**Contents**

[Dropbox Link for all Files/Process/Code: 2](#_Toc520881701)

[Files Used: 2](#_Toc520881702)

[Creating student and applicant dataset 2](#_Toc520881703)

[Transformations 2](#_Toc520881704)

[Creating students\_data\_final.csv 2](#_Toc520881705)

[Cleansing of applicant\_data.csv file 2](#_Toc520881706)

[Final Raw data (Combined Data) final Cleansing: 3](#_Toc520881707)

[Assumptions 3](#_Toc520881708)

[Total statistics of combined data 3](#_Toc520881709)

[R code 3](#_Toc520881710)

[#Rattle steps: 6](#_Toc520881711)

[Solutions of issues in Rattle 10](#_Toc520881712)

[New Title Level Logic 11](#_Toc520881713)

[Code in R for new levels 12](#_Toc520881714)

## Dropbox Link for all Files/Process/Code:

<https://www.dropbox.com/sh/xudxmiu2g9tac0y/AAC5pj52pc_ZiV_9DbOgx1Tza?dl=0>

Folder - MEM Personas Incl Apps 2018-07

## Files Used:

Below files can be found in folder Datasets:

1. applicants\_data\_July2018-> created from “mem funnel apps” sent in mail by GiaPhu, took the newer sheet.
2. MEM students\_winter 2015 to present
3. MEM students\_winter 2013 to 2014 – sheet1

## Creating student and applicant dataset

### Transformations

For “MEM students\_winter 2013 to 2014” do the below changes



### Creating students\_data\_final.csv

1. Combine “MEM students\_winter 2015 to present” and MEM students\_winter 2013 to 2014 (sheet 4) :->students\_data\_final.csv (match the headers and then remove headers from 2nd file’s data and combine and place data in sheet Student\_ToLoad sheet)
2. Take Student\_ToLoad sheet from students\_data\_final.csv as the final students’ data.
3. Check no blanks in Program Binary, gender binary and Birth
4. Add one column in start called Type and add value = student
5. Added below as they are in Applicant’s data sheet

Add GRE (Required for those with less than 3 years' of experience): Test Date

GMAT (Required for those with less than 3 years' of experience): Test Date

Add Grad GPA, Grad GPA Scale

Are you a veteran?

Status

### Cleansing of applicant\_data.csv file

Total records -1244, we need to make it like Student\_ToLoad sheet. For “mem funnel apps” we did the below changes and created applicant\_data.csv file.



## Final Raw data (Combined Data) final Cleansing:

The final file is NW\_FinalDataLoad.xls and below changes are done in it

1. Add column- AgeAtTerm = IF(ISBLANK(Birth),"",(App Term Year-YEAR(Birth)))
2. Removed all records underage below 21 i.e. AgeAtTerm<21(13 records removed)
3. Create a column-No of years of work experience after undergraduate degree? And copy values from How many years of full time work after undergraduate degree? And then update blanks and 10+ yrs exp as below

If Age >22 and No of yr exp= Blanks, calculate as Age-23

If Age<=22 and No of yrs exp = Blanks, put zero

If no of yrs exp=10+, exp = Age-23

### Assumptions

If Gender= blanks🡪Gender Binary by default takes it as Male or 0

If Program Blanks – Program Binary by default takes it to 0 or Part time, change those values back to 1

### Total statistics of combined data

Check

104 columns

1244-applicants

227- students

13 underage records deleted

Total – 1458 record to be clustered

## R code

install.packages("stringr")#extract the year part

install.packages("lubridate")#extract the year

install.packages("dplyr")

install.packages("pmml")

install.packages("rattle")

library(pmml)

library(stringr)

library(lubridate)

library(dplyr)

library(rattle)

#set the working directory

#setwd("C:/Users/Samtha Reddy/Dropbox (Personal)/Copy of Work/ClusterAnalysis\_7-06-2018")

setwd("C:/Users/Samtha Reddy/Dropbox (Converge Consulting)/Converge Consulting/Client/Northwestern/NW Analytics 18-05/MEM Personas Incl Apps 2018-07")

nwmem\_df <- read.csv(file = "NW\_FinalDataLoad.csv", header = TRUE, sep = ",")

#View(nwmem\_df)

