Notebook 4

Group 10 (Umar, Cory, Caroline, Benji)

10/9/2022

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.

```
# Code to split data into training and test datasets

# Importing data sets

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
library(class)
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.1.3
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
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)
```

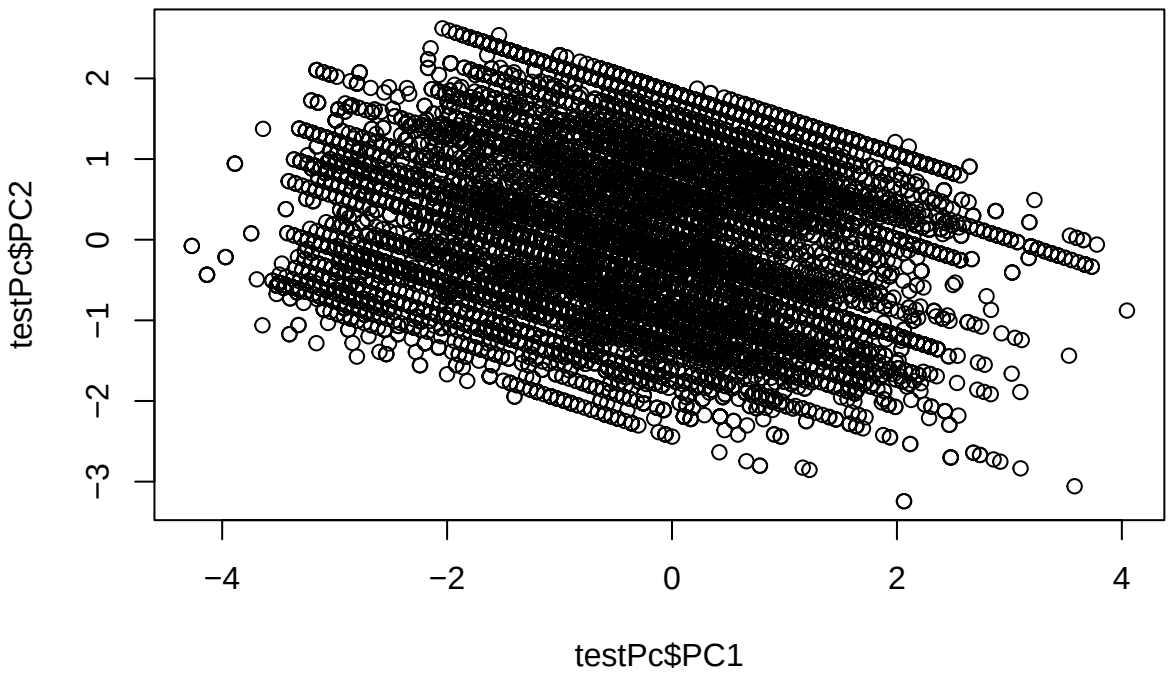
PCA

```
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
```

```
## Created from 100 samples and 4 variables
##
## Pre-processing:
##   - centered (4)
##   - ignored (0)
##   - principal component signal extraction (4)
##   - scaled (4)
##
## PCA needed 4 components to capture 95 percent of the variance
```

PCA Plotting

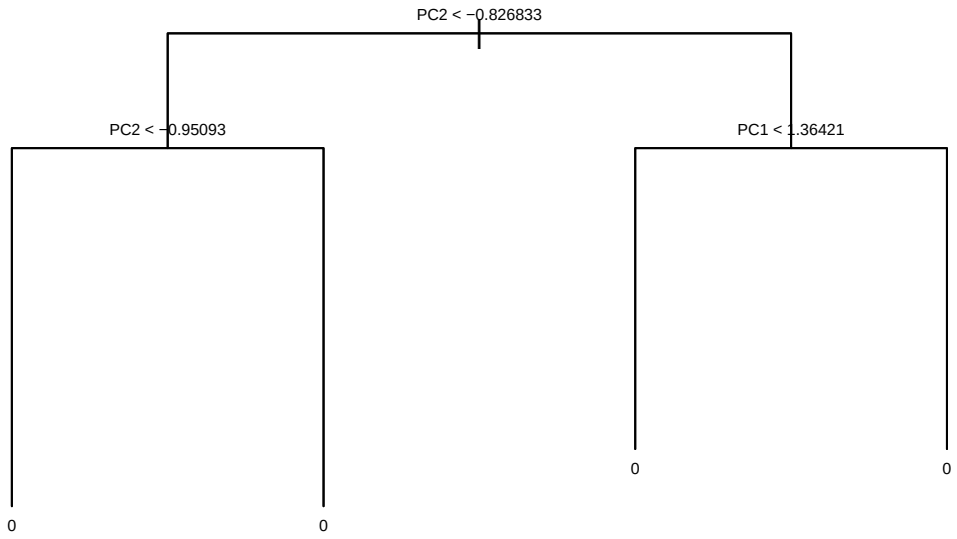
```
trainPc <- predict(pcaOut, train[, 1:4])
testPc <- predict(pcaOut, test[,])

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




```
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)
```

```
## [1] 0.8871447
```

```
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)
```

```
## [1] 0.8871447
```

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

LDA

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

LD$means
```

```
##        age      job   marital education  housing      loan campaign
## 0 45.13402 3.597938 1.0309278  3.608247 1.051546 0.3092784 1.134021
## 1 43.66667 4.000000 0.6666667  3.666667 2.000000 0.0000000 1.000000
```

Predict on test

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

mean(LDpred$class == test$deposit)
```

```
## [1] 0.8871447
```

```
# output is too long

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

```
## Levels: 0 1
```

Plot

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