NRSG 741 Homework 8.2

Tommy Flynn

4/11/2018

The GitHub Repository can be found here https://github.com/tommyflynn/N741_Homework/tree/master/Flynn_HW_08

K-nearest neighbor Let's try a variation on the NHANES data set again.

```r
# Create the NHANES dataset with SleepTrouble, change Yes/No to 1/0 numeric,
#and filter out NA
lut <- c("Yes" = "1", "No" = "0")
people <- NHANES %>% dplyr::select(Age, Gender, BMI, HHIncome, PhysActive,
                                   SleepTrouble) %>%
  mutate(Gender = as.numeric(Gender),HHIncome = as.numeric(HHIncome),
         PhysActive = as.numeric(PhysActive), SleepTrouble = as.numeric(lut[SleepTrouble])) %>%
  filter(!is.na(Age), !is.na(Gender), !is.na(BMI), !is.na(HHIncome),
         !is.na(PhysActive), !is.na(SleepTrouble))

#check the subset
knitr::kable(summary(people), caption = "Summary of People Data Subset with SleepTrouble",
             format = "markdown")
```

	Age	Gender	BMI	HHIncome	PhysActive	SleepTrouble
	Min. :16.0	Min. :1.000	Min. :15.02	Min. : 1.000	Min. :1.000	Min. :0.0000
	1st Qu.:30.0	1st Qu.:1.000	1st Qu.:23.90	1st Qu.: 6.000	1st Qu.:1.000	1st Qu.:0.0000
	Median :44.0	Median :1.000	Median :27.50	Median : 8.000	Median :2.000	Median :1.0000
	Mean :45.1	Mean :1.496	Mean :28.63	Mean : 8.221	Mean :1.549	Mean :0.7445
	3rd Qu.:58.0	3rd Qu.:2.000	3rd Qu.:32.01	3rd Qu.:11.000	3rd Qu.:2.000	3rd Qu.:1.0000
	Max. :80.0	Max. :2.000	Max. :81.25	Max. :12.000	Max. :2.000	Max. :1.0000

Create the NHANES dataset again, just like we did in class, only using sleep trouble (variable name = SleepTrouble) as the dependent variable, instead of SleepTrouble. (I'm assuming you meann instead of Diabetes?)

Problem 1

What is the marginal distribution of sleep trouble?

```r
# What is the marginal distribution of sleep trouble?
knitr::kable(tally(~ SleepTrouble, data = people, format = "percent"),
             caption = "Marginal Distribution of SleepTrouble", format = "markdown")
```

SleepTrouble	Freq
0	25.55066
1	74.44934

Problem 2

Apply the k-nearest neighbor procedure to predict SleepTrouble from the other covariates, as we did for SleepTrouble. Use k = 1, 3, 5, and 20.

```
# Apply knn procedure to predict SleepTrouble

# Let's try different values of k to see how that affects performance
knn.1 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 1)
knn.3 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 3)
knn.5 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 5)
knn.20 <- knn(train = people, test = people, cl = people$SleepTrouble, k = 20)
```

Now let's see how well these classifiers work overall

Problem 3

```
# Calculate the percent predicted correctly
100*sum(people$SleepTrouble == knn.1)/length(knn.1)
```

```
## [1] 100
```

```
100*sum(people$SleepTrouble == knn.3)/length(knn.3)
```

```
## [1] 92.24101
```

```
100*sum(people$SleepTrouble == knn.5)/length(knn.5)
```

```
## [1] 88.6031
```

```
100*sum(people$SleepTrouble == knn.20)/length(knn.20)
```

```
## [1] 78.6841
```

Problem 4

What about success overall?

```
# Another way to look at success rate against increasing k
table(knn.1, people$SleepTrouble)
```

```
##
## knn.1    0    1
##     0 1798    0
##     1    0 5239
```

```
table(knn.3, people$SleepTrouble)
```

```
##
## knn.3    0    1
##     0 1430  178
##     1  368 5061
```

```
table(knn.5, people$SleepTrouble)
```

```
##
## knn.5    0    1
##     0 1210  214
##     1  588 5025
```

```r
table(knn.20, people$SleepTrouble)
```

```
##
## knn.20    0    1
##      0  442  144
##      1 1356 5095
```