#Age At Term calculated in excel

#Create Binary factor for "U.S Citizen or not"

uscitizen <- which(nwmem\_df$Country.of.Citizenship=="USA")

nwmem\_df$BinaryUSCitizen <- 0 #So all rows including with column value blanks get 0

nwmem\_df$BinaryUSCitizen[uscitizen] <- 1

#check if all USA citizen have value 1

#nwmem\_df$Country.of.Citizenship[nwmem\_df$BinaryUSCitizen==1]

# make gpa as numeric

nwmem\_df$Undergrad.GPA <- as.numeric(as.character(nwmem\_df$Undergrad.GPA))

# Warning message: NAs introduced by coercion

nwmem\_df$Undergrad.GPA.scale <- as.numeric(as.character(nwmem\_df$Undergrad.GPA.scale))

#Warning message: NAs introduced by coercion

# create percentage for gpa/gpa scale

nwmem\_df$GPAPercent <- nwmem\_df$Undergrad.GPA /nwmem\_df$Undergrad.GPA.scale

# Calculate GPA for 4 point scale

nwmem\_df$GPA4pt <- nwmem\_df$GPAPercent\*4

# prep new column with lowest rank

nwmem\_df$TitleLevel <- 1

# create Title column for this specific data file

nwmem\_df$Title <- as.character(nwmem\_df$Current.Employment.1..Job.Title)

#pull out index of titles

chief <- which(grepl("chief", nwmem\_df$Title, ignore.case = T))

ceo<- which(grepl("cEO", nwmem\_df$Title, ignore.case = T))

president<- which(grepl("president", nwmem\_df$Title, ignore.case = T))

Cofounder<- which(grepl("Co founder", nwmem\_df$Title, ignore.case = T))# I added this

coo<- which(grepl("coo", nwmem\_df$Title, ignore.case = T))

cfo<- which(grepl("cfo", nwmem\_df$Title, ignore.case = T))

vicepresident<- which(grepl("vice president", nwmem\_df$Title, ignore.case = T))

vp<- which(grepl("vp", nwmem\_df$Title, ignore.case = T))

attorney<- which(grepl("attorney", nwmem\_df$Title, ignore.case = T))

director<- which(grepl("director", nwmem\_df$Title, ignore.case = T))

vice <- which(grepl("vice", nwmem\_df$Title, ignore.case = T))

controller<- which(grepl("contoller", nwmem\_df$Title, ignore.case = T))

head<- which(grepl("head", nwmem\_df$Title, ignore.case = T))

engineer<- which(grepl("engineer", nwmem\_df$Title, ignore.case = T))

lead<- which(grepl("lead", nwmem\_df$Title, ignore.case = T))

senior <- which(grepl("senior", nwmem\_df$Title, ignore.case = T))

sr <- which(grepl("sr", nwmem\_df$Title, ignore.case = T))

specialist<- which(grepl("specialist", nwmem\_df$Title, ignore.case = T))

analyst<- which(grepl("analyst", nwmem\_df$Title, ignore.case = T))

supervisor<- which(grepl("supervisor", nwmem\_df$Title, ignore.case = T))

manager<- which(grepl("manager", nwmem\_df$Title, ignore.case = T))

# fill in appropriate rank for the record:

# 5: chief, ceo or president

nwmem\_df$TitleLevel[chief] <- 5

nwmem\_df$TitleLevel[ceo] <- 5

nwmem\_df$TitleLevel[president] <- 5

nwmem\_df$TitleLevel[coo] <- 5

nwmem\_df$TitleLevel[cfo] <- 5

nwmem\_df$TitleLevel[Cofounder] <- 5

# 4: vp or vicepresident

nwmem\_df$TitleLevel[vp] <- 4

nwmem\_df$TitleLevel[vicepresident] <- 4

nwmem\_df$TitleLevel[vice] <- 4

# 3: attorney, director, specialist, Manager

nwmem\_df$TitleLevel[attorney] <- 3

nwmem\_df$TitleLevel[director] <- 3

nwmem\_df$TitleLevel[specialist] <- 3

nwmem\_df$TitleLevel[manager] <- 3

nwmem\_df$TitleLevel[controller] <- 3

nwmem\_df$TitleLevel[head] <- 3

# 2: analyst, Supervisor

nwmem\_df$TitleLevel[analyst] <- 2

nwmem\_df$TitleLevel[supervisor] <- 2

nwmem\_df$TitleLevel[engineer] <- 2

nwmem\_df$TitleLevel[lead] <- 2

nwmem\_df$TitleLevel[senior] <- 2

nwmem\_df$TitleLevel[sr] <- 2

# 1: everything else

# this field used in clustering hence changing it to numeric from factor

nwmem\_df$No.of..years.of.work.experience.after.undergraduate.degree. <- as.numeric(as.character(nwmem\_df$No.of..years.of.work.experience.after.undergraduate.degree.))

