Stat 432 HW 05

Name: Your Name, netID: yournetID

Summer 2024

Include the R code for this HW.

```
knitr::opts_chunk$set(echo = TRUE)
library(ISLR2)
library(GGally)
library(tibble)
library(dplyr)
library(knitr)
library(kableExtra)
library(caret)
#add more libraries as needed.
```

Question 1 (k-NN, tree for classification)

Use `hw5data1.Rdata` to answer this question.

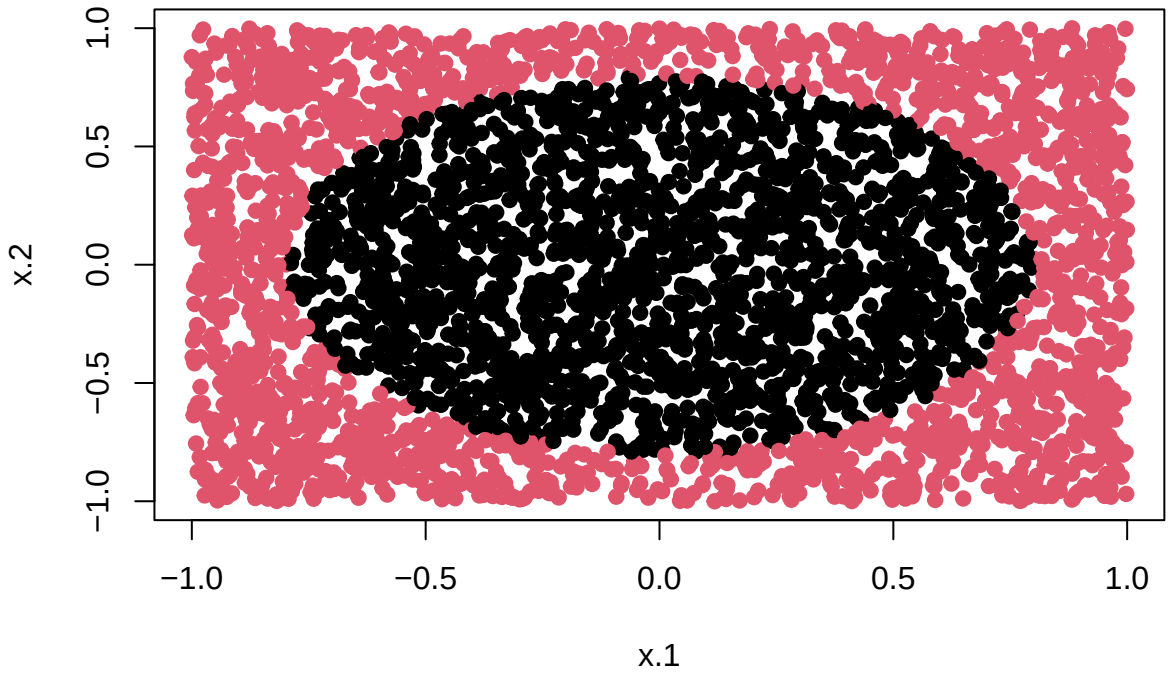
```
load("hw5data1.Rdata") #put this file in your working directory or in the same folder as your HW Rmd fi
str(circle.trn)#your training data
```

```
## tibble [4,000 x 3] (S3: tbl_df/tbl/data.frame)
##  $ x.1    : num [1:4000] -0.479 0.2 0.32 0.888 0.333 ...
##  $ x.2    : num [1:4000] 0.36 0.784 0.581 -0.271 0.532 ...
##  $ classes: Factor w/ 2 levels "1","2": 1 2 1 2 1 1 1 2 2 2 ...
```

```
str(circle.tst)# your test data
```

```
## tibble [1,000 x 3] (S3: tbl_df/tbl/data.frame)
##  $ x.1    : num [1:1000] 0.9886 -0.0187 -0.4419 0.7234 0.1634 ...
##  $ x.2    : num [1:1000] 0.352 -0.234 0.278 0.209 -0.259 ...
##  $ classes: Factor w/ 2 levels "1","2": 2 1 1 1 1 2 1 1 2 1 ...
```

```
plot(x.2~x.1,data=circle.trn,col=circle.trn$classes,pch=19)
```



- `classes` variable: categorical response variable
- `x.1`, `x.2`: feature variables

(a) The given graphic is the plot of data, using different color for different classes. Based on given information, would decision tree work well? Explain why or why not.

(b) Conduct k-NN classification using `train()` function of `caret` package, with 10-fold cross validation. Use the grid of odd numbers, from 1 to 101 for `k`. Choose best k. (For this problem, do not need to consider scaling.)

(c) Conduct tree classification using `rpart()` function of `rpart` package. Use `cp=0` to grow a big tree. Then create the cp-table and cp vs size of plot. (By default, this function use 10-fold cross validation. This number can be controlled using `xval=` option if necessary. No need to change for this HW.) Based on the result, choose the optimal cp value.

(d) Using the models chosen from (b) and (c), refit the models to the whole training data and report test accuracy. Which method is performing better on our test data?

Question 2

This question relates to the `Boston` data set of `ISLR2` package.

```
set.seed(42)
trn.idx=sample(1:nrow(ISLR2::Boston),450)
tst.boston=ISLR2::Boston[-trn.idx,]
trn.boston=ISLR2::Boston[trn.idx,]
```

We are splitting the data into two parts: a testing data that contains 56 observations, and the rest 450 observations as training data.

- The goal is to model `crim` (our response variable) with all the other variables in the data.
- Use `train` function of `caret` package for this question.

(a) Conduct linear regression with 10-fold CV. Report CV error for the chosen parameter. (RMSE or MSE, either way is ok. Just need to be consistent throughout this problem.) In this HW, use:

```
control=trainControl(method = "cv",number=10)

lm.boston<-train(formula,data=data,
                 method = 'lm',
                 trControl=control
                )
```

(b) Conduct k-NN regression with 10-fold CV. Choose optimal tuning parameter. Report CV error for the chosen parameter. Use train function of caret package.

Consider two different pre-processing setups.

- Setup 1: Numeric variables not scaled.
- Setup 2: *Numeric variables are scaled* to have mean 0 and standard deviation 1. You need to add `preProcess = c("center","scale")` option inside the train function.

Which setup and `k` gives the lowest error?

(c) Conduct ridge regression with 10-fold CV. In this HW, use the `train()` function of the `caret` package:

```
control=trainControl(method = "cv",number=10)

lasso<-train(formula,data=data,
                 method = 'glmnet',
                 trControl=control,preProc = c("center","scale"),
                 tuneGrid = expand.grid(alpha = 1, lambda =seq(from=0,to=1,by=0.01))
                 #alpha=1 indicates lasso method. You can choose your own grid of lambda.
                )

ridge<-train(formula,data=data,
                 method = 'glmnet',
                 trControl=control,preProc = c("center","scale"),
                 tuneGrid = expand.grid(alpha = 0, lambda =seq(from=0,to=1,by=0.01))
  #alpha=0 indicates ridge regression method. You can choose your own grid of lambda.

                )
```

Find best tuning parameter for each lasso and ridge regression, and report CV error.

(d) Conduct Bagging, Random Forest, and Boosting with 10-fold CV. Use the `train()` function of the `caret` package. Find best tuning parameter for each methods.

(e) Based on (a)-(d), pick the best method and train your whole training data set using the chosen method and tuning parameter(s). Report the test MSE.