Practice-8

Harsh

20/07/2020

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 Dataset
sns_data <- read.csv("C:\\Users\\harsh\\Desktop\\Introduction to Machine learning and Data Mining\\Prac

#Exploring Dataset
str(sns_data)
```

```
## 'data.frame':    30000 obs. of  40 variables:
##  $ gradyear    : int  2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
##  $ gender      : Factor w/ 2 levels "F","M": 2 1 2 1 NA 1 1 2 1 1 ...
##  $ 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
#Checking the distribution of gender with NA present or not
table(sns_data$gender, useNA = "ifany")
```

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

```r
#Exploring age feature we observe that it contains age from 3 to 107
summary(sns_data$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
#Since we are working with teen data we remove all the ages above 20 and below 13
sns_data$age <- ifelse(sns_data$age >= 13 & sns_data$age < 20, sns_data$age, NA)
summary(sns_data$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
#Assigning dummy codes to gender
sns_data$female <- ifelse(sns_data$gender == "F" & !is.na(sns_data$gender), 1, 0)
sns_data$no_gender <- ifelse(is.na(sns_data$gender), 1, 0)

#Check count of dummy codes for gender and comparing with original
table(sns_data$gender, useNA = "ifany")
```

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

```r
table(sns_data$female, useNA = "ifany")
```

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

```r
table(sns_data$no_gender, useNA = "ifany")
```

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

```r
#Calculating mean of age with and without NA's
mean(sns_data$age)
```

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

```r
mean(sns_data$age, na.rm = TRUE)
```

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

```r
#Computing mean of age by grouping with graduation year
aggregate(data = sns_data, 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
#use the ave() function, which returns a vector with the group means repeated such that the result is e
ave_age <- ave(sns_data$age, sns_data$gradyear, FUN = function(x) mean(x, na.rm = TRUE))

#Imputing missing age values
sns_data$age <- ifelse(is.na(sns_data$age), ave_age, sns_data$age)
summary(sns_data$age)
```

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

```r
#Selecting interest features
interests <- sns_data[5:40]

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

#Using kmeans to divide interests in 5 clusters
teen_clusters <- kmeans(interests_z, 5)

#Checking the size of the clusters and centers of the cluster
teen_clusters$size
```

```
## [1]  4532   799   992 21104  2573
```

```r
teen_clusters$centers
```

```
##     basketball      football      soccer    softball  volleyball    swimming
## 1  0.691113262  6.727539e-01  0.38067647  0.47591721  0.52420846  0.27396470
## 2 -0.120690492  2.912151e-02 -0.07912876 -0.01657931 -0.07894078  0.05474645
## 3  0.368052565  3.868988e-01  0.16377343  0.15275451  0.11501116  0.27039883
## 4 -0.160682729 -1.637488e-01 -0.08323897 -0.10735951 -0.11015531 -0.08701455
## 5 -0.003790687 -9.251287e-05 -0.02634663 -0.01143697 -0.03964725  0.10989856
##   cheerleading     baseball      tennis      sports        cute         sex
## 1    0.4983954  0.52123566  0.14905047  0.42515136  0.48252147 -0.01885430
## 2   -0.1172888 -0.10996937  0.01263926 -0.11911311 -0.03426725 -0.04601933
## 3    0.2336643  0.29544097  0.10796143  0.88410590  0.52852362  2.10968354
## 4   -0.1118792 -0.11302959 -0.04244594 -0.12632236 -0.16107607 -0.09646297
## 5   -0.0138793 -0.07076385  0.04006445 -0.01661117  0.27813691  0.02532670
##           sexy         hot      kissed       dance        band    marching
## 1  0.182167844  0.47122056 -0.05662232  0.48495895 -0.05182835 -0.12374132
## 2 -0.006678998 -0.06831472 -0.05518171  0.03691674  3.40260230  4.63656528
## 3  0.555226382  0.37746322  3.01748806  0.48057525  0.39987621 -0.01501156
## 4 -0.072114550 -0.12942860 -0.13507524 -0.14264205 -0.12921000 -0.13531188
## 5  0.058636897  0.10727924  0.06140006  0.11902712 -0.05970564 -0.10622102
##        music        rock         god      church       jesus       bible
## 1  0.2116275  0.20651379  0.37145960  0.58053318  0.33156763  0.30979226
## 2  0.3752308  0.14172968  0.05332025  0.02808396  0.04616909  0.03030165
## 3  1.2509139  1.27147862  0.43784696  0.16174254  0.10714716  0.06816154
## 4 -0.1363694 -0.11256534 -0.10606570 -0.13932521 -0.07600400 -0.06534609
## 5  0.1469600  0.02530495  0.03031813  0.04914697 -0.01626706 -0.04537192
##          hair       dress       blonde        mall    shopping     clothes
## 1  0.20095240  0.36090065  0.044377174  0.49842308  0.64552048 -0.13632186
## 2 -0.05885403  0.06220454 -0.014851174 -0.09468429 -0.06746337 -0.06527789
## 3  2.62567228  0.54261740  0.377758653  0.67472910  0.30451623  1.06788645
## 4 -0.19898959 -0.12270447 -0.027271767 -0.16920391 -0.20648920 -0.31419265
## 5  0.28414980  0.14223691  0.004491073  0.27918668  0.46019138  2.42570773
##     hollister abercrombie         die       death       drunk       drugs
## 1  0.60407327  0.57312079  0.04192437  0.11532602 -0.01142446 -0.06506395
## 2 -0.16538183 -0.15168982 -0.02824896  0.02214204 -0.08869649 -0.08434850
## 3  0.43805227  0.51292075  1.76164302  0.95933372  1.89360211  2.85639424
## 4 -0.15214542 -0.14472191 -0.09175009 -0.07602191 -0.08541347 -0.11278084
## 5  0.06638517  0.02689875  0.00828357  0.04366822  0.01817207 -0.03542634
```

