STP 598: Homework 3

Antonio Campbell, Sinta Sulistyo, Atta Ullah, Penny Wu

2/22/2021

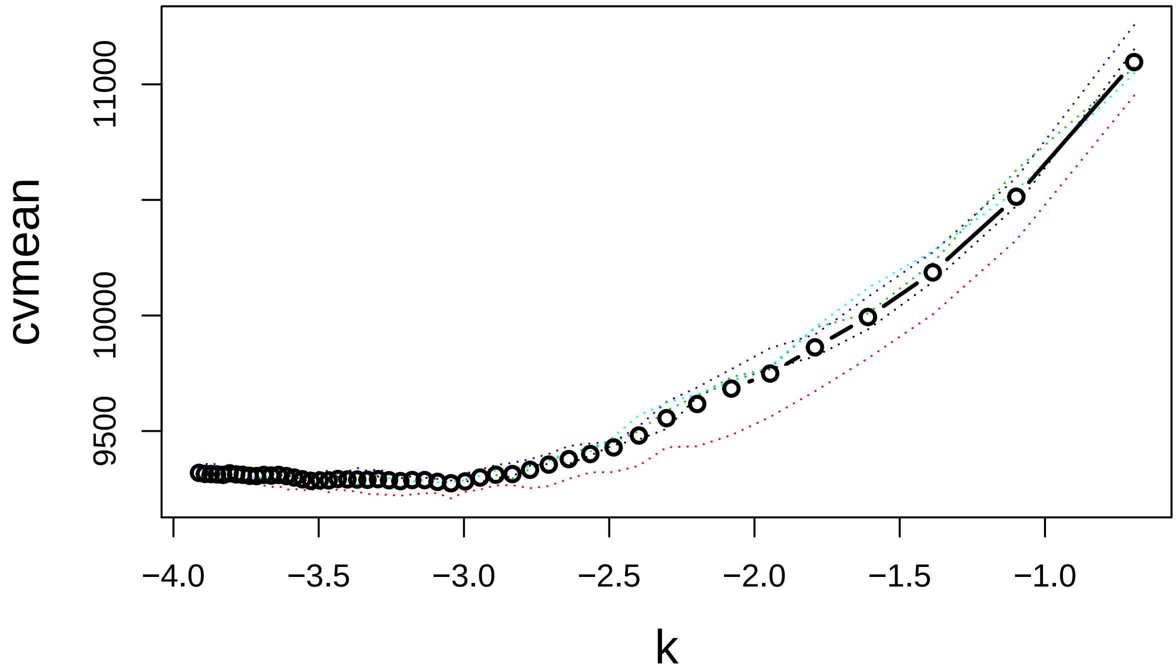
1. Using Cross Validation

We are going to use the used cars data again.Previously, we used the "eye-ball" method to choose k for a kNN fit for mileage predicting price.Use 5-fold cross-validation to choose k. How does your fit compare with the eyeball method?

price	mileage	year
43995	36858	2008
44995	46883	2012
25999	108759	2007
33880	35187	2007
34895	48153	2007
5995	121748	2002

