Module 5

Ryan S Dunn

11/25/2021

Data Science Using Python and R: Chapter 14 - Page 211: Questions #11, 12, 13, 14, & 15

11. "Subset the variables VMail Plan, Int'l Plan, CustServ Calls, and Churn into their own data frame. Change CustServ Calls into an ordered factor."

```
#import the entire data frame
churn_train <- read.csv("~/Documents/USD MS-ADS/Applied Data Mining 502/Module 5/Datasets/Churn_Training
#head(churn_train)
```

```
#subset variables into their own data frame
churn_subset <- subset(x=churn_train, select = c("VMail.Plan", "Intl.Plan", "CustServ.Calls", "Churn"))
#head(churn_subset)

#change CustServ.Calls to an ordered factor
churn_subset$CustServ.Calls <- ordered(as.factor(churn_subset$CustServ.Calls))
```

12. "Create tables for each of the four variables. Include both counts and proportions in each table. Use the tables to discuss the "baseline" distribution of each variable."

```
#put each variable into its own table
t1 <- table(churn_subset$VMail.Plan)
t2 <- table(churn_subset$Intl.Plan)
t3 <- table(churn_subset$CustServ.Calls)
t4 <- table(churn_subset$Churn)

#create the count/proportion table to each variable
t11  <- rbind(t1, round(prop.table(t1), 4))
colnames(t11) <- c("VMail.Plan = no", "VMail.Plan = yes")
rownames(t11) <- c("Count","Proportion")

t22 <- rbind(t2, round(prop.table(t2), 4))
colnames(t22) <- c("Intl.Plan = no", "Intl.Plan = yes")
rownames(t22) <- c("Count","Proportion")

t33 <- rbind(t3, round(prop.table(t3),4))
rownames(t33) <- c("Count","Proportion")

t44 <- rbind(t4, round(prop.table(t4), 4))
colnames(t44) <- rbind("Churn = False", "Churn = True")
rownames(t44) <- c("Count","Proportion")
```

```
#view the proportion tables
t11
```

```
##            VMail.Plan = no VMail.Plan = yes
## Count            2170.0000         830.0000
## Proportion          0.7233           0.2767
```

```
t22
```

```
##            Intl.Plan = no Intl.Plan = yes
## Count           2705.0000        295.0000
## Proportion         0.9017          0.0983
```

```
t33
```

```
##                   0        1        2        3        4       5       6      7
## Count      626.0000 1068.000 679.0000 383.0000 149.0000 61.0000 22.0000 8.0000
## Proportion   0.2087    0.356   0.2263   0.1277   0.0497  0.0203  0.0073 0.0027
##                  8     9
## Count      2e+00 2e+00
## Proportion 7e-04 7e-04
```

```
t44
```

```
##            Churn = False Churn = True
## Count           2564.0000     436.0000
## Proportion         0.8547       0.1453
```

13. "Obtain the association rules using the settings outlined in Section 14.4."

```
#install.packages("arules")
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
#develop all association rules with min support as 0.01 with antecedents with exactly one item, and sup
all.rules <- apriori(data = churn_subset, parameter = list(supp = 0.01, target = "rules", conf = 0.4, m
```

```
## Warning: Column(s) 1, 2, 4 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5    0.01      2
##  maxlen target  ext
##       2  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
```

```
##
## Absolute minimum support count: 30
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3000 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2
```

```
## Warning in apriori(data = churn_subset, parameter = list(supp = 0.01, target =
## "rules", : Mining stopped (maxlen reached). Only patterns up to a length of 2
## returned!
```

```
##  done [0.00s].
## writing ... [32 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#view the top 10 rules sorted by lift
inspect(head(all.rules, by = 'lift', n = 10))
```

```
##      lhs                    rhs             support    confidence coverage
## [1]  {CustServ.Calls=5} => {Churn=True}    0.01200000 0.5901639  0.02033333
## [2]  {CustServ.Calls=4} => {Churn=True}    0.02266667 0.4563758  0.04966667
## [3]  {Intl.Plan=yes}    => {Churn=True}    0.04233333 0.4305085  0.09833333
## [4]  {Churn=True}       => {VMail.Plan=no} 0.12066667 0.8302752  0.14533333
## [5]  {CustServ.Calls=3} => {VMail.Plan=no} 0.09933333 0.7780679  0.12766667
## [6]  {VMail.Plan=yes}   => {Churn=False}   0.25200000 0.9108434  0.27666667
## [7]  {CustServ.Calls=1} => {Churn=False}   0.32000000 0.8988764  0.35600000
## [8]  {CustServ.Calls=3} => {Churn=False}   0.11433333 0.8955614  0.12766667
## [9]  {CustServ.Calls=4} => {VMail.Plan=no} 0.03733333 0.7516779  0.04966667
## [10] {CustServ.Calls=2} => {Churn=False}   0.20066667 0.8865979  0.22633333
##      lift     count
## [1]  4.060761  36
## [2]  3.140201  68
## [3]  2.962214 127
## [4]  1.147846 362
## [5]  1.075670 298
## [6]  1.065729 756
## [7]  1.051727 960
## [8]  1.047849 343
## [9]  1.039186 112
## [10] 1.037361 602
```

