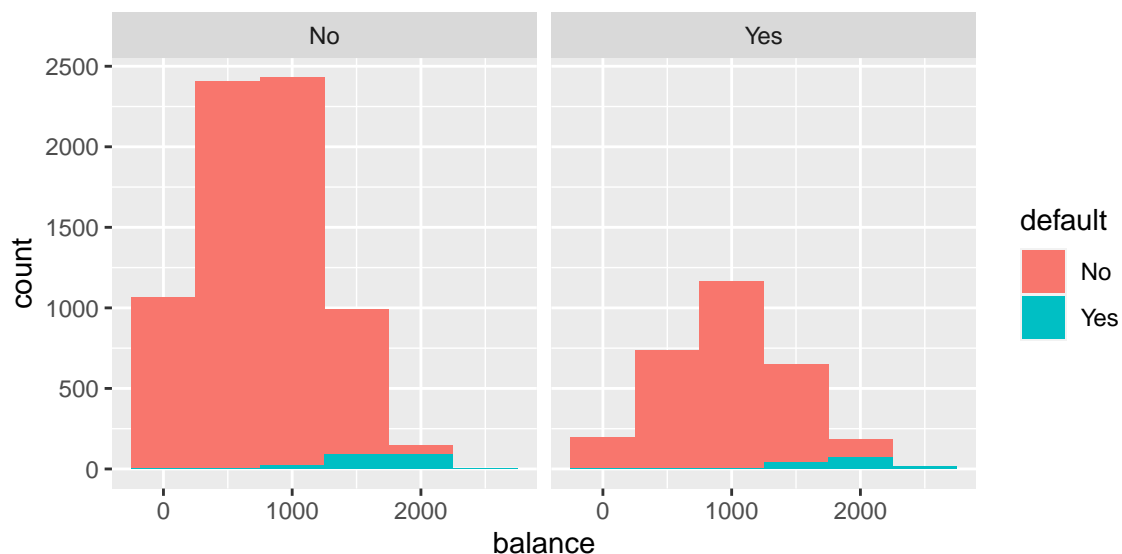


## Classification 2 Homework

```
library(tidyverse)
library(MASS)
library(caret)
library(kableExtra)
library(e1071)
library(ISLR)
library(rpart)
library(DMwR)
```

### Data

Using the Default data from ISLR



### Comparison of Algorithms

Use the following template that compares algorithms and then adds a SMOTE section and compares again. Follow this template, except use the Employee Turnover data.

#### Logistic Regression

```
glModS <- glm(default ~ student + balance + income, data = xTrain, family = binomial)
glmPred <- predict(glModS, type = "response", newdata = xTest)
xTest$GLM = if_else(glmPred < .5, "No", "Yes")

CM = confusionMatrix(factor(xTest$GLM), factor(xTest$default), positive = "Yes")

Summary = data.frame(Algorithm = "GLM",
```

```

Sensitivity = CM$byClass[1],
Specificity = CM$byClass[2],
PosPredVal = CM$byClass[3],
NegPredVal = CM$byClass[4],
Prevalence = CM$byClass[8])

```

## Linear Discriminant Analysis

```

lda.fit <- lda(default ~ student + balance + income, xTrain)
lda.pred <- predict(lda.fit, xTest)

xTest$LDA = lda.pred$class

CM = confusionMatrix(xTest$LDA, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "LDA",
  Sensitivity = CM$byClass[1],
  Specificity = CM$byClass[2],
  PosPredVal = CM$byClass[3],
  NegPredVal = CM$byClass[4],
  Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

## Naive Bayes

```

NBmodel <- naiveBayes(default ~ student + balance + income, data = xTrain)
xTest$NB <- predict(NBmodel, xTest, prob = TRUE)

CM = confusionMatrix(xTest$NB, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "NB",
  Sensitivity = CM$byClass[1],
  Specificity = CM$byClass[2],
  PosPredVal = CM$byClass[3],
  NegPredVal = CM$byClass[4],
  Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

## Decision Tree

```

Treefit <- rpart(default ~ student + balance + income,
  data = xTrain,
  method="class")

xTest$Tree = predict(Treefit, type = "class", newdata = xTest) # factor

CM = confusionMatrix(xTest$Tree, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "Tree",
  Sensitivity = CM$byClass[1],

```

```

Specificity = CM$byClass[2],
PosPredVal = CM$byClass[3],
NegPredVal = CM$byClass[4],
Prevalence = CM$byClass[8])

