# Practice 8

Smit Patil

7/20/2020

—Problem 1—

Build an R Notebook of the social networking service example in the textbook on pages 296 to 310. Show each step and add appropriate documentation.

```
#Importing social network service data
teens <- read.csv("snsdata.csv")

#Looking at the head and the structure of the data
head(teens)
```

```
##   gradyear gender    age friends basketball football soccer softball volleyball
## 1     2006      M 18.982       7          0        0      0        0          0
## 2     2006      F 18.801       0          0        1      0        0          0
## 3     2006      M 18.335      69          0        1      0        0          0
## 4     2006      F 18.875       0          0        0      0        0          0
## 5     2006   <NA> 18.995      10          0        0      0        0          0
## 6     2006      F     NA     142          0        0      0        0          0
##   swimming cheerleading baseball tennis sports cute sex sexy hot kissed dance
## 1        0            0        0      0      0    0   0    0   0      0     1
## 2        0            0        0      0      0    1   0    0   0      0     0
## 3        0            0        0      0      0    0   0    0   0      0     0
## 4        0            0        0      0      0    1   0    0   0      0     0
## 5        0            0        0      0      0    0   1    0   0      5     1
## 6        0            0        0      0      0    0   1    0   0      0     0
##   band marching music rock god church jesus bible hair dress blonde mall
## 1    0        0     0    0   0      0     0     0    0     0      0    0
## 2    0        0     2    2   1      0     0     0    6     4      0    1
## 3    2        0     1    0   0      0     0     0    0     0      0    0
## 4    0        0     0    1   0      0     0     0    0     0      0    0
## 5    1        0     3    0   1      0     0     0    1     0      0    0
## 6    0        1     2    0   0      0     0     0    0     1      0    0
##   shopping clothes hollister abercrombie die death drunk drugs
## 1        0       0         0           0   0     0     0     0
## 2        0       0         0           0   0     0     0     0
## 3        0       0         0           0   0     1     0     0
## 4        0       0         0           0   0     0     0     0
## 5        2       0         0           0   0     0     1     1
## 6        1       0         0           0   0     0     1     0
```

```r
str(teens)
```

```
## 'data.frame':    30000 obs. of  40 variables:
##  $ gradyear    : int  2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
##  $ gender      : chr  "M" "F" "M" "F" ...
##  $ age         : num  19 18.8 18.3 18.9 19 ...
##  $ friends     : int  7 0 69 0 10 142 72 17 52 39 ...
##  $ basketball  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ football    : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ soccer      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ softball    : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ volleyball  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ swimming    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cheerleading: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ baseball    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ tennis      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sports      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cute        : int  0 1 0 1 0 0 0 0 0 1 ...
##  $ sex         : int  0 0 0 0 1 1 0 2 0 0 ...
##  $ sexy        : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ hot         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ kissed      : int  0 0 0 0 5 0 0 0 0 0 ...
##  $ dance       : int  1 0 0 0 1 0 0 0 0 0 ...
##  $ band        : int  0 0 2 0 1 0 1 0 0 0 ...
##  $ marching    : int  0 0 0 0 0 1 1 0 0 0 ...
##  $ music       : int  0 2 1 0 3 2 0 1 0 1 ...
##  $ rock        : int  0 2 0 1 0 0 0 1 0 1 ...
##  $ god         : int  0 1 0 0 1 0 0 0 0 6 ...
##  $ church      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ jesus       : int  0 0 0 0 0 0 0 0 0 2 ...
##  $ bible       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hair        : int  0 6 0 0 1 0 0 0 0 1 ...
##  $ dress       : int  0 4 0 0 0 1 0 0 0 0 ...
##  $ blonde      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mall        : int  0 1 0 0 0 0 2 0 0 0 ...
##  $ shopping    : int  0 0 0 0 2 1 0 0 0 1 ...
##  $ clothes     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hollister   : int  0 0 0 0 0 0 2 0 0 0 ...
##  $ abercrombie : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ die         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ death       : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ drunk       : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ drugs       : int  0 0 0 0 1 0 0 0 0 0 ...
```

```r
#Creating a gender table for the teens
table(teens$gender)
```

```
## 
##     F     M 
## 22054  5222
```

```r
#Creating a gender table along with the count of NA for the teens
table(teens$gender, useNA = "ifany")
```

```
##
##     F     M  <NA>
## 22054  5222  2724
```

```r
#Summarising the age values of the teens
summary(teens$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   3.086  16.312  17.287  17.994  18.259 106.927    5086
```

```r
#Setting age values to NA for ages below 13 and above 20
teens$age <- ifelse(teens$age >= 13 & teens$age < 20, teens$age, NA)

#Summarising the new age values
summary(teens$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   13.03   16.30   17.27   17.25   18.22   20.00    5523
```

```r
#Assigining the value 1 if gender is equal to F and the gender is not equal to NA,
#otherwise it assigns the value 0
teens$female <- ifelse(teens$gender == "F" & !is.na(teens$gender), 1, 0)

#Assigning the value 0 in no_gender and if gender is NA else 1