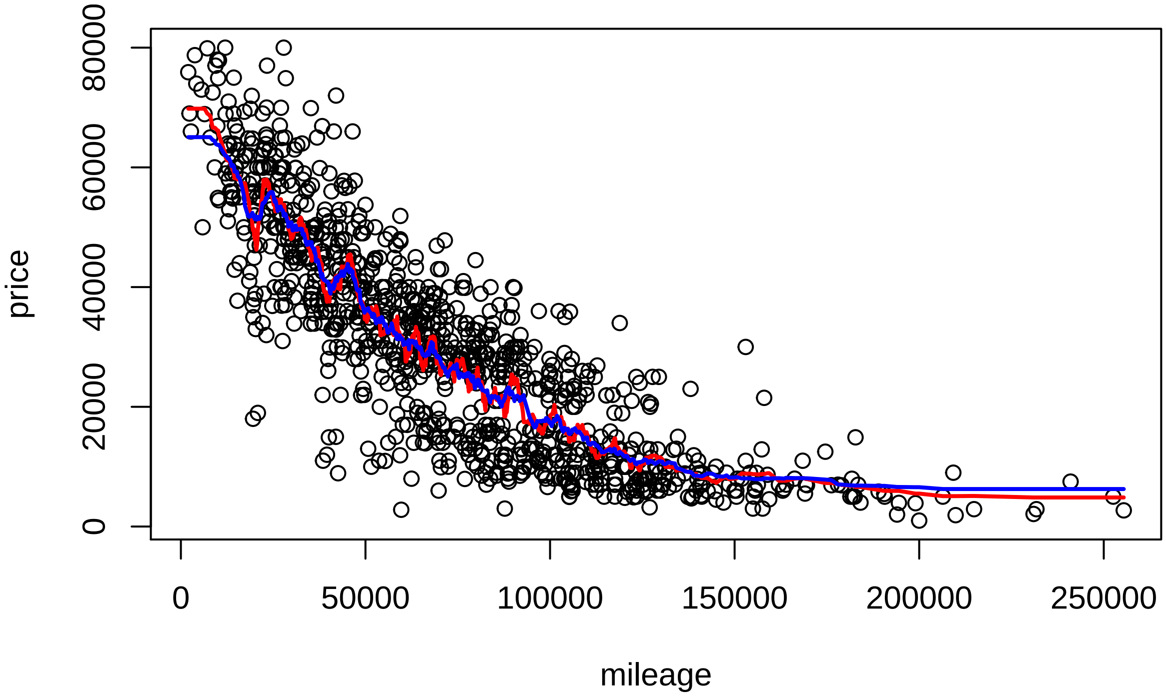
Plot the data and then add the fit using the k you chose using cross-validation and the k you choose by eye-ball.

```r
#get variables we want
x1 = cbind(carDat$mileage)
colnames(x1) = c("mileage")
y = carDat$price

#run cross val several times
set.seed(99)
kv = 2:50 #these are the k values (k as in kNN) we will try
cvmean = rep(0,length(kv)) #will keep average rmse here
ndocv = 5 #number of CV splits to try
n=length(y)
cvmat = matrix(0,length(kv),ndocv) #keep results for each split

for(i in 1:ndocv) {
  cvtemp = docvknn(x1,y,kv,nfold=5)
  cvmean = cvmean + cvtemp
  cvmat[,i] = sqrt(cvtemp/n)
}

cvmean = cvmean/ndocv
cvmean = sqrt(cvmean/n)
```



```r
kbest2 = kbest = kv[which.min(cvmean)]
cat("the best k is: ",kbest,"\n")
```

```
## the best k is:  20
```



Comparing our eyeball fit to the one provided by cross validation we see the two are pretty similar, but we favor the model that is more complicated with $k = 20$. The two fit RMSEs follow:

EyeballedK	BestK
9013.226	8810.022

We see that as the name suggests, we have the best fit if we use the k chosen through cross validation.

Use kNN with the k you chose using cross-validation to get a prediction for a used car with 100,000 miles on it. Use all the observations as training data to get your prediction (given your choice of k).

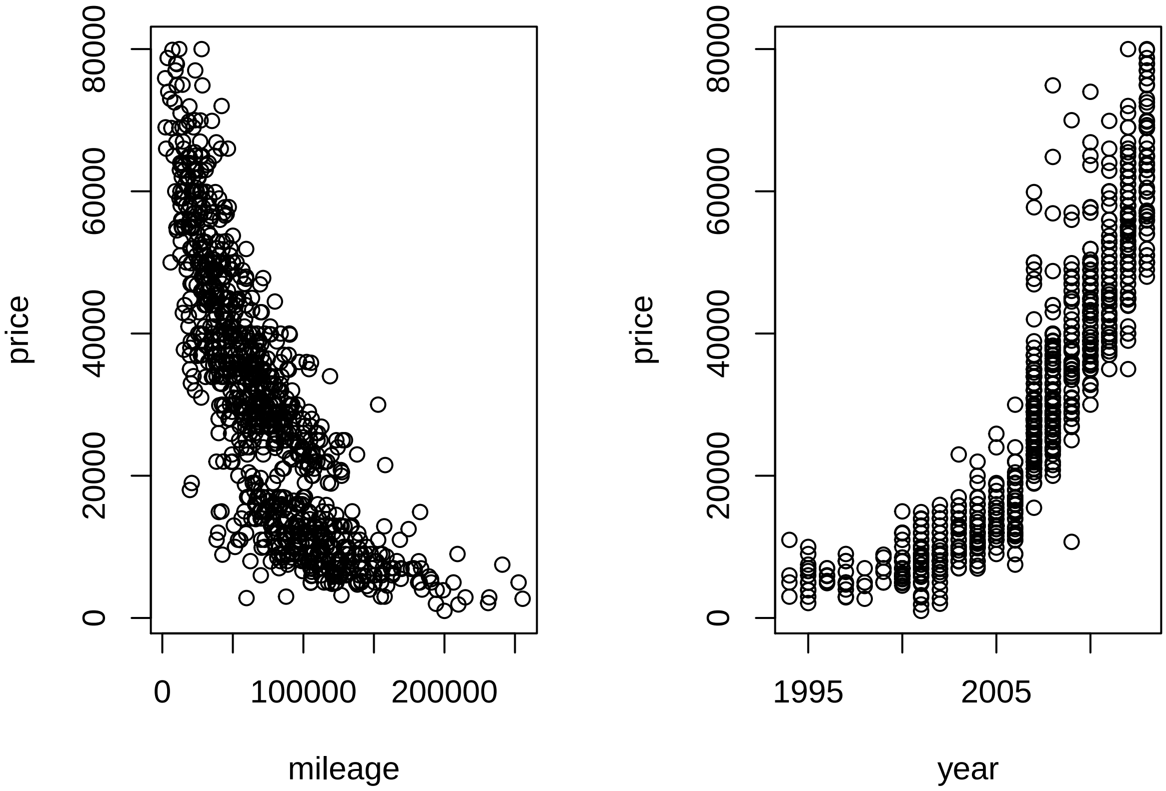
```
near3 = kknn(y~x1,data.frame(x1, y),data.frame(x1=100000),k=20,kernel = "rectangular")
cat("knn predicted value: ",near3$fitted,"\n")
```

```
## knn predicted value:  18251.55
```