```

```
Summary = bind_rows(Summary, Summaryadd)
```

## Support Vector Machine

```

svmMod <- svm(default ~ student + balance + income, data = xTrain)
xTest$SVM <- predict(svmMod, xTest)

CM = confusionMatrix(xTest$SVM, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "SVM",
  Sensitivity = CM$byClass[1],
  Specificity = CM$byClass[2],
  PosPredVal = CM$byClass[3],
  NegPredVal = CM$byClass[4],
  Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

## SMOTE Sampling

### Data Creation

```

smoteData <- SMOTE(default ~ student + balance + income, data = Default, perc.over = 350, perc.under=130,
prop.table(table(smoteData$default)))

```

### Logistic Regression with SMOTE

```

glModSmote <- glm(default ~ student + balance + income, data = smoteData, family = binomial)
glmPredSmote <- predict(glModSmote, type = "response", newdata = xTest)
xTest$GLMSmote = if_else(glmPredSmote < .5, "No", "Yes")

CM = confusionMatrix(factor(xTest$GLMSmote), factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "GLMSmote",
  Sensitivity = CM$byClass[1],
  Specificity = CM$byClass[2],
  PosPredVal = CM$byClass[3],
  NegPredVal = CM$byClass[4],
  Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

### LDA with SMOTE

```

lda.fit <- lda(default ~ student + balance + income, smoteData)
lda.pred <- predict(lda.fit, xTest)

```

```

xTest$LDASmote = lda.pred$class

CM = confusionMatrix(xTest$LDASmote, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "LDASmote",
                        Sensitivity = CM$byClass[1],
                        Specificity = CM$byClass[2],
                        PosPredVal = CM$byClass[3],
                        NegPredVal = CM$byClass[4],
                        Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

### Naive Bayes with SMOTE

```

model <- naiveBayes(default ~ student + balance + income, data = smoteData)
xTest$NBSmote <- predict(model, xTest, prob = TRUE)

CM = confusionMatrix(xTest$NBSmote, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "NBSmote",
                        Sensitivity = CM$byClass[1],
                        Specificity = CM$byClass[2],
                        PosPredVal = CM$byClass[3],
                        NegPredVal = CM$byClass[4],
                        Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

### Decision Tree with SMOTE

```

TreefitSmote <- rpart(default ~ student + balance + income,
                      data = xTrain,
                      method="class")

xTest$TreeSmote = predict(TreefitSmote, type = "class", newdata = xTest) # factor

CM = confusionMatrix(xTest$TreeSmote, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "TreeSmote",
                        Sensitivity = CM$byClass[1],
                        Specificity = CM$byClass[2],
                        PosPredVal = CM$byClass[3],
                        NegPredVal = CM$byClass[4],
                        Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

### SVM with SMOTE

```

svmMod <- svm(default ~ student + balance + income, data = smoteData)
xTest$SVMSmote <- predict(svmMod, xTest)

CM = confusionMatrix(xTest$SVMSmote, factor(xTest$default), positive = "Yes")

Summaryadd = data.frame(Algorithm = "SVMSmote",
                        Sensitivity = CM$byClass[1],
                        Specificity = CM$byClass[2],
                        PosPredVal = CM$byClass[3],
                        NegPredVal = CM$byClass[4],
                        Prevalence = CM$byClass[8])

Summary = bind_rows(Summary, Summaryadd)

```

## Results and Review

```

knitr::kable(Summary) %>%
  kable_styling(full_width = F, bootstrap_options = "striped", font_size = 9)

```

Algorithm	Sensitivity	Specificity	PosPredVal	NegPredVal	Prevalence
GLM	0.2960526	0.9961019	0.7500000	0.9728426	0.038
LDA	0.2236842	0.9974012	0.7727273	0.9701719	0.038
NB	0.2171053	0.9945426	0.6111111	0.9698429	0.038
Tree	0.2960526	0.9963617	0.7627119	0.9728495	0.038
SVM	0.1250000	0.9989605	0.8260870	0.9665577	0.038
GLMSmote	0.8552632	0.8666840	0.2021773	0.9934465	0.038
LDASmote	0.8881579	0.8300416	0.1711027	0.9947057	0.038
NBSmote	0.8355263	0.8523909	0.1827338	0.9924357	0.038
TreeSmote	0.2960526	0.9963617	0.7627119	0.9728495	0.038
SVMSmote	0.8486842	0.8731809	0.2090762	0.9932013	0.038