#Cleansing

#1. Checking if any na's in AgeatTerm,

#which(is.na(nwmem\_df$Age))#3 records

#---------------------------------------------

#---- spliting dataset by program type

ft <-subset(nwmem\_df, nwmem\_df$Program.Binary==1)

pt <-subset(nwmem\_df, nwmem\_df$Program.Binary==0)

# ---

write.csv(nwmem\_df, file = "NWStudentApplicantUpdatedinR.csv",row.names=FALSE)

write.csv(ft, file = "NWStudentApplicantFullTimeUpdatedinR.csv",row.names=FALSE)

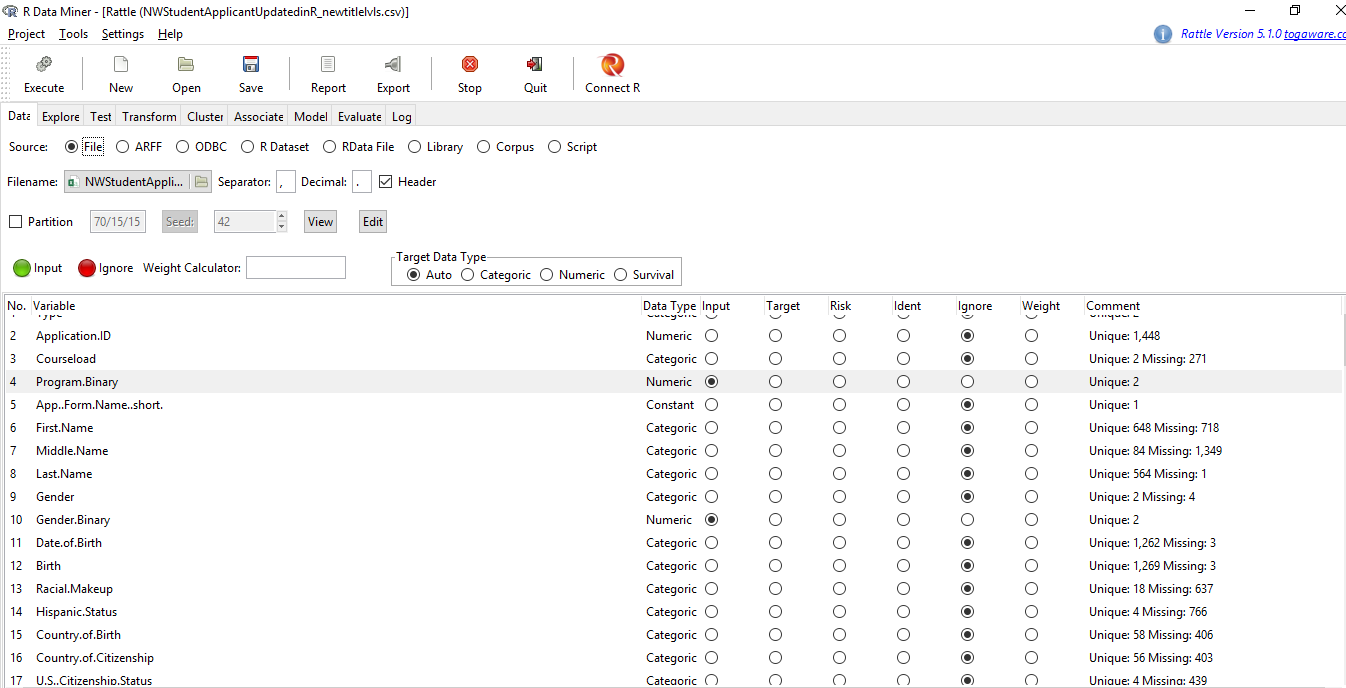
write.csv(pt, file = "NWStudentApplicantParTimeUpdatedinR.csv",row.names=FALSE)

