Titanic Guided Project

Data Processing

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```r
library(naivebayes)
```

```
naivebayes 1.0.0 loaded
```

```
For more information please visit:
```

```
https://majkamichal.github.io/naivebayes/
```

```r
library(MASS)
```

```

Attaching package: 'MASS'
```

```
The following object is masked from 'package:dplyr':

    select
```

```r
X <- read.csv("train.csv",header = T)
X <- X %>% dplyr::select(Survived,Pclass,Sex,Age,Fare,SibSp,Parch)
```

Imputing age

```r
meanAge = mean(X$Age[!is.na(X$Age)])
for(i in 1:length(X$Age)){
  if(is.na(X$Age[i]) == 1){X$Age[i] = meanAge}
}
```

We will make sex numerical

```r
for(i in 1:891){
  if(X$Sex[i] == "male"){X$Sex[i] = 1}
  else{X$Sex[i] = 0}
}
X$Survived <- as.integer(X$Survived)
X$Sex <- as.integer(X$Sex)
```

splitting the data (notice 80% 0f 891 $\approx$ 713)

```r
set.seed(533)
trainIndex <- sample(1:891, size = 713, replace = F)
Xtrain <- X[trainIndex,]
Xtest <- X[-trainIndex,]
```

Logistic Regression

fit to a logistic regression

```r
# make sure Survived is a factor
Xtrain$Survived <-as.factor(Xtrain$Survived)
# train the logit
fitLogit <- train(Survived ~ .,
                  data= Xtrain,
                  method = "glm",
                  family = "binomial")
# print summary
summary(fitLogit)
```

```
Call:
NULL

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  5.418964   0.617709   8.773  < 2e-16 ***
Pclass      -1.110354   0.155855  -7.124 1.05e-12 ***
Sex         -2.964240   0.230730 -12.847  < 2e-16 ***
Age         -0.043505   0.009105  -4.778 1.77e-06 ***
Fare         0.002363   0.002521   0.937  0.34854    
SibSp       -0.443198   0.132048  -3.356  0.00079 ***
Parch       -0.073278   0.140618  -0.521  0.60229    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 959.55  on 712  degrees of freedom
Residual deviance: 614.40  on 706  degrees of freedom
AIC: 628.4
```

```
Number of Fisher Scoring iterations: 5
```

looking at the above summary, it seems as though Pclass and Sex have a major influence on survival.

```
# doing this to get the accuracy
Xtest <- Xtest %>%
  mutate(logityhat = predict(fitLogit, newdata =., type = "raw"))
# confusion matrix
Xtest$Survived <- as.factor(Xtest$Survived)
confusionMatrix(Xtest$logityhat,Xtest$Survived)$table -> cm.log
# display for accuracy, precision, recall, f1score
table1 <- function(tab,name, printtnr = F){
  tab[1,1] -> tn
  tab[2,2] -> tp
  tab[1,2] -> fp
  tab[2,1] -> fn
  a = (tp+tn)/(tp+tn+fp+fn)
  p = tp/(tp+fp)
  r = tp/(tp+fn)
  f1 = 2*p*r/(p+r)
  tnr = tn/(tn+fp)
  if(printtnr == 0){
    sprintf("%s || Accuracy: %f | Precision: %f | Recall(TPR): %f | F1: %f",name,a,p,r,f1)
  } else{
    sprintf("%s || Accuracy: %f | Precision: %f | Recall(TPR): %f | F1: %f | TNR: %f",name,a
  }
  }

table1(cm.log,"Logistic Regression")
```

```
[1] "Logistic Regression || Accuracy: 0.747191 | Precision: 0.701754 | Recall(TPR): 0.588235
```

Linear Discriminant Analysis (LDA) or Quadratic Discriminant Analysis (QDA)

let's fit a QDA

