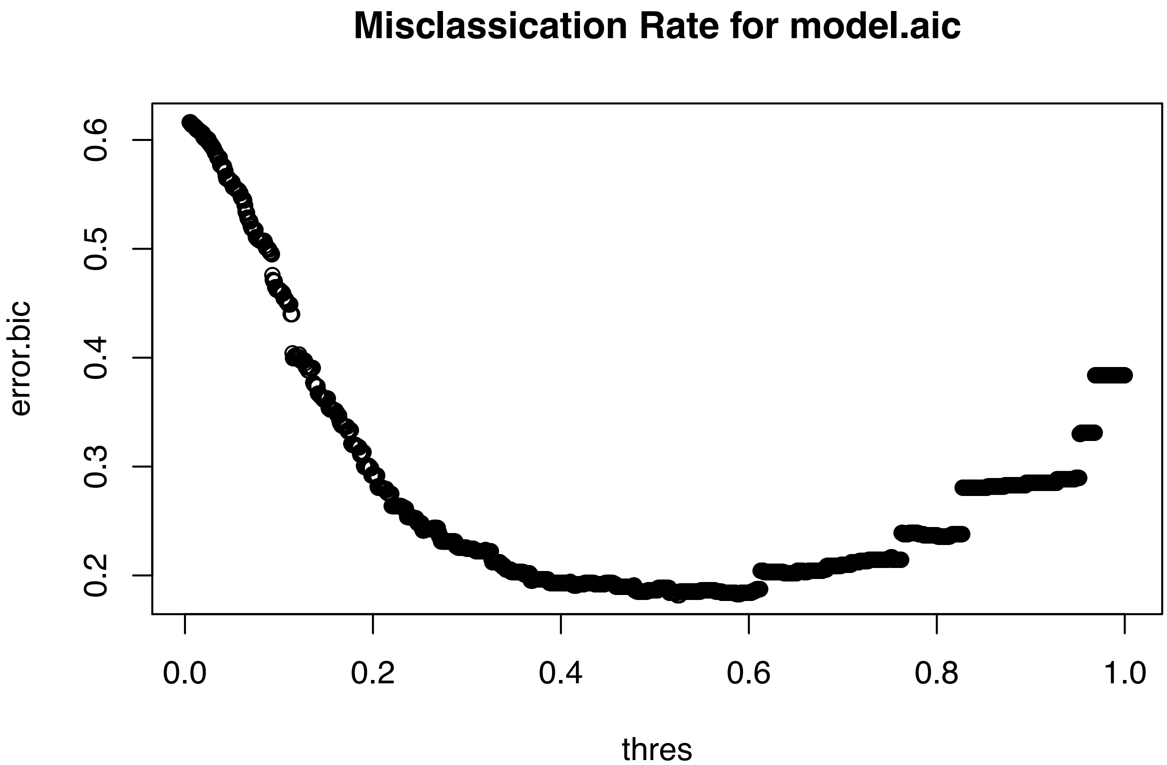
```r
summary(model.aic)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + SibSp + Embarked + AgeSex,
##     family = binomial, data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6475  -0.5951  -0.4068   0.6101   2.6428
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.47404    0.32713  10.620  < 2e-16 ***
## Pclass2     -1.40084    0.30413  -4.606 4.10e-06 ***
## Pclass3     -2.59078    0.27942  -9.272  < 2e-16 ***
## Sexmale     -0.82483    0.33881  -2.435 0.014912 *
## SibSp       -0.33900    0.09788  -3.463 0.000533 ***
## EmbarkedQ   -0.04869    0.37635  -0.129 0.897063
## EmbarkedS   -0.48615    0.23853  -2.038 0.041545 *
## AgeSex      -0.06741    0.01082  -6.233 4.58e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  766.11  on 883  degrees of freedom
## AIC: 782.11
##
## Number of Fisher Scoring iterations: 5
```

BIC

```r
# best model from bic
model.bic =  glm(Survived ~ Pclass + SibSp + AgeSex, family = binomial,
                 data = train)

# Find the minimum of misclassificaton rates of model.bic
predicted.bic <- predict(model.bic, newdata = train, type = "response")
predclass.bic <- list()
error.bic <- c()

for (i in 1:length(thres)) {
  predclass.bic[[i]] <- ifelse(predicted.bic < i*0.0005, 0, 1)

  # Misclassification rate
  error.bic[i] <- mean(predclass.bic[[i]] != train$Survived)
}

plot(y = error.bic, x = thres, main = "Misclassication Rate for model.aic")
```




```
# Minimum of the Misclassication rate
min2 <- min(error.bic)
min2
```

```
## [1] 0.1818182
```

```
optim2 <- which(error.bic == min(error.bic))
cat("Optimal threshold for model.bic :", optim2 * 0.0005)
```

```
## Optimal threshold for model.bic : 0.5195 0.52 0.5205 0.521 0.5215
```

```
summary(model.bic)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + SibSp + AgeSex, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5750  -0.6412  -0.3952   0.6245   2.8752
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.278261   0.297281  11.027  < 2e-16 ***
## Pclass2     -1.742547   0.293978  -5.927 3.08e-09 ***
## Pclass3     -2.840984   0.265005 -10.720  < 2e-16 ***
## SibSp       -0.394794   0.094766  -4.166 3.10e-05 ***
## AgeSex      -0.091175   0.006484 -14.061  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  777.68  on 886  degrees of freedom
## AIC: 787.68
##
## Number of Fisher Scoring iterations: 5
```