# do clustering in Rattle

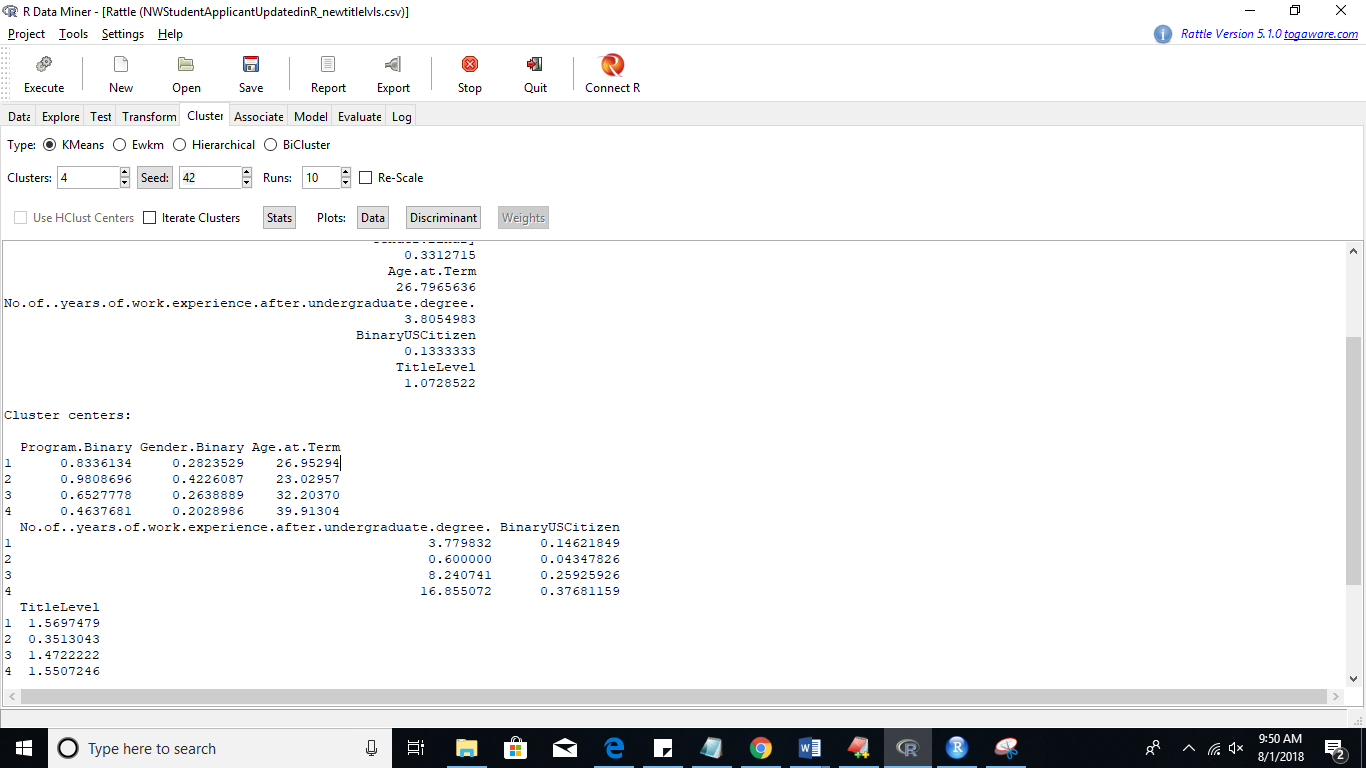
rattle()—

### #Rattle steps:

1. Rattle s/w opens up a separate window as shown below:

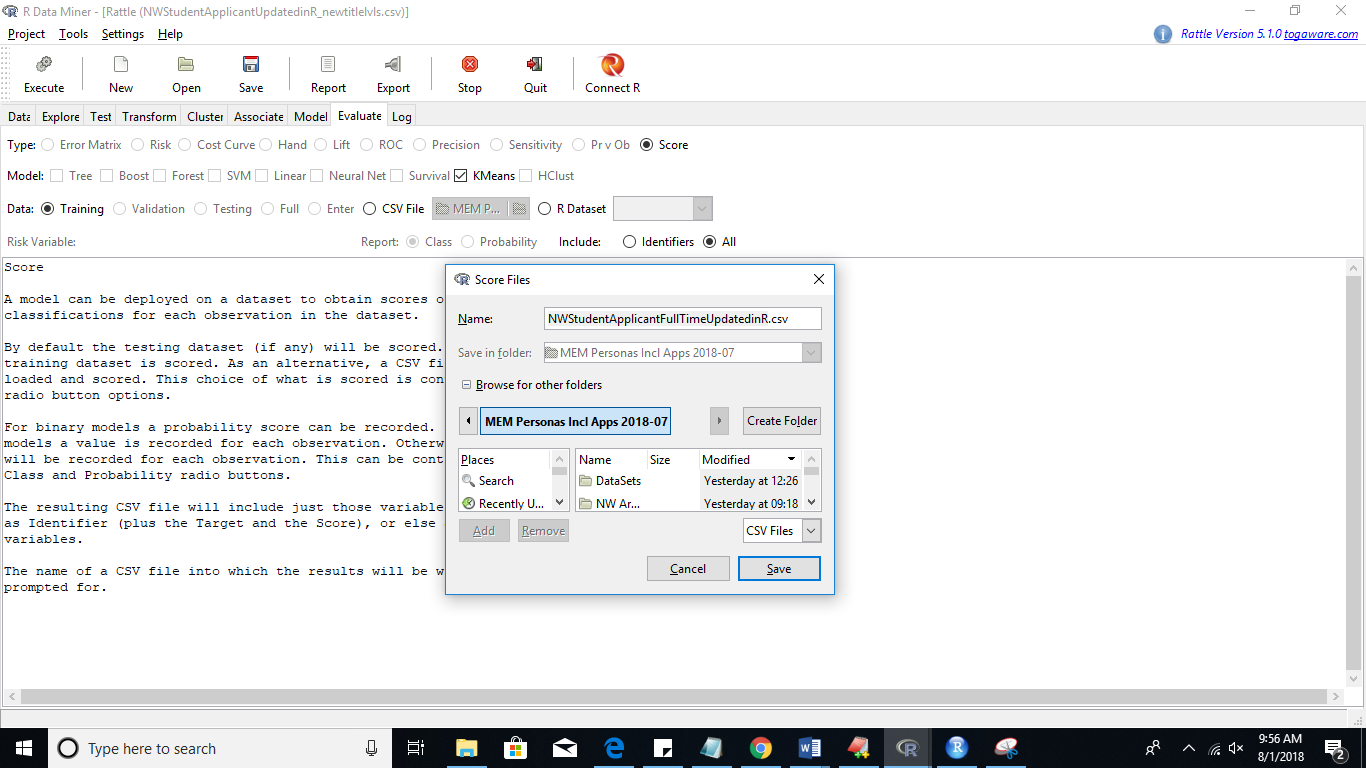


1. Load the file “NWStudentApplicantUpdatedinR.csv” we last saved in R in Filename field. Uncheck the Partition field. Click on Execute. It will show all the variables of the file.
2. In input field, select “ignore” for all variables except- Gender.Binary, Program.Binary, Age.at.Term, No.of..years.of.work.experience.after.undergraduate.degree., BinaryUSCitizen, TitleLevel.
3. Click on Cluster tab-> clusters=4, seed =42(this is just a random number to ensure reproducibility when running the same data) runs =10, Re-scale = uncheck. And click on Execute.



1. Click on Evaluate tab, select Training, Include: All ( this is done ensure you get cluster value for each record all with records details or else you will just get cluster values), click execute.

A window will pop up asking to give name and path to save the score file. Click on save. We save the file as “NWStudentApplicantUpdatedinR\_train\_score\_all.csv”.



1. You can change the cluster value as 5 in cluster tab and click execute to get 5 cluster analysis.
2. Continue below steps in R

#reloaded the file –“ NWStudentApplicantUpdatedinR\_train\_score\_all.csv “ we got from Rattle which has cluster information of each record.