teens$no_gender <- ifelse(is.na(teens$gender), 1, 0)

#Creating a table for gender along with the NA
table(teens$gender, useNA = "ifany")
```

```
##
##     F     M  <NA>
## 22054  5222  2724
```

```r
#Creating a table for female along with the NA
table(teens$female, useNA = "ifany")
```

```
##
##     0     1
##  7946 22054
```

```r
#Creating a table for no_gender along with the NA
table(teens$no_gender, useNA = "ifany")
```

```
##
##     0     1
## 27276  2724
```

```r
#Calculating the mean age of the teens
mean(teens$age)
```

```
## [1] NA
```

```r
#Calculating the mean age of the teens with NA removed
mean(teens$age, na.rm = TRUE)
```

```
## [1] 17.25243
```

```r
#calculating the mean age for levels of gradyear after removing the NA values
aggregate(data = teens, age ~ gradyear, mean, na.rm = TRUE)
```

```
##   gradyear      age
## 1     2006 18.65586
## 2     2007 17.70617
## 3     2008 16.76770
## 4     2009 15.81957
```

```r
#Calculatin the average age value
ave_age <- ave(teens$age, teens$gradyear, FUN = function(x) mean(x, na.rm = TRUE))

#replacing the NA values in age with their average age
teens$age <- ifelse(is.na(teens$age), ave_age, teens$age)

#Summarising the age values of the teens
summary(teens$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.03   16.28   17.24   17.24   18.21   20.00
```

```r
#Creating a new data frame for the 36 features of interest
interests <- teens[5:40]

#Applying z-score standardization to the interests data frame
interests_z <- as.data.frame(lapply(interests, scale))

#Dividing the teens into 5 clusters using kmeans
teen_clusters <- kmeans(interests_z, 5)

#Printing the teens cluster sizes
teen_clusters$size
```

```
## [1]   803 24603   942   748  2904
```

```r
#Looking at the teens cluster along with their centers
teen_clusters$centers
```

```
##    basketball    football       soccer    softball  volleyball    swimming
## 1 -0.10936370  0.02189808 -0.141041524  0.01272154 -0.08549971  0.04354354
## 2 -0.07365317 -0.07406622 -0.144848674 -0.03659560 -0.03947468 -0.04895182
## 3  0.13874300  0.22926380  0.004807491  0.08488702  0.21593034  0.25953585
## 4  0.27584113  0.24313487  4.969313228  0.03329546  0.13774687  0.14602413
## 5  0.53818286  0.48444752 -0.015360429  0.27041241  0.25255193  0.28088409
##   cheerleading    baseball       tennis      sports        cute         sex
## 1  -0.10320973 -0.10087569  0.006916883 -0.11999981 -0.01902962 -0.04671851
## 2  -0.04816609 -0.03970659 -0.023796453 -0.07805010 -0.11135838 -0.09944693
## 3   0.49236470  0.02879711  0.165781933  0.09273394  0.38081723  0.03755427
## 4  -0.03317141  0.04999311  0.105187660  0.43223243  0.03414669 -0.04120894
## 5   0.28543817  0.34207391  0.118823319  0.55301693  0.81637726  0.85387601
##          sexy         hot      kissed       dance        band    marching
## 1 -0.04098435 -0.06669152 -0.04613613  0.02312810  3.38765823  4.63879117
## 2 -0.06486409 -0.07021642 -0.13997255 -0.08954979 -0.12715702 -0.13518935
## 3  0.12257411  0.41816952  0.09125873  0.23055674 -0.09889323 -0.11183725
## 4 -0.01168715  0.10687259 -0.01363204 -0.01842752 -0.04552873 -0.09485162
## 5  0.52411802  0.45014855  1.17252859  0.68223862  0.18435522 -0.07664811
##         music        rock           god       church        jesus       bible
## 1  0.38335388  0.14452723  8.470537e-02  0.053338583  0.062105946  0.02963143
## 2 -0.11264824 -0.10532587 -7.098219e-02 -0.075812622 -0.044031953 -0.04426398
## 3  0.11948593  0.04093766  2.820427e-02 -0.007348012  0.004440388 -0.04718437
## 4  0.05241487  0.10946969 -4.419084e-05  0.116007002  0.011891319  0.03294256
## 5  0.79610517  0.81089204  5.688089e-01  0.600046712  0.351366909  0.37363616
##          hair        dress      blonde        mall    shopping     clothes
## 1 -0.04686731  0.066206522 -0.01438977 -0.09605527 -0.05064419 -0.03226492
## 2 -0.18444446 -0.087010100 -0.02544750 -0.10965198 -0.10547860 -0.14280134
## 3  0.45050080  0.161725059  0.06383945  0.68440364  0.94738678  0.59545601
## 4  0.03156095  0.002965488  0.03097203  0.05090003  0.21215692 -0.03133497
## 5  1.42132996  0.665627573  0.19088706  0.72042648  0.54566998  1.03366703
##      hollister abercrombie          die       death       drunk       drugs
## 1 -0.169152744 -0.14739125 -0.019606317  0.02358006 -0.08623239 -0.08119538
## 2 -0.144040818 -0.14493568 -0.098172184 -0.07458093 -0.10053744 -0.12079255
## 3  3.848532290  3.90292482  0.038378427  0.09330691  0.05621484  0.05552332
## 4 -0.008713743 -0.04434402  0.009190433 -0.01667018 -0.03282272 -0.02013593
## 5  0.020959487  0.01405715  0.822330302  0.59936431  0.86582789  1.03299509
```

```r
#Adding cluster column from the teen_clusters data to the teens data
teens$cluster <- teen_clusters$cluster

#Looking at the first 5 rows of the teens data and their cluster
teens[1:5, c("cluster", "gender", "age", "friends")]
```

```
##   cluster gender    age friends
## 1       2      M 18.982       7
## 2       5      F 18.801       0
## 3       2      M 18.335      69
## 4       2      F 18.875       0
## 5       5   <NA> 18.995      10
```

```r
#calculating the mean age for each cluster using the aggregate function
aggregate(data = teens, age ~ cluster, mean)
```

```
##   cluster      age
```

```
## 1       1 17.38627
## 2       2 17.27217
## 3       3 16.88461
## 4       4 16.99488
## 5       5 17.07778
```

```
#calculating the female percentage for each cluster using the aggregate function
aggregate(data = teens, female ~ cluster, mean)
```

```
##   cluster    female
## 1       1 0.7334994
## 2       2 0.7175141
## 3       3 0.8428875
## 4       4 0.7620321
## 5       5 0.8429752
```

```
#calculating the mean friends for each cluster using the aggregate function
aggregate(data = teens, friends ~ cluster, mean)
```

```
##   cluster  friends
## 1       1 32.86052
## 2       2 28.70707
## 3       3 42.08493
## 4       4 36.37701
## 5       5 36.45420
```