```
# training on the QDA
fitQda <- (train(method = "qda", Survived ~ ., data = Xtrain))
# accuracy
Xtest <- Xtest %>%
  mutate(qdayhat = predict(fitQda, newdata =., type = "raw"))
# comparing accuracy
confusionMatrix(Xtest$logityhat,Xtest$Survived)$overall["Accuracy"]
```

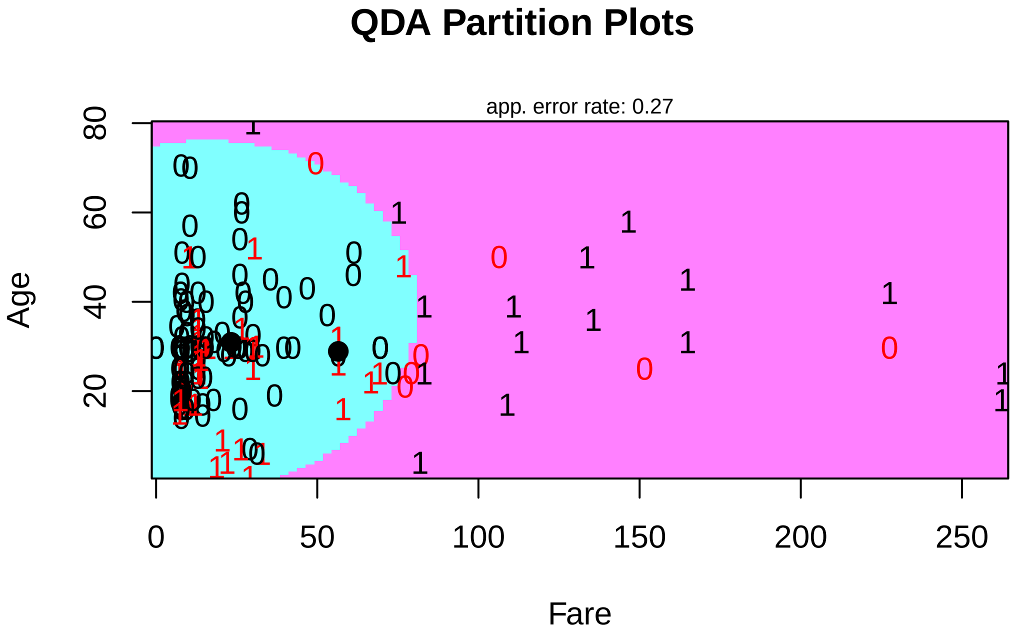
```
Accuracy
0.747191
```

```
confusionMatrix(Xtest$qdayhat,Xtest$Survived) -> cm.qda
cm.qda$overall["Accuracy"]
```

```
 Accuracy
0.7640449
```

the QDA accuracy is better than that of the Logistic. Although this is strange as QDA assumes normality in the predictors and I didn't check for that.

```
library(klaR)
partimat(formula = Survived ~ Age + Fare,data=Xtest,method="qda",main="QDA Partition Plots")
```



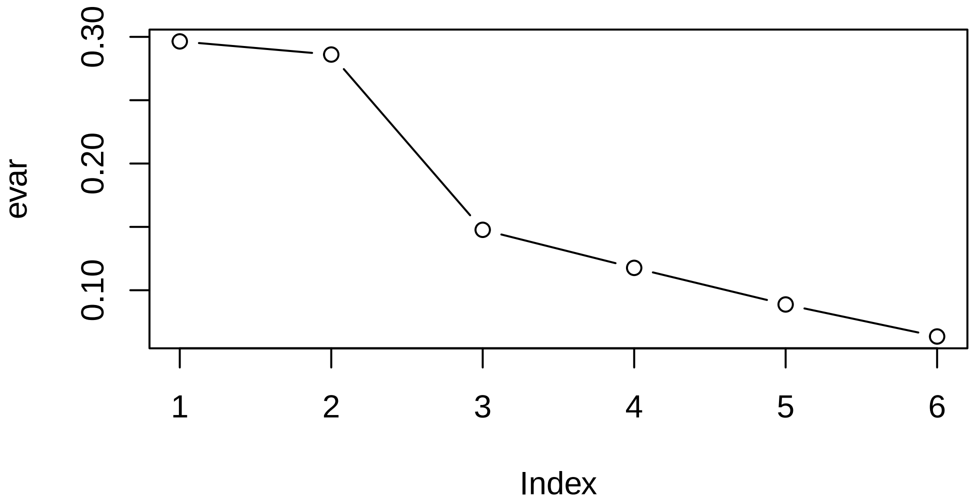
Naive Bayes

```
fitNb <- train(method = "naive_bayes", Survived ~ . , data = Xtrain)
Xtest <- Xtest %>%
  mutate(nbyhat = predict(fitNb, newdata =., type = "raw"))
confusionMatrix(Xtest$nbyhat,Xtest$Survived) -> cm.nb
cm.nb$overall["Accuracy"]
```

```
Accuracy
0.747191
```

Naive Bayes has a similar accuracy to logistic regression. Let's see if certain variables improve it's performance metrics.

