R Notebook

Problem 1 (50 Points)

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.

Step 1 - Exploring and preparing the data

```
teens <- read.csv("snsdata.csv")
str(teens)
```

```
## '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
table(teens$gender)
```

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

```r
# Since the previous command doesn't show the no. of NA, specify that the table show the NA if they are
table(teens$gender, useNA = "ifany")
```

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

```r
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
# If the age is not between the range of 13 - 19, then put NA in its place
teens$age <- ifelse(teens$age >=13 & teens$age < 20, teens$age, NA)
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
```

Data preparation – dummy coding missing values

```r
teens$female <- ifelse(teens$gender == "F" & !is.na(teens$gender),1,0)
teens$no_gender <- ifelse(is.na(teens$gender),1,0)
str(teens)
```

```
## 'data.frame':    30000 obs. of  42 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 ...
##  $ female      : num  0 1 0 1 0 1 1 0 1 1 ...
##  $ no_gender   : num  0 0 0 0 1 0 0 0 0 0 ...
```

```
table(teens$gender, useNA = "ifany")
```

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

```
table(teens$female, useNA = "ifany")
```

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

```
table(teens$no_gender, useNA = "ifany")
```

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

Data preparation – imputing the missing values

```
mean(teens$age)
```

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

```
mean(teens$age, na.rm = T)
```

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

```
# Calculate the mean age by graduation year after removing the NA values
aggregate(data = teens, age ~ gradyear, mean, na.rm = T)
```

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

```
ave_age <- ave(teens$age, teens$gradyear, FUN = function(x) mean(x, na.rm = T))
table(ave_age)
```

```
## ave_age
## 15.8195733445096 16.7677007371007 17.7061723749799 18.6558579508727
```

```
##             7500             7500             7500             7500
```

```
teens$age <- ifelse(is.na(teens$age), ave_age, teens$age)
summary(teens$age)
```

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

Step 2 – training a model on the data

```
library(stats)
# Only consider the 36 features that represent the interest of teens
interests <- teens[5:40]
# Apply z score standardization
interests_z <- as.data.frame(lapply(interests, scale))
# Use k-means algorithm to divide the teenager's interest data into 5 clusters
set.seed(2345)
teen_clusters <- kmeans(interests_z,5)
```

Step 3 - evaluating model performance

```
# Obtain the size of the kmeans clusters
teen_clusters$size
```

```
## [1]   871   600  5981  1034 21514
```

```
# Examine the coordinates of the cluster centroids
teen_clusters$centers
```

```
##    basketball   football      soccer    softball  volleyball    swimming
## 1  0.16001227  0.2364174  0.10385512  0.07232021  0.18897158  0.23970234
## 2 -0.09195886  0.0652625 -0.09932124 -0.01739428 -0.06219308  0.03339844
## 3  0.52755083  0.4873480  0.29778605  0.37178877  0.37986175  0.29628671
## 4  0.34081039  0.3593965  0.12722250  0.16384661  0.11032200  0.26943332
## 5 -0.16695523 -0.1641499 -0.09033520 -0.11367669 -0.11682181 -0.10595448
##   cheerleading    baseball      tennis      sports        cute
## 1    0.3931445  0.02993479  0.13532387  0.10257837  0.37884271
## 2   -0.1101103 -0.11487510  0.04062204 -0.09899231 -0.03265037
## 3    0.3303485  0.35231971  0.14057808  0.32967130  0.54442929
## 4    0.1856664  0.27527088  0.10980958  0.79711920  0.47866008
## 5   -0.1136077 -0.10918483 -0.05097057 -0.13135334 -0.18878627
##            sex        sexy         hot      kissed       dance        band
## 1  0.020042068  0.11740551  0.41389104  0.06787768  0.22780899 -0.10257102
## 2 -0.042486141 -0.04329091 -0.03812345 -0.04554933  0.04573186  4.06726666
## 3  0.002913623  0.24040196  0.38551819 -0.03356121  0.45662534 -0.02120728
## 4  2.028471066  0.51266080  0.31708549  2.97973077  0.45535061  0.38053621
## 5 -0.097928345 -0.09501817 -0.13810894 -0.13535855 -0.15932739 -0.12167214
##      marching      music        rock         god      church       jesus
## 1 -0.10942590  0.1378306  0.05905951  0.03651755 -0.00709374  0.01458533
## 2  5.25757242  0.4981238  0.15963917  0.09283620  0.06414651  0.04801941
## 3 -0.10880541  0.2844999  0.21436936  0.35014919  0.53739806  0.27843424
## 4 -0.02014608  1.1367885  1.21013948  0.41679142  0.16627797  0.12988313
## 5 -0.11098063 -0.1532006 -0.12460034 -0.12144246 -0.15889274 -0.08557822
##         bible        hair       dress      blonde        mall    shopping
## 1 -0.03692278  0.43807926  0.14905267  0.06137340  0.60368108  0.79806891
## 2  0.05863810 -0.04484083  0.07201611 -0.01146396 -0.08724304 -0.03865318
## 3  0.22990963  0.23612853  0.39407628  0.03471458  0.48318495  0.66327838
## 4  0.08478769  2.55623737  0.53852195  0.36134138  0.62256686  0.27101815
```

```
## 5 -0.06813159 -0.20498730 -0.14348036 -0.02918252 -0.18625656 -0.22865236
##         clothes  hollister abercrombie          die       death
## 1  0.5651537331  4.1521844  3.96493810  0.043475966  0.09857501
## 2 -0.0003526292 -0.1678300 -0.14129577  0.009447317  0.05135888
## 3  0.3759725120 -0.0553846 -0.07417839  0.037989066  0.11972190
## 4  1.2306917174  0.1610784  0.26324494  1.712181870  0.93631312
## 5 -0.1865419798 -0.1557662 -0.14861104 -0.094875180 -0.08370729
##          drunk       drugs
## 1  0.035614771  0.03443294
## 2 -0.086773220 -0.06878491
## 3 -0.009688746 -0.05973769
## 4  1.897388200  2.73326605
## 5 -0.087520105 -0.11423381
```