—Problem 2—

Provide 100-300 word answers to each of the following interview questions:

```
1. What are some of the key differences between SVM and Random Forest for classification?
When is each algorithm appropriate and preferable? Provide examples.
-> Random Forests are suitable for multiclass problems, while SVMs are suitable for
two-class problems. To implement multiclass in SVM, we need to convert it into multiple
binary classification problems. Random forests can manage incredibly large data sets,
while SVMs can be sluggish to train if there are many features or examples in the input
dataset. Random Forest gives us the probability of a class, while SVM provides us with
the distance and will require further probability calculations. Random Forest performs
well on certain features and spaces of high dimensions with many training data sets.
In comparison, SVM works well on linear and nonlinear dependencies, and SVM will be the
best choice to use with any nonlinear data collection.

2. Why might it be preferable to include fewer predictors over many?
-> There is an excellent likelihood for many predictors that there is a correlation between
most of them. However, some of the predictors are unlikely to have a significant effect on
the dependent variables, making them irrelevant. Selecting a limited amount of features also
decreases the probability of overfitting a pattern. Although all of the predictor variables
may be important, running the model may require a lot of computing power. Looking at the
future implementation of the model, we want to apply models not only to the same collection
but also to the general population from which the training data came. Therefore it is always
easier to understand and enforce simpler models that suit data well.
```

3. You are asked to provide R-Squared for a kNN regression model. How would you respond to that request?
-> R-squared (R2) is a statistical measure that reflects the proportion of the variance for a dependent variable, which is explained in a regression model by an independent variable or variables. In simpler terms, it is a measure of goodness of fit of a linear model. In contrast, kNN (K-nearest neighbors) is a classification algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). Unlike the regression algorithms, classification algorithms have different evaluation metrics such as 'accuracy,' 'true-negative,' 'false-positive,' etc. Hence asking about the accuracy of the kNN model would be a more suitable point.

4. How can you determine which features to include when building a multiple regression model?
-> Feature selection is used to minimize the number of features when constructing a multiple regression model. The selection process aims to reduce the collection of predictor variables to those needed and account for almost as much variance as the total collection accounts for. Essentially, selection helps assess the degree of significance of each predictor variable. It also assists in determining results after statistically removing the other predictor variables. Four selection procedures are used to yield the most appropriate regression equation: forward selection, backward elimination, stepwise selection, and block-wise selection. The first three of those four approaches are called methods of statistical regression. Researchers also use sequential regression methods (hierarchical or block-wise), which do not rely on statistical results to pick predictors.