IntroCV: Assignment 3- Auto Data

Cross-validation for Classification

The same ideas apply for classification. Here, the loss function is the error rate so generally you'll be trying to estimate the the error rate (or at least the values of parameters which minimize error rate.

As an example, let's play around with the Auto data. Use year as the response variable.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ISLR) ## for Auto data
library(class) ## for knn
```

Use the Auto Data

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```
with(Auto,table(year))
```

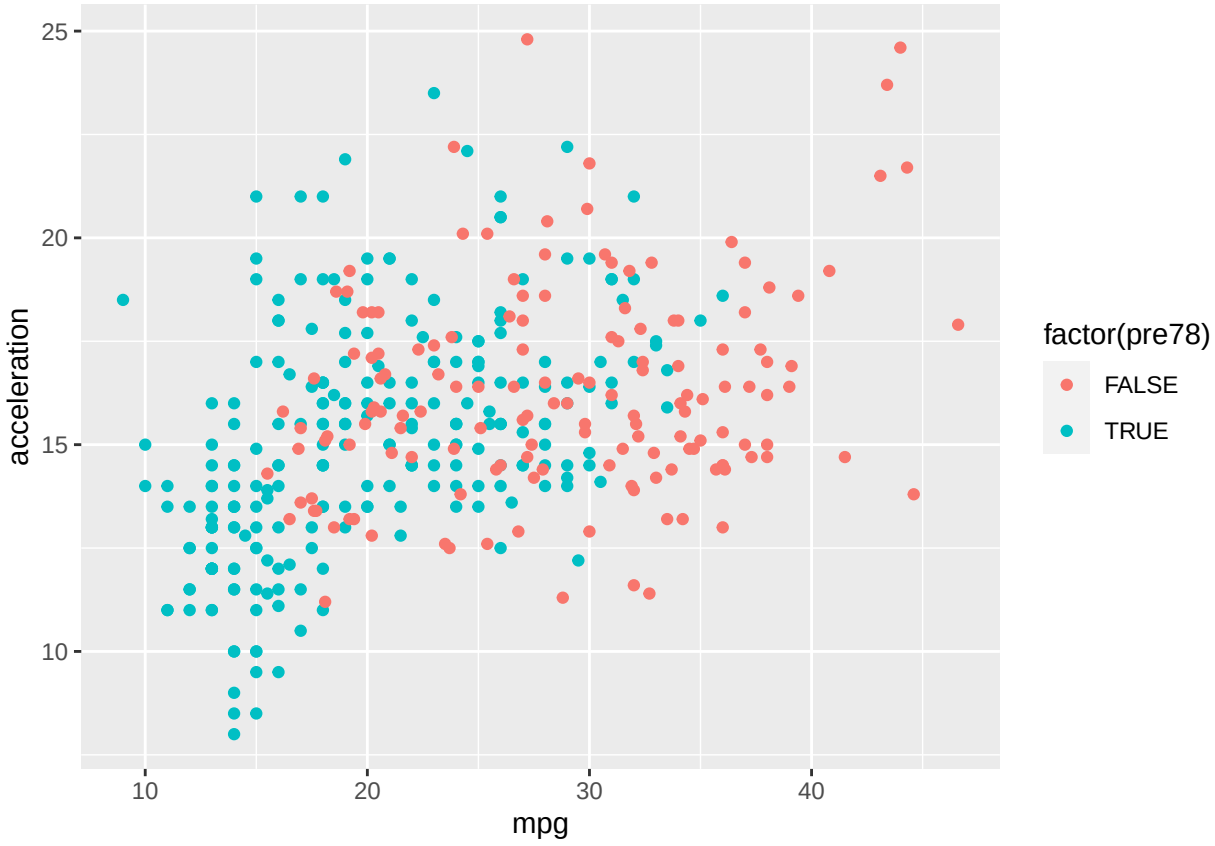
```
## year
## 70 71 72 73 74 75 76 77 78 79 80 81 82
## 29 27 28 40 26 30 34 28 36 29 27 28 30
```

Question: Can we predict which cars were made before 1978?

A quick glance at the data....

```
Auto <- Auto %>%
  mutate(pre78 = year < 78)

Auto%>%
  ggplot()+
  geom_point(aes(mpg,acceleration,color=factor(pre78)))
```



Use Cross-validition with KNN to do the prediction. We need to select the best value of k. Use 10-fold cross validation.

```
numFolds <- 10
n <- nrow(Auto)
folds <- sample(1:numFolds,n,rep=T)
```

Get started with fixed value of k.

```
kval <- 2
errs <- numeric(numFolds)
for(fold in 1:numFolds){
 train.df <- Auto[folds != fold,]
 test.df <- Auto[folds == fold,]
 train.x <- data.matrix(train.df[c("mpg","acceleration")])
 resp.x<- data.matrix(train.df[c("pre78")])
 test.x <- data.matrix(test.df[c("mpg","acceleration")])
 mod.knn <- knn(train.x,test.x,resp.x,k=kval)
 errs[fold] <- with(test.df,mean(pre78 != (mod.knn==1)))
}
mean(errs)
```

```
## [1] 0.3391313
```

Assignment 3: Classification 1

For the predictive question above using KNN, what is the best value of k? Is it k = 1? Use 10-fold cross-validation. To estimate the best possible Caution: For some reason, R is knn is unhappy with k=2.

