Data Mining HW 4 Submission

Patrick Chase

5/3/2021

message=FALSE, warning=FALSE, echo=FALSE, error=TRUE

# 1.

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.72330  
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.04756  
## Cumulative Proportion 0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.90009  
## PC8 PC9 PC10 PC11  
## Standard deviation 0.70817 0.58054 0.4772 0.18119  
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298  
## Cumulative Proportion 0.94568 0.97632 0.9970 1.00000

PCA 1 through PCA 9 capture more than 95% of the variability in our data set. I’ll rely on only those for my PCA model.

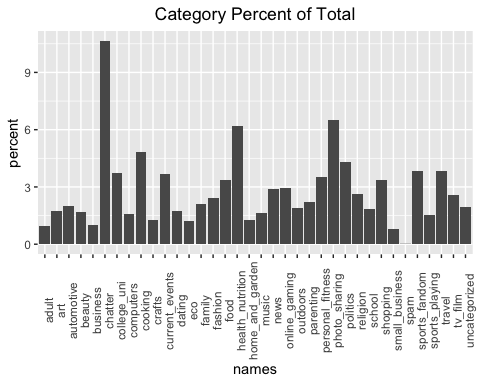
## # A tibble: 1 x 6  
## pca\_red\_accuracy rmse\_pca\_red pca\_qual\_accura… rmse\_pca\_qual cluster\_redl\_ac…  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 99.2 7.81 0.616 0.727 54.9  
## # … with 1 more variable: cluster\_qual\_accuracy <dbl>

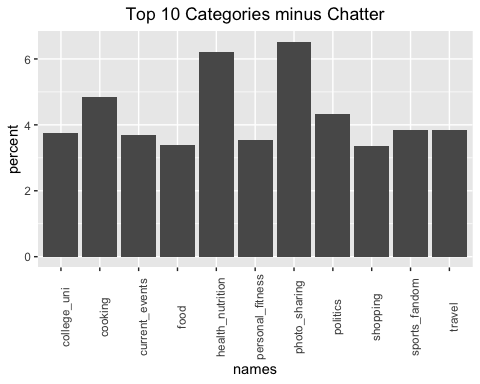
For this specific task, I think PCA makes a lot more sense, particularly for the binary classification between red and white where I can achieve accuracy of 98%. My PCA model estimating quality is just barely wasn’t particularly good. As far as I can tell, the K-means models aren’t outputting information that is practically useful. I’m not sure if that’s because I made a mistake or if there is something else going on.

# 2.

## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror

## Error in eval(substitute(select), nl, parent.frame()): object 'X' not found

 This first graph is showing the percentage breakdown for each category in our data set. Let’s take a closer look by focusing on the top 10 categories removing chatter as it is not substantively useful.

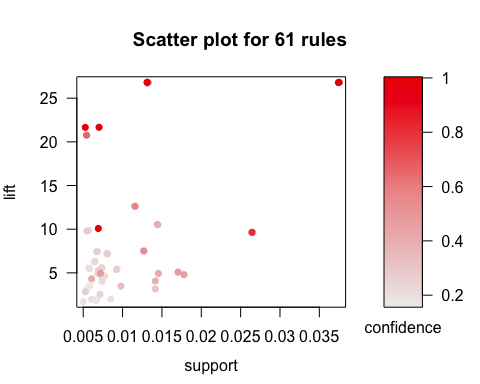


NutrientH20’s social media followers seem to be focused on a typical range of what many consumers on twitter are. That said, a point of interest is the relatively large proportion that is related to food in general, when we aggregate between cooking, food, and health\_nutrition. This in combination with the large share of photo\_sharing gives us an idea of what some followers may be interested. The market segment in this case would be foodies who enjoy sharing their food endevours with their friends through twitter. This could point to a marketing strategy of engaging with chefs turned social media influencers. This would expand reach and give NutrientH20’s consumers what they are interested.

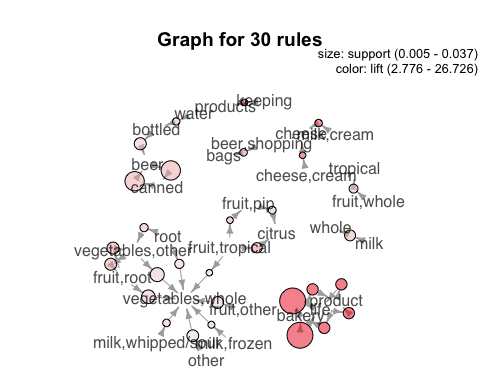
# 3.