#before loading the csv fix-> racial.makeup column remove <U+0095> ctl+H

nwmem\_df\_cl <- read.csv(file = "NWStudentApplicantUpdatedinR\_train\_score\_all.csv", header = TRUE, sep = ",")

#%of full time in each cluster

tapply(nwmem\_df\_cl$Program.Binary, nwmem\_df\_cl$kmeans, mean)

#% of student and applicant in each cluster

tapply(nwmem\_df\_cl$Type, nwmem\_df\_cl$kmeans, summary)

# Checking geography

#default all are level 3- all other us states

nwmem\_df\_cl$GeoLevel <- 3

nwmem\_df\_cl$US.State <- as.character(nwmem\_df\_cl$US.State)

international<-which(is.na(nwmem\_df\_cl$US.State))

nwmem\_df\_cl$GeoLevel[international]<-5

coast<-which(nwmem\_df\_cl$US.State %in% c('Maine','ME', 'New Hampshire ','NH', 'Massachusetts','MA', 'Rhode Island','RI', 'Connecticut','CT', 'New York','NY', 'New Jersey','NJ', 'Delaware','DE', 'Maryland','MD', 'Virginia','VA', 'North Carolina','NC', 'South Carolina','SC', 'Georgia','GA', 'Florida.','FL', 'California ','CA', 'Oregon','OR', 'Washington','WA', 'Alaska','AK', 'Hawaii','HI'))

nwmem\_df\_cl$GeoLevel[coast]<-4

border <- which(nwmem\_df\_cl$US.State %in% c('Wisconsin','WI','Michigan','MI','Indiana','IN','Kentucky','KY','Missouri','MO','Iowa','IA','Tennessee','TN'))

nwmem\_df\_cl$GeoLevel[border]<-2

IL <- which(nwmem\_df\_cl$US.State %in% c('IL','IL - Illinois','IL.','Illinois','Illionis'))

nwmem\_df\_cl$GeoLevel[IL]<-1

#% of geo in each cluster

nwmem\_df\_cl$GeoLevel<-as.factor(nwmem\_df\_cl$GeoLevel)

tapply(nwmem\_df\_cl$GeoLevel, nwmem\_df\_cl$kmeans, summary)

#-----------------5 Clusters Analysis------------------------------#

#reloaded the file we got from Rattle for 5 clusters which has cluster information of each record

#before loading the csv fix-> racial.makeup column remove <U+0095> ctl+H

setwd("C:/Users/Samtha Reddy/Dropbox (Personal)/Copy of Work/ClusterAnalysis\_7-06-2018")

nwmem\_df\_5cl <- read.csv(file = "NWStudentApplicantUpdatedinR\_train\_score\_all\_5Clusters.csv", header = TRUE, sep = ",")

#% of student and applicant in each cluster

tapply(nwmem\_df\_5cl$Type, nwmem\_df\_5cl$kmeans, summary)

nwmem\_df\_cl\_students <- nwmem\_df\_cl[nwmem\_df\_cl$Type =="student", ]

tapply(nwmem\_df\_cl\_students$GeoLevel, nwmem\_df\_cl\_students$kmeans, summary)