compare misclassfication rate between aic and bic

```
cat("AIC minimum misclassification rate :  ", min1, "\n")
```

```
## AIC minimum misclassification rate :   0.1773288
```

```
cat("BIC minimum misclassification rate :  ", min2, "\n")
```

```
## BIC minimum misclassification rate :   0.1818182
```

```
cat(min1, " is smaller than ", min2, ".\n Therefore, choose the model from AIC and predict the test set
```

```
## 0.1773288  is smaller than  0.1818182 .
##  Therefore, choose the model from AIC and predict the test set
```

```
predicted.aic <- predict(model.aic, newdata = test, type = "response")

predclass.aic <- ifelse(predicted.aic < optim1 * 0.0005, 0, 1)

final <- data.frame(PassengerId = testID)
final$Survived <- predclass.aic

write.csv(final, file = "final.csv", row.names=FALSE)
```

score and rank from Kaggle

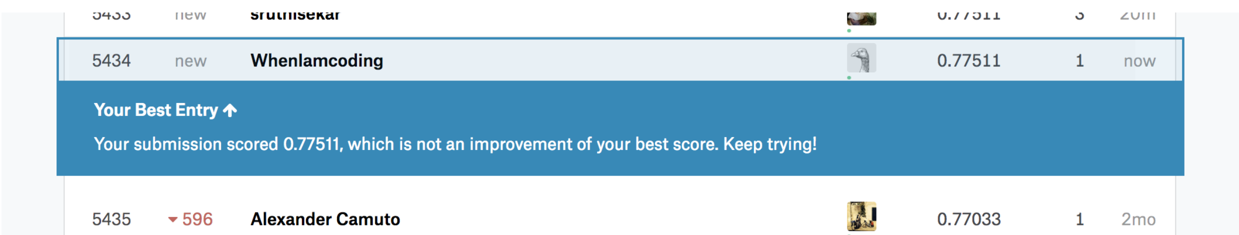


Figure 1: Kaggle score

2. Since we proved $\nabla \ell(\beta) = X^T(Y-P)$ and $\hat{\beta}$ maximize the loglikelihood, $\Rightarrow \nabla \ell(\hat{\beta}) = 0 = X^T(Y-\hat{P})$ where $\hat{P}_i = \frac{e^{x_i^T\hat{\beta}}}{1+e^{x_i^T\hat{\beta}}}$

we say a and b are orthogonal when $\langle a, b\rangle = 0 = a^T b = 0$

Therefore, $X^T(Y-\hat{P}) = \langle X, Y-\hat{P}\rangle = 0$ $\Rightarrow$ columns of X and $Y-\hat{P}$ is orthogonal.

3. (a) ① $\frac{\hat{\beta}_0}{S.E(\hat{\beta}_0)} = 0.313$ $\Rightarrow$ $S.E(\hat{\beta}_0) = \frac{0.6864}{0.313} = 2.19297$

or as given in $(X^TWX)^{-1}_{1,1} = 4.8038479$, $S.E(\hat{\beta}_0) = \sqrt{4.8038479}$

$= 2.19177$

② $\sqrt{0.04331330 7} \approx 0.20812$

⑤ AIC $= -2\log(\text{maximum value of likelihood in } m) + 2(1+p)$

$=$ residual deviance $+ 2(1+3) = 222.18$

$\Rightarrow$ residual deviance $= 222.18 - 8 = 214.18$

⑥ Df of residual deviance : $n-p-1 = 212-3-1 = 208$

③ Null deviance $\Rightarrow$ let's get the mean of response first

$\Rightarrow \bar{Y} = E(Y) = \hat{P} = \frac{79}{212}$. null deviance $= -2\left(\bar{y}\cdot\log\bar{y} + (1-\bar{y})\cdot\log(1-\bar{y})\right)n$

$\Rightarrow 279.987$

④ df of null deviance : $n-1 = 211$

⑦ Since $(X^TWX)^{-1}$ is symmetric, it is -0.255928180

⑧ $SE(\hat{\beta}_3) = \sqrt{(X^TWX)^{-1}_{4,4}}$ $\Rightarrow$ $(0.3131)^2 = (X^TWX)^{-1}_{4,4} = 0.0980$

(b) $\hat{p}_i = \dfrac{e^{x_i^T\hat{\beta}}}{1+e^{x_i^T\hat{\beta}}}$ $\Rightarrow x_i^T\hat{\beta} = [1, \log(265), \log(26), 3.5] \begin{bmatrix} 0.6864 \\ -0.9040 \\ 0.5027 \\ 1.1153 \end{bmatrix}$

$= 1.17014$

$e^{x_i^T\hat{\beta}} = 3.24833$

$\Rightarrow \hat{p}_i = 0.764613$ $\Rightarrow$ predicted probability of new site