```
pca <- prcomp(Xtrain[,2:7], scale. = T)
evar = pca$sdev^2 / sum(pca$sdev^2)
plot(evar, type = 'b')
```



looks like 3 components are relevant for our analysis, maybe we can see which ones

```
pca$rotation[, 1:3]
```

```
              PC1         PC2         PC3
Pclass -0.2881179 -0.60622454 -0.07153728
Sex    -0.3923492 -0.02697763  0.88199087
Age    -0.1544670  0.53855145  0.09766212
Fare    0.4883438  0.39447871  0.22518505
SibSp   0.4585138 -0.35557015  0.38240226
Parch   0.5389579 -0.24430058  0.10245408
```

rotation tells me that Fare is important in three components as it places in all three, so it must be important. SibSp places in two components, while Sex seems to be very influential in component three, so I will choose it.

```
fitNb2 <- train(method = "naive_bayes", Survived ~ Fare + SibSp + Sex , data = Xtrain)
Xtest <- Xtest %>%
  mutate(nb2yhat = predict(fitNb2, newdata =., type = "raw"))
confusionMatrix(Xtest$nb2yhat,Xtest$Survived) -> cm.nb2
table1(cm.nb2$table,"Naive Bayes with less variables")
```

```
[1] "Naive Bayes with less variables || Accuracy: 0.758427 | Precision: 0.666667 | Recall(TP
```

```
table1(cm.nb$table,"Naive Bayes")
```

```
[1] "Naive Bayes || Accuracy: 0.747191 | Precision: 0.701754 | Recall(TPR): 0.588235 | F1: 0
```

slight improvement in accuracy and TNR, making it relatively better than all other models so far. Let's see if all other models improve after this variables reduction.

```
# log regression
fitLogit <- train(Survived ~ Fare + SibSp + Sex ,
                  data= Xtrain,
                  method = "glm",
                  family = "binomial")
Xtest$logityhat <- predict(fitLogit, newdata =Xtest, type = "raw")
confusionMatrix(Xtest$logityhat,Xtest$Survived)$table -> cm.log

# QDA
fitQda <- (train(method = "qda", Survived ~ Fare + SibSp + Sex , data = Xtrain))
Xtest$nbyhat <- predict(fitQda, newdata =Xtest, type = "raw")
confusionMatrix(Xtest$nbyhat,Xtest$Survived)$table -> cm.qda2

# performance metrics
table1(cm.log,"Logistic Regression with less variables",T)
```

```
[1] "Logistic Regression with less variables || Accuracy: 0.752809 | Precision: 0.649123 | R
```

```
table1(cm.qda2,"QDA with less variables",T)
```

```
[1] "QDA with less variables || Accuracy: 0.758427 | Precision: 0.666667 | Recall(TPR): 0.61
```

```
table1(cm.nb2$table,"Naive Bayes with less variables",T)
```

```
[1] "Naive Bayes with less variables || Accuracy: 0.758427 | Precision: 0.666667 | Recall(TP
```

Only using the three variables, Naive Bayes and Logistic Regression improved, however QDA seems to have taken a dive in performance considering the previous accuracy score was around .76.

Model Comparison

```
# performance metrics
table1(cm.log,"Logistic Regression with less variables",T)
```

```
[1] "Logistic Regression with less variables || Accuracy: 0.752809 | Precision: 0.649123 | Re
```

```
table1(cm.qda$table,"QDA",T)
```

```
[1] "QDA || Accuracy: 0.764045 | Precision: 0.736842 | Recall(TPR): 0.608696 | F1: 0.666667
```

```
table1(cm.nb2$table,"Naive Bayes with less variables",T)
```

```
[1] "Naive Bayes with less variables || Accuracy: 0.758427 | Precision: 0.666667 | Recall(TP
```

model	Accuracy	Precision	Recall(TPR)	F1	TNR
Logistic Regression (less variables)	0.752809	0.649123	0.606557	0.627119	0.829060
QDA (all variables)	0.764045	0.736842	0.608696	0.666667	0.862385
Naive Bayes (less variables)	0.758427	0.666667	0.612903	0.638655	0.836207

QDA dominates in all of these performance metrics, including accuracy. It also has the highest true positive and and true negative rate; it is the best model for predicting who survived as well as who did not.

Conclusion

Quadratic Discriminant Analysis with all the 6 variables is the model that has the most reliable results. If we are predicting who survived the Titanic, given these 6 particular variables from the data, then this the model I would move forward with.