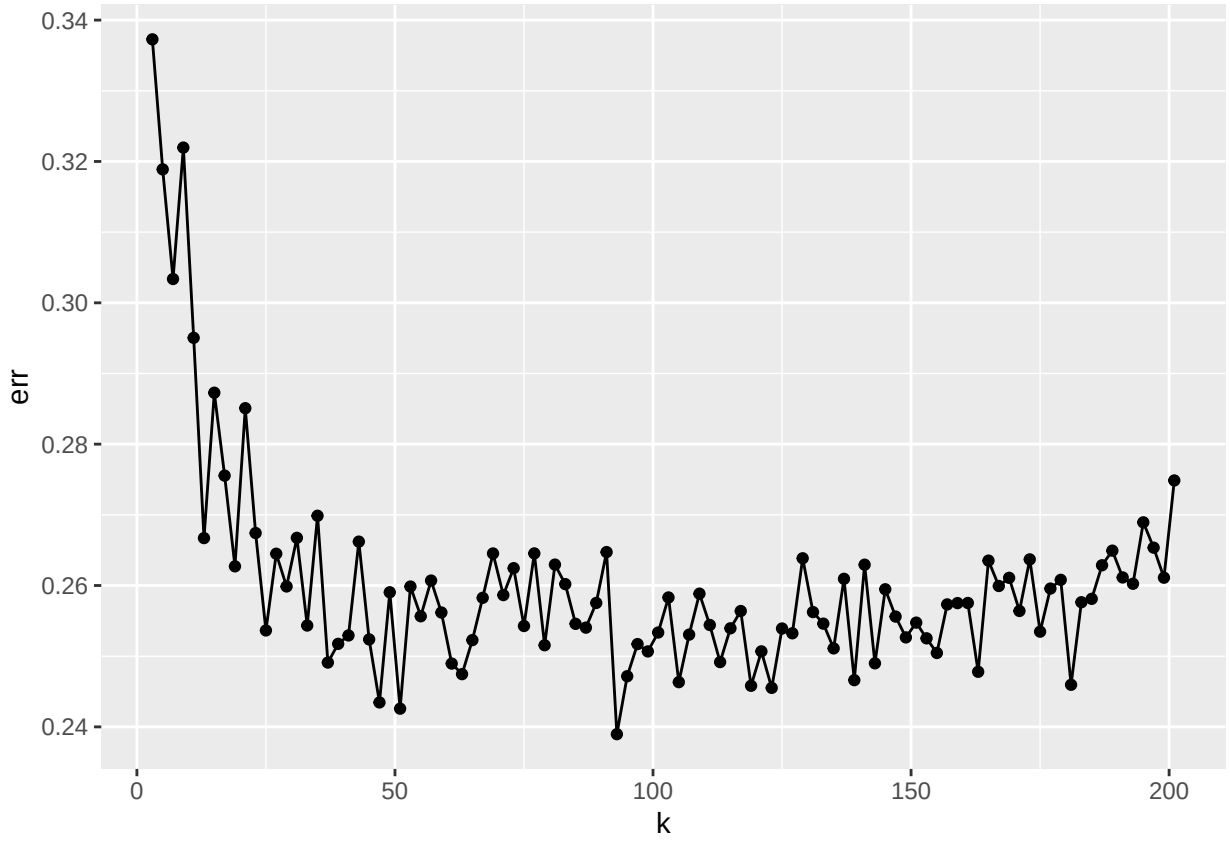
```
kVal <- 40
fold <- 1
errCV <- function(kVal,numFolds=10){
  errs <- numeric(numFolds)
  folds <- sample(1:numFolds,n,rep=T)
  for(fold in 1:numFolds){

    train.df <- Auto[folds != fold,]
    test.df <- Auto[folds == fold,]
    train.x <- data.matrix(train.df[c("mpg","acceleration")])
    resp.x<- data.matrix(train.df[c("pre78")])
    test.x <- data.matrix(test.df[c("mpg","acceleration")])
    mod.knn <- knn(train.x,test.x,resp.x,k=kVal)
    errs[fold] <- with(test.df,mean(pre78 != (mod.knn==1)))
  }
  mean(errs)
}
errCV(260)
```

```
## [1] 0.3687498
```

```
## Skip 2
kMax <- 100
kVals <- 2*(1:kMax)+1
errs <- map_dbl(kVals,errCV)


data.frame(k=kVals,err=errs) %>%
  ggplot()+
  geom_point(aes(k,err))+
    geom_line(aes(k,err))
```




```
## Looks like it settles  down around k=30 or so.
```