```r
#Adding a new column cluster to the dataset
sns_data$cluster <- teen_clusters$cluster

#Getting the data for first 5 users
sns_data[1:5, c("cluster", "gender", "age", "friends")]
```

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

```r
#check average age for each cluster
aggregate(data = sns_data, age ~ cluster, mean)
```

```
##   cluster      age
## 1       1 17.03157
## 2       2 17.36853
## 3       3 17.08832
## 4       4 17.29666
## 5       5 17.12975
```

```r
#check average gender for each cluster
aggregate(data = sns_data, female ~ cluster, mean)
```

```
##   cluster    female
## 1       1 0.8232568
## 2       2 0.7321652
## 3       3 0.8014113
## 4       4 0.7008150
## 5       5 0.8367664
```

```r
#check average number of friends for each cluster
aggregate(data = sns_data, friends ~ cluster, mean)
```

```
##   cluster  friends
## 1       1 39.01743
## 2       2 32.78348
## 3       3 30.59476
## 4       4 27.88580
## 5       5 32.45667
```

Problem 2:

1. What are some of the key differences between SVM and Random Forest for classification? When is each algorithm appropriate and preferable? Provide examples.

- SVM models perform better on sparse data than random forest trees. Also SVM generally perform better on linear dependencies
- SVM are less interpretable compared to Random forest
- Random forest tend to overfit the model whereas SVM does not
- Random forest is used for multiclass classification where as SVM is used for binary classification
- An example of SVM is Handwriting recognition and classificaiton of genes of a patient based on gene and proteins.
- An example of Random forest is Credit score decision making where applicant is rejected or not

2. Why might it be preferable to include fewer predictors over many?

- Usually if we select many predictors it only makes the model overfitted
- Also adding many features sometime increase computation time and causes decrease in performance
- Because of this it is necessary to remove irrelevant predictors and it is recommended to use fewer and important features

- Getting too many features means getting more data. It is not always possible to get all the data, so missing or sparse data can impact the model's performance.

3. You are asked to provide R-Squared for a kNN regression model. How would you respond to that request?

- R-squared is a measure of goodness of a linear model. Since kNN is a non-linear regression model it would make no sense calculating R-squared for that.
- Because of this it is recommended to use different measures to calculate the accuracy of the model

4. How can you determine which features to include when building a multiple regression model?

- To decide which feature is to be included in the multiple regression model we can make use of different selection/elimination methods
- For eg. Backward elimination, Stepwise elimination, Forward selection.
- In Backward elimination, we select all features and then eliminate a single feature based on the p-value or AIC value which is not significant. After eliminating all insignificant features, we are left with most significant features which are included in the model
- In Forward selection, the reverse takes place we start with an empty equation and try every features each time and select the most significant one
- In Stepwise selection requires an analysis of the contribution of the predictor variable previously entered in the equation at each step