2. kNN, Cars Data with Mileage and Year

Use kNN to get a prediction for a 2008 car with 75,000 miles on it! Remember to use cross-validation to choose k and scale your x's !!

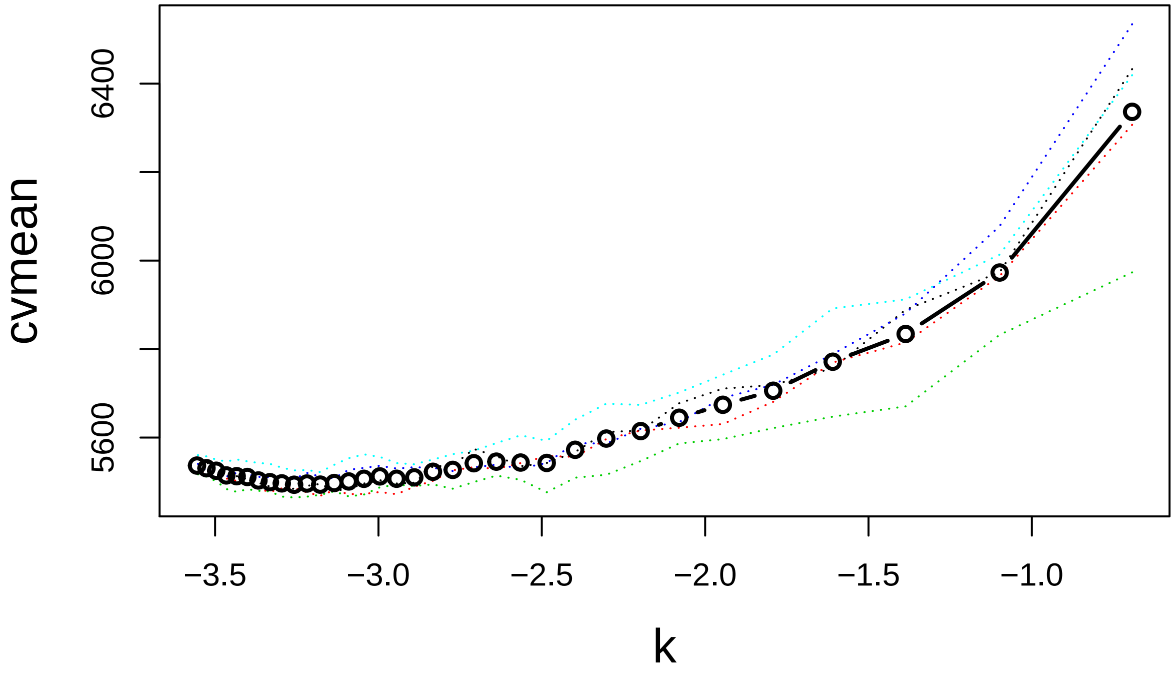
```
#get variables we want
x2 = cbind(carDat$mileage, carDat$year)
colnames(x2) = c("mileage","year")
mmsc=function(x) {return((x-min(x))/(max(x)-min(x)))}
xs = apply(x2,2,mmsc) #apply scaling function to each column of x
```