## Solutions of issues in Rattle

1. Error: When trying to save score file Rattle does not prompt to give the new file name

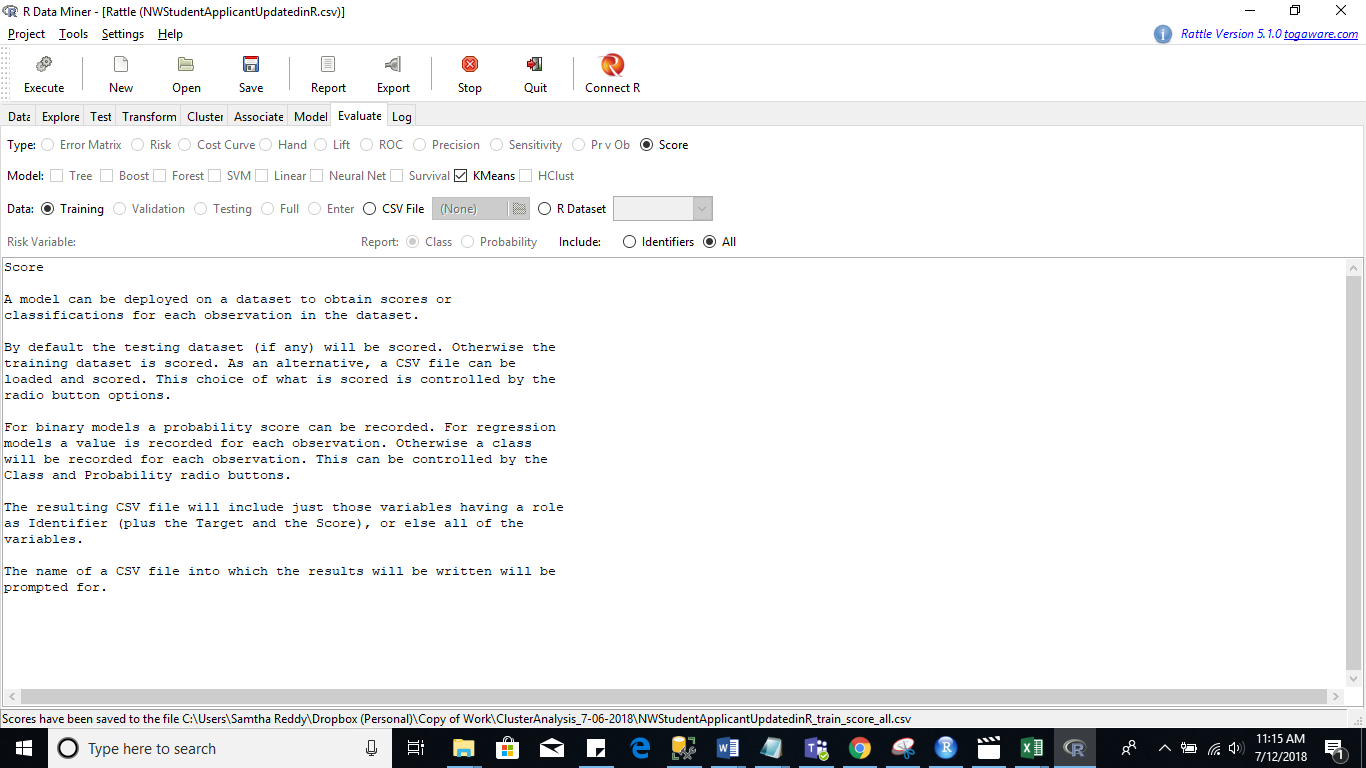
Fix: Make sure to install and load package pmml before you had loaded and opened Rattle.

1. Error: Error in. subset2(x, i, exact = exact): attempt to select less than one element in get1index.

Fix: Make sure to install and load(dplyr) before loading and running Rattle()

1. Error in if (length(grep("\\.csv$", testname)) && !getSelectedVariables("target") %in% : missing value where TRUE/FALSE needed. (Also, when prompted to give csv filename always give a new name- IT WILL NOT ABLE TO OVERWRITE IT if you are folder is in DROPBOX, it will say permission denied.)

Fix: ALWAYS SELECT TRAINING or else you will keep getting in R window like below.



## New Title Level Logic

Word search from top to bottom

Convert all titles to uppercase, match on upper case of keywords and terms below:

Level 6 – C-Suite

Level 5 – VP/ SVP/ Senior Executives

Level 4 – Director

Level 3 – Manager/ Supervisor/ Lead

Level 2 – Analyst/ Engineer

Level 1 – Intern/ Student

Level 0 – Not Categorized

Level 6

Chief

CEO

CFO

COO

Pres

Chair

Managing

Found

Partner

Level 5

Vice

VP

Head

Level 4

Dir

Fellow

Level 3

Manage

Mang

Sup

Lead

Senior

Sr

Consultant

Office

Teach

Level 2

Anal

Engineer

Ass

Research

Dev

Tech

Staff

Level 1

Intern

Student

Train

### Code in R for new levels

chief <- which(grepl("chief", nwmem\_df$Title, ignore.case = T))

ceo<- which(grepl("cEO", nwmem\_df$Title, ignore.case = T))

president<- which(grepl("president", nwmem\_df$Title, ignore.case = T))

founder<- which(grepl("found", nwmem\_df$Title, ignore.case = T))

coo<- which(grepl("coo", nwmem\_df$Title, ignore.case = T))

cfo<- which(grepl("cfo", nwmem\_df$Title, ignore.case = T))

chair<- which(grepl("chair", nwmem\_df$Title, ignore.case = T))

managing<- which(grepl("managing", nwmem\_df$Title, ignore.case = T))

partner<- which(grepl("partner", nwmem\_df$Title, ignore.case = T))

vice<- which(grepl("vice", nwmem\_df$Title, ignore.case = T))

vp<- which(grepl("vp", nwmem\_df$Title, ignore.case = T))

