title: "Notebook 4" author: "Group 10 (Umar, Cory, Caroline, Benji)" date: "10/9/2022" output: pdf_document: default

html_notebook: default

Introduction

About this set

Bank marketing was downloaded from archive.ics.uci.edu. The set includes direct marketing campaign phone calls from a portugese banking institution.

80:20 Training and Test Sets

In the code block below users can obtain the code used to read a dataset in csv format and install the accompanying tools to split dataset into training and testing sets.

```{r}

# Code to split data into training and test datasets

# Importing data sets

library(caret) library(class) library(tree) library(MASS)

data <- read.csv("bank-additional-full.csv") data <- subset(data, select = -c(pdays, duration, default)) replace_unknowns <- function(df) { for(col in colnames(df)) { if(has_unknown(df,col)) { n_unk <- sum(df[,col]=="unknown") idx <- which(df[,col]=="unknown") df[idx,col] <- sample(col[!col=="unknown"],n_unk,replace=TRUE) } } df } cats <- names(data)[sapply(data,is.character)] encode <- function(df,col) { as.numeric(factor(df[,col]))-1 } for(cat in cats) { data[,cat] <- encode(data,cat) } data$deposit <- as.factor(data$deposit)

### Applying PCA

```{r}

i <- sample(1:150, 100, replace = FALSE)

train <- data[i,]
test <- data[-i,]

set.seed(2354)

pcaOut <- preProcess(train[,1:4], method = c("center", "scale", "pca"))
pcaOut
```

PCA Plotting

```{r}

trainPc <- predict(pcaOut, train[, 1:4]) testPc <- predict(pcaOut, test[,])

plot(testPc$PC1, testPc$PC2, pch = c(23, 21, 22)[unclass(testPc$Species)], bg = c("red", "green","blue")[unclass(test$Species)])

```
```{r}

trainDf <- data.frame(trainPc$PC1, trainPc$PC2, train$deposit)
testDf <- data.frame(testPc$PC1, testPc$PC2, test$deposit)

set.seed(2354)

pred <- knn(train = trainDf[,1:2], test = testDf[,1:2], cl = trainDf[,3], k = 3)
mean(pred == test$deposit)
```

```{r}

train <- subset(train, select = -c(contact, month, day, previous, poutcome, evr, cpi, cci, euribor3m, employees))

colnames(trainDf) <- c("PC1", "PC2", "deposit") colnames(testDf) <- c("PC1", "PC2", "deposit")

set.seed(2354)

tre <- tree(deposit~., data = trainDf) plot(tre) text(tre, cex = 0.5, pretty = 0)

pred <- predict(tre, newdata = testDf, type = "class") mean(pred == test$deposit)

LD <- lda(deposit~., data = train)
```

Applying LDA

```{r}

LD <- lda(deposit~., data = train)

LD$means
```

Predict on test

```{r}

LDpred <- predict(LD, newdata = test, type = "class")

mean(LDpred$class == test$deposit)

# output is too long

cat("Levels: 0 1")
```
```

```
### LDA Plotting

```{r}

plot(LDpred$x[,1], pch = c(23, 21, 22)[unclass(LDpred$class)],
     bg = c("red","green","blue")[unclass(testPc$deposit)])
```

Applying Classification

PCA Classification

```{r} trainDf$deposit <- as.factor(trainPc$deposit)

pclass_model <- glm(deposit ~ marital+housing+loan, family = "binomial", data = trainPc)

summary(pclass_model)

```
LCA Classification
```{r}
LD$deposit <- as.factor(training_set$deposit)

Lclass_model <- glm(deposit ~ marital+housing+loan, family = "binomial", data = LD)

summary(Lclass_model)
```

Conclusion

```{r} library(modelr)

mse <- Lclass_model %>% mutate(error = LDpred - Lclass_model, sq.error = error^2) %>% summarise(mse = mean(sq.error))

mse <- pclass_model %>% mutate(error = pred - pclass_model, sq.error = error^2) %>% summarise(mse = mean(sq.error))

```
```