(c) ① Residual deviance will decrease / stay the same since Residual deviance $= -2\cdot\sum_{i=1}^{n}(y_i\log p_i + (1-y_i)\log(1-p_i))$ and we treat it as residual sum of squares in logistic regression. As we add more variable, residual sum of squares decrease and so does residual deviance in here

② null deviance will be the same, since null deviance only take into the model with the only intercept, it is not affected by the number of variables in the model.

null deviance : $-2\cdot n(\bar{y}\log\bar{y} + (1-\bar{y})\log(1-\bar{y}))$ $\Rightarrow$ only depends on the mean of response

4 (a) $\log\frac{p_i}{1-p_i} = \theta_i = x_i^T\beta = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$

$y_i \sim Ber(p_i)$ where $p_i = \dfrac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$

since $\beta^{(m+1)} = \beta^{(m)} + (X^TWX)^{-1}X^T(Y-P)$

$= \beta^{(m)} + (X^TWX)^{-1}X^TY - (X^TWX)^{-1}X^TP$

and it only depends on Y for $(X^TWX)^{-1}\underline{X^TY}$ term.

From here, we can see that β depends on Y only though the vector X^TY

(b) $\log\frac{p_i}{1-p_i} = \theta_i = x_i^T\beta \Rightarrow \hat{p}_i = \dfrac{e^{x_i^T\hat{\beta}}}{1+e^{x_i^T\hat{\beta}}} = \dfrac{e^{\hat{\beta}_0+\hat{\beta}_1x_{i1}+\cdots+\hat{\beta}_px_{ip}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1x_{i1}+\cdots+\hat{\beta}_px_{ip}}}$

(c) $\frac{1}{n}\sum_{i=1}^{n} \hat{P}_i = \bar{\hat{P}}$ and $E(Y) = \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

prove $\sum_{i=1}^{n} \hat{P}_i = \sum_{i=1}^{n} y_i$ ($\because y$ is binary $(0,1)$)

$\Rightarrow \frac{1}{n}\sum^{n} \hat{P}_i = \frac{1}{n}\sum^{n} y_i = \bar{Y} = E(Y) = \bar{\hat{P}}$

Since the mean of the fitted values is equal to $E(Y)$

$\sum_{i=1}^{n} \hat{P}_i = \sum_{i=1}^{n} y_i$ $\Rightarrow X^T(Y-\hat{P})=0 \Rightarrow \begin{pmatrix} 1 & X_1 & X_2 \\ \vdots & \vdots & \vdots \\ 1 & & \end{pmatrix}^T \begin{pmatrix} Y-P \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \cdots 1 \\ \end{pmatrix}\begin{pmatrix} Y-P \end{pmatrix}$

$\Rightarrow$ first row $\Rightarrow$ $Y-P=0 \Rightarrow \sum Y_i = \sum P_i$

(d) Residual deviance : $-2\log($maximum value of likelihood in model m$)$ $= -2\cdot\sum_{i=1}^{n} y_i \log p_i + (1-y_i)\log(1-p_i)$

5 (a) $n=4601$, $p=6$

① $\frac{\hat{\beta}_0 - 0}{S.E(\hat{\beta}_0)} = \frac{4.11947}{0.36342} = 11.335$

② $\frac{\hat{\beta}_5 - 0}{S.E(\hat{\beta}_5)} = \frac{x-0}{0.02800} = 12.345 \Rightarrow 0.34566 = x$

③ null deviance $= (\bar{y}\log\bar{y} + (1-\bar{y})\cdot\log(1-\bar{y}))\cdot n\cdot(-2)$

$\Rightarrow \bar{y} = \frac{1813}{4601}$ so it is 6170.15.

④ df of null deviance : $n-1 = 4600$

⑤ df of residual deviance : $n-p-1 = 4601-6-1 = 4594$

⑥ AIC $= -2\log($maximum likelihood value in model m$) + 2(1+P)$

$=$ residual deviance $+ 2(1+P)$

$= 3246.1 + 2\cdot 7 = 3259.1$

(b) $\hat{p}_i = \dfrac{e^{x_i^T \hat{\beta}}}{1+e^{x_i^T \hat{\beta}}}$ $\hat{\beta} = \begin{bmatrix} 4.11447 \\ 0.30228 \\ 0.32586 \\ 0.00984 \\ 0.34563 \\ 0.16947 \\ -0.11418 \end{bmatrix}$ $x_T = \begin{bmatrix} 1 \\ \lg(157) \\ \lg(0.868 + 0.001) \\ \lg(2.894 + 0.001) \\ \lg(0.001) \\ \lg(0.001) \\ \lg(0.001) \end{bmatrix}$

$x_T^T \hat{\beta} = 3.13016$

$e^{x_T^T \hat{\beta}} = 22.8776$

$\hat{p}_T \approx 0.95812$

(c) Since residual deviance plays roll like RSS in logistic Regression, the small residual deviance the better fit when the models have the same number of predictors.

Since M_1 and M_2 have 6 predictors (except intercept), M_1 is better because its residual deviance is smaller than M_2's. I would choose M_1 over M_2.