#attorney<- which(grepl("attorney", nwmem\_df$Title, ignore.case = T))

#controller<- which(grepl("contoller", nwmem\_df$Title, ignore.case = T))

head<- which(grepl("head", nwmem\_df$Title, ignore.case = T))

dire<- which(grepl("dir", nwmem\_df$Title, ignore.case = T))

fellow<- which(grepl("fellow", nwmem\_df$Title, ignore.case = T))

lead<- which(grepl("lead", nwmem\_df$Title, ignore.case = T))

senior <- which(grepl("senior", nwmem\_df$Title, ignore.case = T))

sr <- which(grepl("sr", nwmem\_df$Title, ignore.case = T))

sup<- which(grepl("sup", nwmem\_df$Title, ignore.case = T))

manage<- which(grepl("manage", nwmem\_df$Title, ignore.case = T))

mang<- which(grepl("mang", nwmem\_df$Title, ignore.case = T))

consultant<- which(grepl("consultant", nwmem\_df$Title, ignore.case = T))

office<- which(grepl("office", nwmem\_df$Title, ignore.case = T))

teach<- which(grepl("teach", nwmem\_df$Title, ignore.case = T))

engineer<- which(grepl("engineer", nwmem\_df$Title, ignore.case = T))

specialist<- which(grepl("specialist", nwmem\_df$Title, ignore.case = T))

anal<- which(grepl("anal", nwmem\_df$Title, ignore.case = T))

staff<- which(grepl("staff", nwmem\_df$Title, ignore.case = T))

ass<- which(grepl("ass", nwmem\_df$Title, ignore.case = T))

tech<- which(grepl("tech", nwmem\_df$Title, ignore.case = T))

research<- which(grepl("research", nwmem\_df$Title, ignore.case = T))

dev<- which(grepl("dev", nwmem\_df$Title, ignore.case = T))

intern<- which(grepl("intern", nwmem\_df$Title, ignore.case = T))

student<- which(grepl("student", nwmem\_df$Title, ignore.case = T))

train<- which(grepl("train", nwmem\_df$Title, ignore.case = T))

# fill in appropriate rank for the record:

# 6: chief, ceo or president

nwmem\_df$TitleLevel[chief] <- 6

nwmem\_df$TitleLevel[ceo] <- 6

nwmem\_df$TitleLevel[president] <- 6

nwmem\_df$TitleLevel[coo] <- 6

nwmem\_df$TitleLevel[cfo] <- 6

nwmem\_df$TitleLevel[founder] <- 6

nwmem\_df$TitleLevel[chair] <- 6

nwmem\_df$TitleLevel[managing] <- 6

nwmem\_df$TitleLevel[partner] <- 6

# 5: vp or vicepresident

nwmem\_df$TitleLevel[vice] <- 5

nwmem\_df$TitleLevel[vp] <- 5

nwmem\_df$TitleLevel[head] <- 5

# 4: Dir,Fellow

nwmem\_df$TitleLevel[dire] <- 4

nwmem\_df$TitleLevel[fellow] <- 4

# 3: Manager, lead, supervisor

nwmem\_df$TitleLevel[manage] <- 3

nwmem\_df$TitleLevel[mang] <- 3

nwmem\_df$TitleLevel[sup] <- 3

nwmem\_df$TitleLevel[lead] <- 3

nwmem\_df$TitleLevel[senior] <- 3

nwmem\_df$TitleLevel[sr] <- 3

nwmem\_df$TitleLevel[consultant] <- 3

nwmem\_df$TitleLevel[office] <- 3

nwmem\_df$TitleLevel[teach] <- 3

# 2: analyst, engineer

nwmem\_df$TitleLevel[engineer] <- 2

nwmem\_df$TitleLevel[specialist] <- 2

nwmem\_df$TitleLevel[anal] <- 2

nwmem\_df$TitleLevel[staff] <- 2

nwmem\_df$TitleLevel[ass] <- 2

nwmem\_df$TitleLevel[tech] <- 2

nwmem\_df$TitleLevel[research] <- 2

nwmem\_df$TitleLevel[dev] <- 2

nwmem\_df$TitleLevel[intern] <- 1

nwmem\_df$TitleLevel[student]<- 1

nwmem\_df$TitleLevel[train] <- 1

# 0: everything else