14. "Subset the rules from the previous exercise so none of the antecedents contain the Churn variable. Display the rules, sorted by descending lift value."

```
#identify which rules have Churn in the antecedent, lhs
all.rules.ant.df <- as(as(attr(all.rules, "lhs"), "transactions"), "data.frame")

#return true and false vectors
t_true <- all.rules.ant.df$items == "{Churn=True}"
t_false <- all.rules.ant.df$items == "{Churn=False}"

#single vector of zeros and ones, where the ones indicate antecedents that do not contain Churn
non.churn.ant <- abs(t_true + t_false -1)
```

```r
#subset all rulls to good rules, with only the roles that have a non.churn.ant = 1
good.rules <- all.rules[non.churn.ant == 1]
```

```r
#view the top 10 good rules by lift
inspect(head(good.rules, by = "lift", n = 10))
```

```
##      lhs                    rhs             support    confidence coverage
## [1]  {CustServ.Calls=5} => {Churn=True}    0.01200000 0.5901639  0.02033333
## [2]  {CustServ.Calls=4} => {Churn=True}    0.02266667 0.4563758  0.04966667
## [3]  {Intl.Plan=yes}    => {Churn=True}    0.04233333 0.4305085  0.09833333
## [4]  {CustServ.Calls=3} => {VMail.Plan=no} 0.09933333 0.7780679  0.12766667
## [5]  {VMail.Plan=yes}   => {Churn=False}   0.25200000 0.9108434  0.27666667
## [6]  {CustServ.Calls=1} => {Churn=False}   0.32000000 0.8988764  0.35600000
## [7]  {CustServ.Calls=3} => {Churn=False}   0.11433333 0.8955614  0.12766667
## [8]  {CustServ.Calls=4} => {VMail.Plan=no} 0.03733333 0.7516779  0.04966667
## [9]  {CustServ.Calls=2} => {Churn=False}   0.20066667 0.8865979  0.22633333
## [10] {Intl.Plan=no}     => {Churn=False}   0.79866667 0.8857671  0.90166667
##      lift     count
## [1]  4.060761   36
## [2]  3.140201   68
## [3]  2.962214  127
## [4]  1.075670  298
## [5]  1.065729  756
## [6]  1.051727  960
## [7]  1.047849  343
## [8]  1.039186  112
## [9]  1.037361  602
## [10] 1.036389 2396
```

15. "Obtain association rules using the confidence difference criterion outlined in Section 14.6."

```r
#include confidence different criterion in association rule setting
rules.condiff <- apriori(data = churn_subset, parameter = list(arem = "diff", aval = TRUE,
                         minval = 0.4, supp = 0.01, target = "rules", conf = 0.05, minlen = 2, maxlen =
```

```
## Warning: Column(s) 1, 2, 4 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem aval originalSupport maxtime support minlen maxlen
##        0.05    0.4    1 diff TRUE            TRUE       5    0.01      2      2
##  target  ext
##   rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 30
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3000 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2
```

```
## Warning in apriori(data = churn_subset, parameter = list(arem = "diff", : Mining
## stopped (maxlen reached). Only patterns up to a length of 2 returned!
```

```
##  done [0.00s].
## writing ... [1 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(rules.condiff, by = "lift", n = 10))
```

```
##     lhs                   rhs          support confidence diff      coverage
## [1] {CustServ.Calls=5} => {Churn=True} 0.012   0.5901639  0.4448306 0.02033333
##     lift     count
## [1] 4.060761 36
```