Step 5 – improving model performance

```
# Add the clusters as a column on the teens data frame
teens$cluster <- teen_clusters$cluster
# Examine how the cluster assignment relates to the individual characteristics
teens[1:5, c("cluster", "gender", "age", "friends")]
```

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

```
# Look at the demographic characteristics of the clusters
aggregate(data = teens, age ~ cluster, mean)
```

```
##   cluster      age
## 1       1 16.86497
## 2       2 17.39037
## 3       3 17.07656
## 4       4 17.11957
## 5       5 17.29849
```

```
aggregate(data = teens, female ~ cluster, mean)
```

```
##   cluster    female
## 1       1 0.8381171
## 2       2 0.7250000
## 3       3 0.8378198
## 4       4 0.8027079
## 5       5 0.6994515
```

```
aggregate(data = teens, friends ~ cluster, mean)
```

```
##   cluster  friends
## 1       1 41.43054
## 2       2 32.57333
## 3       3 37.16185
## 4       4 30.50290
## 5       5 27.70052
```

Problem 2 (50 Points)

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

1. (10 Points) What are various ways to predict a binary response variable? Can you compare two of them and tell me when one would be more appropriate? What's the difference between these? (SVM, Logistic Regression, Naive Bayes, Decision Tree, etc.)

A binary response variable can be predicted by using the following algorithms: knn Naive Bayes Logistic Regression Decision Tree Random Forest SVM Neural Network

Comparing knn and decision tree:

Decision Tree comes under the category of "Eager Learners", because it first builds a classification model on the training dataset before being able to actually classify an unseen observation from test dataset. The KNN-based classifier comes under the category of "Lazy Learner". It does not build any classification model. It directly learns from the training instances (observations). It starts processing data only after it is given a test observation to classify. Since KNN performs instance-based learning, a well-tuned K can model complex decision spaces having arbitrarily complicated decision boundaries which is not easily modeled by decision trees. Decision tree excludes unimportant features so it would work better on a dataset which has a large number of features than knn since it cannot exclude any features.

Difference between the following:

SVM: It creates a flat boundary called a hyperplane between points of data plotted in multidimensional space to create fairly homogeneous partitions that represent examples and their feature values.

Logistic Regression: It explains the relationship between one dependent binary variable and one or more independent variables. The logistic regression works by estimating probabilities using a logistic function.

Naive Bayes: It is based on Bayesian methods to determine empirical probabilities of each outcome based on frequencies of feature values. When the classifier is then applied to unlabeled cases, it uses the empirical probabilities to predict the most likely class for the new case.