## distribution of transactions with duplicates:  
## 1   
## 14

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.15 0.1 1 none FALSE TRUE 5 0.005 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: 49   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[6928 item(s), 9835 transaction(s)] done [0.01s].  
## sorting and recoding items ... [99 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 done [0.00s].  
## writing ... [61 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

 I chose a low level for support because I don’t think a grocer would be particularly interested in focusing on products that all of their customers already buy. My threshhold for confidence was 15% because I think it would be more useufl to show associations of products in order to inform store organization. I decided to subset my rules to focus on those with the highest confidence in the graphic below.

## Available control parameters (with default values):  
## main = Graph for 30 rules  
## max = 100  
## nodeCol = c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF", "#EE1E1EFF", "#EE2222FF", "#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF", "#EE3131FF", "#EE3434FF", "#EE3737FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF", "#EE4444FF", "#EE4747FF", "#EE4A4AFF", "#EE4D4DFF", "#EE5050FF", "#EE5353FF", "#EE5656FF", "#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6262FF", "#EE6666FF", "#EE6969FF", "#EE6C6CFF", "#EE6F6FFF", "#EE7272FF", "#EE7575FF", "#EE7878FF", "#EE7B7BFF", "#EE7E7EFF", "#EE8181FF", "#EE8484FF", "#EE8888FF", "#EE8B8BFF", "#EE8E8EFF", "#EE9191FF", "#EE9494FF", "#EE9797FF", "#EE9999FF", "#EE9B9BFF", "#EE9D9DFF", "#EE9F9FFF", "#EEA0A0FF", "#EEA2A2FF", "#EEA4A4FF", "#EEA5A5FF", "#EEA7A7FF", "#EEA9A9FF", "#EEABABFF", "#EEACACFF", "#EEAEAEFF", "#EEB0B0FF", "#EEB1B1FF", "#EEB3B3FF", "#EEB5B5FF", "#EEB7B7FF", "#EEB8B8FF", "#EEBABAFF", "#EEBCBCFF", "#EEBDBDFF", "#EEBFBFFF", "#EEC1C1FF", "#EEC3C3FF", "#EEC4C4FF", "#EEC6C6FF", "#EEC8C8FF", "#EEC9C9FF", "#EECBCBFF", "#EECDCDFF", "#EECFCFFF", "#EED0D0FF", "#EED2D2FF", "#EED4D4FF", "#EED5D5FF", "#EED7D7FF", "#EED9D9FF", "#EEDBDBFF", "#EEDCDCFF", "#EEDEDEFF", "#EEE0E0FF", "#EEE1E1FF", "#EEE3E3FF", "#EEE5E5FF", "#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")  
## itemnodeCol = #66CC66FF  
## edgeCol = #ABABABFF  
## labelCol = #000000B3  
## itemLabels = TRUE  
## measureLabels = FALSE  
## precision = 3  
## arrowSize = 0.5  
## alpha = 0.5  
## cex = 1  
## layout = NULL  
## layoutParams = list()  
## engine = igraph  
## plot = TRUE  
## plot\_options = list()  
## verbose = FALSE

 This graphic shows rules that seem to follow the typical organization of a grocery store. Dairy products are largely purchased in tandem with each other, and most customers seem to be getting whole vegetables along with many other products.

# 4.

For my data cleaning procedure I basically followed you’re work exactly. And even with that I struggled to transition it to a usable format where I could use the principal components I identified to build a classification model. That said, I had intended to build a model where I’d focus on classifying the work of one Mure Dickie. I imported the Document Term Matrix into a normal matrix and normalized it. I had intended to use my PCs as variables in a logit model, with the outcome variable being Author == “MureDickie”. I suspect my classification may have been alright, although probably not great. My issue, ultimately was a data processing one. I was ultimately able to get my model to run, but I have no idea what the accuracy of it is.

## Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels): 'data' must be a data.frame, not a matrix or an array