Introduction to Data Mining
Exercises 5.10 to-Pg 438
Questions #6,8, 15

6. Consider the market basket transaction shown in table 5.21

Transaction ID	Items Bought
1	{Milk, Beer, Diapers}
2	{Bread, Butter, Milk}
3	{Milk, Diapers, Cookies}
4	{Bread, Butter, Cookies}
5	{Beer, Cookies, Diapers}
6	{Milk, Diapers, Bread, Butter}
7	{Bread, Butter, Diapers}
8	{Beer, Diapers}
9	{Milk, Diapers, Bread, Butter}
10	{Beer, Cookies}

(a) What is the maximum number of association rules that can be extracted from this data? (including rules that have zero support)

$R = 3^d - 2^{d+1} + 1$

where d = items $= 6$

$R = 3^6 - 2^{6+1} + 1 = 729 - 128 + 1$

$R = 602$

(b) What is the maximum size of frequent itemsets that can be extracted, (assuming minsup 70)?

Transaction IDs 6 & 9 both have 4 items in the itemset, therefor the max size of frequent itemsets that can be extracted is 4.

(c) Write an expression for the maximum number of size-3 itemsets that can be derived from this data set.

6 items & 3 itemsets:

$$\frac{6!}{3!3!} = \frac{6 \cdot 5 \cdot 4}{3 \cdot 2 \cdot 1} = \frac{2}{1} \cdot \frac{2}{1} \cdot \frac{5}{1} = \frac{20}{1} = 20$$

(d) Find an itemset (of size 2 or larger) that has the largest support.

T	Bread	Butter	Beer	Diapers	Cookies	Milk
1	0	0	1	1	0	1
2	1	1	0	0	0	1
3	0	0	0	1	1	1
4	1	1	0	0	1	0
5	0	0	1	1	1	0
6	1	1	0	1	0	1
7	1	1	0	1	0	0
8	0	0	1	1	0	0
9	1	1	0	1	0	1
10	0	0	1	0	1	0

{Bread, Butter} has the most support. See that each time bread is bought, so is butter, and there are zero instances where one is bought & not the other.

(e) Find a pair of items, a & b, such that the rules $\{a\} \rightarrow \{b\}$ and $\{b\} \rightarrow \{a\}$ have the same confidence.

{Beer, Cookies} = because there is an equal # of 1→1 0→1 0→0 & 1→0 for both are the same a→b & b→a

{Bread, Butter} = because for all instances of bread, there is butter

a→b = b→a

6. Consider the following set of Frequent 3 itemset:

{1,2,3}, {1,2,4}, {1,2,5}, {1,3,4}, {1,3,5}
{2,3,4}, {2,3,5}, {3,4,5}

Assume there are only 5 items in the dataset.

(a) List all candidate 4-itemsets obtained by a candidate generation procedure using the $F_{k-1} \times F_1$ merging strategy.

{1,2,3,4}, {1,2,3,5}
{1,3,4,5}
{2,3,4,5}

#duplicates were not included

(b) List all candidate 4-itemsets obtained by the candidate generation procedure Apriori.

also {1,2,3,4}, {1,2,3,5}
{1,3,4,5}, {2,3,4,5}

(c) List all 4-itemsets that survive the candidate pruning step of the Apriori algorithm.

15. Answer the following questions using the data set in 5.3.4.

(a) Which data set(s) will produce the most number of frequent itemsets?

Dataset (e), because for each transaction, there are many frequent similar itemsets

(b) Which data set(s) will produce the fewest number of frequent itemsets

Dataset (d), because there are virtually zero itemsets with any high frequency.

(c) Which data set(s) will produce the largest frequent itemset.

Dataset (e), because many of the itemsets lines are "long".

(d) Which data set(s) will produce frequent itemsets with highest maximum support.

Dataset (a) - because the frequency is consistant.

(e) Which data set(s) will produce itemsets with wide varying support levels?

Dataset (e), because some item lines are

long & complete and some have virtually
no support.