Decision Tree: It utilizes a tree structure to model the relationships among the features and the potential outcomes. It splits the data into subsets, which are then split repeatedly into even smaller subsets, and so on and so forth until the process stops when the algorithm determines the data within the subsets are sufficiently homogenous.

2. (10 Points) Why might it be preferable to include fewer predictors over many?

There are many reasons why it might be preferable to include fewer predictors over many:

1. Redundancy: When we use a lot of predictors, there is a high chance that there are hidden relationships between some of them which leads to redundany. If the redundant features are not identified early on, it can be a huge drag on the succeeding steps of the data analysis.
2. Irrelevance: It is quite likely that all the predictors do not have a considerable impact on the dependent variable. So, it is important to remove the irrelevant features before data modeling.
3. Overfitting: Even when a large number of predictor variables do not have any relationships between them, it is still preferred to work with fewer predictors. When the data has a large number of predictors, the models often suffer from the problem of overfitting. In overfitting, the model works really well on the training data but performs poorly on the testing data. Therefore, it is a good idea to work with fewer predictors (shortlisted through feature selection or developed through feature extraction). Focusing on those 20% most significant predictor variables will be of great help in building data models with considerable success rate in a reasonable time, without needing non-practical amount of data or other resources.
4. Understandability: Models with fewer predictors are easier to understand and explain. As the data science steps are performed by humans and the results will be presented used by humans, it is important to consider the comprehensive ability of human brain. This is basically a trade-off – letting go of some

potential benefits to the data model's success rate, while simultaneously making it easier to understand and optimize.

5. (10 Points) Given a database of all previous alumni donations to your university, how would you predict which recent alumni are most likely to donate?

Provided that the database of all previous alumni donations have enough demographic data on its alumni so as to represent indicative predictors such as age, major, address, salary etc, I would consider logistic regression to predict which recent alumni are most likely to donate.

Once we have good predictors, logistic regression can be used since it can effectively measure the relationship between the categorical dependent variable, which would be likely to donate or not, and one or more independent variables.

The logistic regression works by estimating probabilities using a logistic function, which is the cumulative logistic distribution. So once we have the most important features for the previous alumni donations, we can make a logistic regression model which would be helpful for prediction of donations by recent alumni.

4. (10 Points) What is R-Squared? What are some other metrics that could be better than R-Squared and why?

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. R-squared is the percentage of the response variable variation that is explained by a linear model.

Some other metrics that could be better than R-Squared are:

1. Ajusted R-squared: The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance.
2. Akaike information criterion(AIC): The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. AIC is founded on information theory. When a statistical model is used to represent the process that generated the data, the representation will almost never be exact; so some information will be lost by using the model to represent the process. AIC estimates the relative information lost by a given model: the less information a model loses, the higher the quality of that model. (In making an estimate of the information lost, AIC deals with the trade-off between the goodness of fit of the model and the simplicity of the model.)
3. The F-test: It evaluates the null hypothesis that all regression coefficients are equal to zero versus the alternative that at least one is not. An equivalent null hypothesis is that R-squared equals zero. A significant F-test indicates that the observed R-squared is reliable and is not a spurious result of oddities in the data set. Thus the F-test determines whether the proposed relationship between the response variable and the set of predictors is statistically reliable and can be useful when the research objective is either prediction or explanation.
4. RMSE: The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

5. (10 Points) How can you determine which features are the most important in your model?

There are many ways to determine which features are the most important in your model:

1. Pearson's Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. We can check the correlation of different features with the dependent feature to

select the ones with the highest correlation.

2. LDA: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.
3. ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.
4. Chi-Square: It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.
5. Stepwise regression: It is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some prespecified criterion such a p-value, AIC, adjusted R-squared.

References:

https://www.kdnuggets.com/2017/02/17-data-science-interview-questions-answers.html/2 http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit https://da5030.weebly.com/unit-07--regression.html https://www.quora.com/Given-a-database-of-all-previous-alumni-donatio https://en.wikipedia.org/wiki/Akaike_information_criterion https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/ https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-meth https://en.wikipedia.org/wiki/Stepwise_regression https://datascience.stackexchange.com/questions/9228/decision-tree-vs-knn