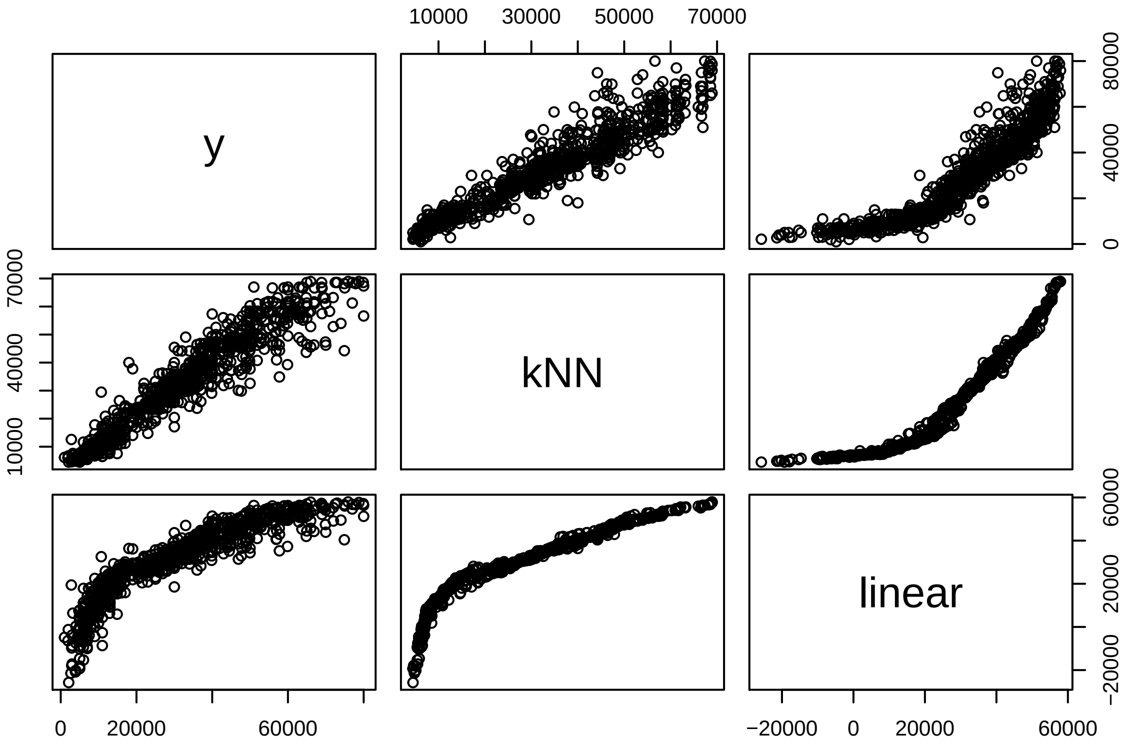
```
#run cross val several times
kv = 2:35 #these are the k values (k as in kNN) we will try
cvmean = rep(0,length(kv)) #will keep average rmse here
ndocv = 5 #number of CV splits to try
n=length(y)
cvmat = matrix(0,length(kv),ndocv) #keep results for each split
```

```
for(i in 1:ndocv) {
  cvtemp = docvknn(xs,y,kv,nfold=5)
  cvmean = cvmean + cvtemp
  cvmat[,i] = sqrt(cvtemp/n)
}
cvmean = cvmean/ndocv
cvmean = sqrt(cvmean/n)
```

```
plot(log(1/kv),cvmean,type="n",ylim=range(cvmat),xlab="k",cex.lab=1.5)

for(i in 1:ndocv) lines(log(1/kv),cvmat[,i],col=i,lty=3) #plot each result
lines(log(1/kv),cvmean,type="b",col="black",lwd=2) #plot average result
```



```
## the best k is:  26
```



```
##                y       kNN    linear
## y      1.0000000 0.9581329 0.9123897
## kNN    0.9581329 1.0000000 0.9525080
## linear 0.9123897 0.9525080 1.0000000
```

```
## knn predicted value:  31646.65
```

Is your predictive accuracy better using (mileage,year) than it was with just mileage?

Comparing the RMSEs we have:

```
kbest2 = kknn(y~x1,data.frame(x1, y),data.frame(x1), k=kbest2,kernel = "rectangular")
k = rmse(kbest2$fitted.values,y)
j = rmse(near2$fitted.values,y)
kable(cbind(Mileage = k, MileageYear = j))
```

Mileage	MileageYear
8810.022	5275.12

So yes, the predictive accuracy is quite better.

3. Choice of Kernel

In our notes examples we used kernel="rectangular" when calling the R function kknn.So, you can weight the y values at the neighbors equally, or weight the closer ones more heavily. Typically default is equal weights.Using the used cars data and predictors (features!!) (mileage,year) try a weighting option other than uniform.

We try some of the different weighting methods and report their RMSEs below.

Triangular	Triweight	Optimal	Normal
5062.027	4838.124	5062.027	5275.12

In this case, we see that tri-weight has the best accuracy among these options.