Homework#3

Neshma Simon

10/6/2020

Study Group: Fareha, Stan, and Hertz

k-nn classification: First we write down a code in order to do k-nn classification. This code will help

```
dat_NYC <- subset(acs2017_ny, (acs2017_ny$in_NYC == 1)&(acs2017_ny$AGE > 20) & (acs2017_ny$AGE < 66))
attach(dat_NYC)
borough_f <- factor((in_Bronx + 2*in_Manhattan + 3*in_StatenI + 4*in_Brooklyn + 5*in_Queens), levels=c(
```

(0,1) interval: In this data we picked the household income as a variable to classify by the boroughs.

```
norm_varb <- function(X_in) {
  (X_in - min(X_in, na.rm = TRUE))/( max(X_in, na.rm = TRUE) - min(X_in, na.rm = TRUE) )
}
```

data fix:

```
is.na(OWNCOST) <- which(OWNCOST == 9999999)
housing_cost <- OWNCOST + RENT
norm_inc_tot <- norm_varb(INCTOT)
norm_housing_cost <- norm_varb(housing_cost)
```

dataframe created:This the data frame we will use to associate the hosuehold incomes with the different

```
data_use_prelim <- data.frame(norm_inc_tot,norm_housing_cost)
good_obs_data_use <- complete.cases(data_use_prelim,borough_f)
dat_use <- subset(data_use_prelim,good_obs_data_use)
y_use <- subset(borough_f,good_obs_data_use)
```

80/20 split: one part to train the algo, then the other part to test how well it works for new data:

```
set.seed(12345)
NN_obs <- sum(good_obs_data_use == 1)
select1 <- (runif(NN_obs) < 0.8)
train_data <- subset(dat_use,select1)
test_data <- subset(dat_use,(!select1))
cl_data <- y_use[select1]
true_data <- y_use[!select1]
```

k-nn algo run and compare against the simple means:

```
summary(cl_data)
prop.table(summary(cl_data))
summary(train_data)
require(class)
```

```
for (indx in seq(1, 9, by= 2)) {
  pred_borough <- knn(train_data, test_data, cl_data, k = indx, l = 0, prob = FALSE, use.all = TRUE)
  num_correct_labels <- sum(pred_borough == true_data)
  correct_rate <- num_correct_labels/length(true_data)
  print(c(indx,correct_rate))
}
```

As we attempted to make it our own, we tried to modify the codes to see if there would be a signifcant change or shift.

```
Just as what was done previously, we tried to go a different route and see if this can work as a 60/40 s
set.seed(12345)
NN_obs <- sum(good_obs_data_use ==1)
select1 <-runif(NN_obs)<0.6
train_data <- subset(dat_use,select1)
test_data <-subset(dat_use,!select1)
cl_data <-y_use[select1]  ##matrix of classes of the K
true_data <-y_use[!select1]

Then, we can do a k-nn algo and compare against the simple means as we have done before.
summary(cl_data)
       Bronx     Manhattan Staten Island      Brooklyn        Queens
        3641          3943          1433          9344          8250
prop.table(summary(cl_data))
       Bronx     Manhattan Staten Island      Brooklyn        Queens
  0.13682312    0.14817181    0.05384991    0.35113299    0.31002217
summary(train_data)
  norm_inc_tot     norm_housing_cost
 Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.01184   1st Qu.:0.02476
 Median :0.02693   Median :0.96917
 Mean   :0.04268   Mean   :0.58874
 3rd Qu.:0.05219   3rd Qu.:0.97784
 Max.   :1.00000   Max.   :1.00000
 library(class)
 for (indx in seq(1, 9, by= 2)) {
  pred_borough <- knn(train_data, test_data, cl_data, k = indx, l = 0, prob = FALSE, use.all = TRUE)
  num_correct_labels <- sum(pred_borough == true_data)
  correct_rate <- num_correct_labels/length(true_data)
  print(c(indx,correct_rate))
}
[1] 1.0000000 0.3480951
[1] 3.0000000 0.3527356
[1] 5.0000000 0.3670581
[1] 7.0000000 0.3744486
[1] 9.0000000 0.3797193
```

```
summary(cl_data)
prop.table(summary(cl_data))
summary(train_data)
require(class)
for (indx in seq(1, 19, by= 2)) {
 pred_borough <- knn(train_data, test_data, cl_data, k = indx, l = 0, prob = FALSE, use.all = TRUE)
```

```
num_correct_labels <- sum(pred_borough == true_data)
correct_rate <- num_correct_labels/length(true_data)
print(c(indx,correct_rate))
}
[1] 1.0000000 0.3195239
[1] 3.0000000 0.3219162
[1] 5.0000000 0.3342864
[1] 7.0000000 0.3411717
[1] 9.0000000 0.3481736
[1] 11.0000000  0.3499825
[1] 13.0000000  0.3488738
[1] 15.0000000  0.3495157
[1] 17.0000000  0.3506827
[1] 19.0000000  0.3512078
summary(correct_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.3512  0.3512  0.3512  0.3512  0.3512  0.3512
mean(correct_rate)
[1] 0.3512078
```

```
We increased our K value to increase to 19 from 15 to see the correlateion. After viewing the results, w

We choose to look at the household incomes in the boroughs to see whether the code is a good determinant
```