**Data Mining CSE 5334 Fall 2016**

**Name of Assignment: Assignment 5**

* Nursery Database was derived from a hierarchical decision model. Originally developed to rank applications for nursery schools. The final decision depended on three sub problems:
* occupation of parents and child's nursery, family structure and
* financial standing, and social and health picture of the family.
* The model was developed within expert system shell for decision making DEX
* The Nursery Database contains examples with the structural information rem oved, i.e., directly relates NURSERY to the eight input attributes: parents, has\_nurs, form, children, housing, finance, social, health.
* **Number of Instances:** 12960 (instances completely cover the attribute space)
* **Number of Attributes: 8**
* Attribute Values:

**Parents** : usual, pretentious, great\_pret

**Has\_nurs** : proper, less proper, improper, critical, very\_crit

**Form** : complete, completed, incomplete, foster

**Children** : 1, 2, 3, more

**Housing** : convenient, less\_conv, critical

**Finance** : convenient, inconv

**Social** : non-prob, slightly\_prob, problematic

**Health** : recommended, priority, not\_recom

Missing Attribute Values: none

Class Distribution (number of instances per class)

**class N N [%]**

-------------------------------------------------------

not\_recom 4320 (33.333 %)

recommend 2 ( 0.015 %)

very\_recom 328 ( 2.531 %)

priority 4266 (32.917 %)

spec\_prior 4044 (31.204 %)

>nursery <- read.csv ("F:/Data mining/Programming assignment/nursery.data", header=FALSE)

> View(nursery)

>colnames(nursery)[1]='Parents'

>colnames(nursery)[2]='has\_nurs'

>colnames(nursery)[2]='Has\_nurs'

>colnames(nursery)[3]='Form'

>colnames(nursery)[4]='Children'

>colnames(nursery)[5]='Housing'

>colnames(nursery)[6]='Finance'

>colnames(nursery)[7]='Social'

>colnames(nursery)[8]='Health'

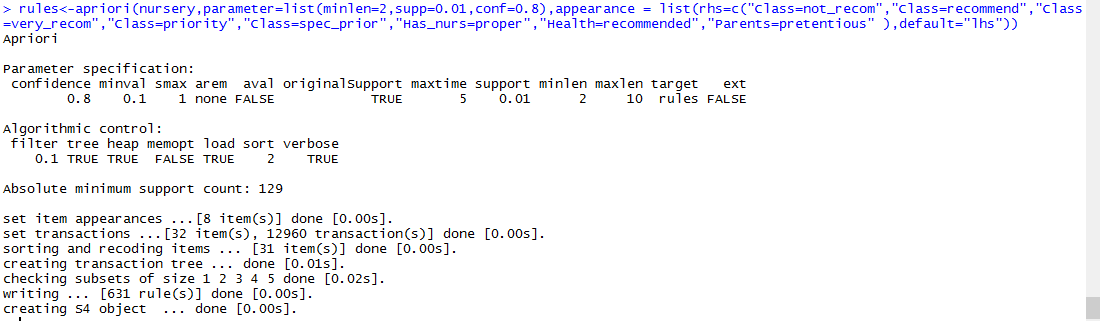
>colnames(nursery)[9]='Class'

>summary(nursery)

>library(arules)

**1] Apriori rules:**

>rules<-apriori(nursery,parameter=list(minlen=2,supp=0.01,conf=0.8),appearance= list(rhs=c("Class=not\_recom","Class=recommend","Class=very\_recom","Class=priority","Class=spec\_prior","Has\_nurs=proper","Health=recommended","Parents=pretentious" ),default="lhs"))



>rules.sorted<-sort(rules,by="lift")

>inspect(rules.sorted)

**2] Show redundant rules**

**#Removing redundant rules**

**# 1.find redundant rules**

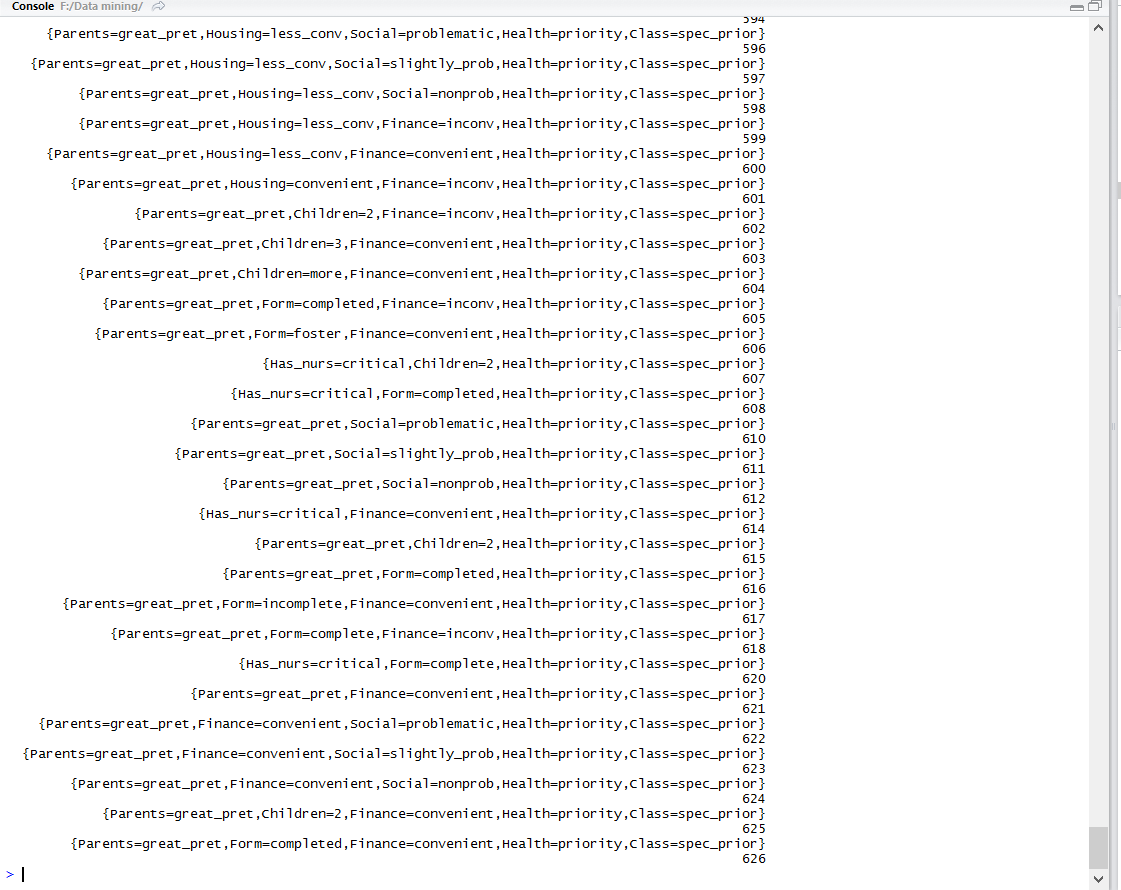
>subset.matrix<-is.subset(rules.sorted,rules.sorted)

>subset.matrix[lower.tri(subset.matrix,diag=T)]<-NA

>redundant<-colSums(subset.matrix,na.rm = T)>=1

**# 2.which rules are redunant**

>which(redundant)



>rdundant\_rules<-which(redundant)

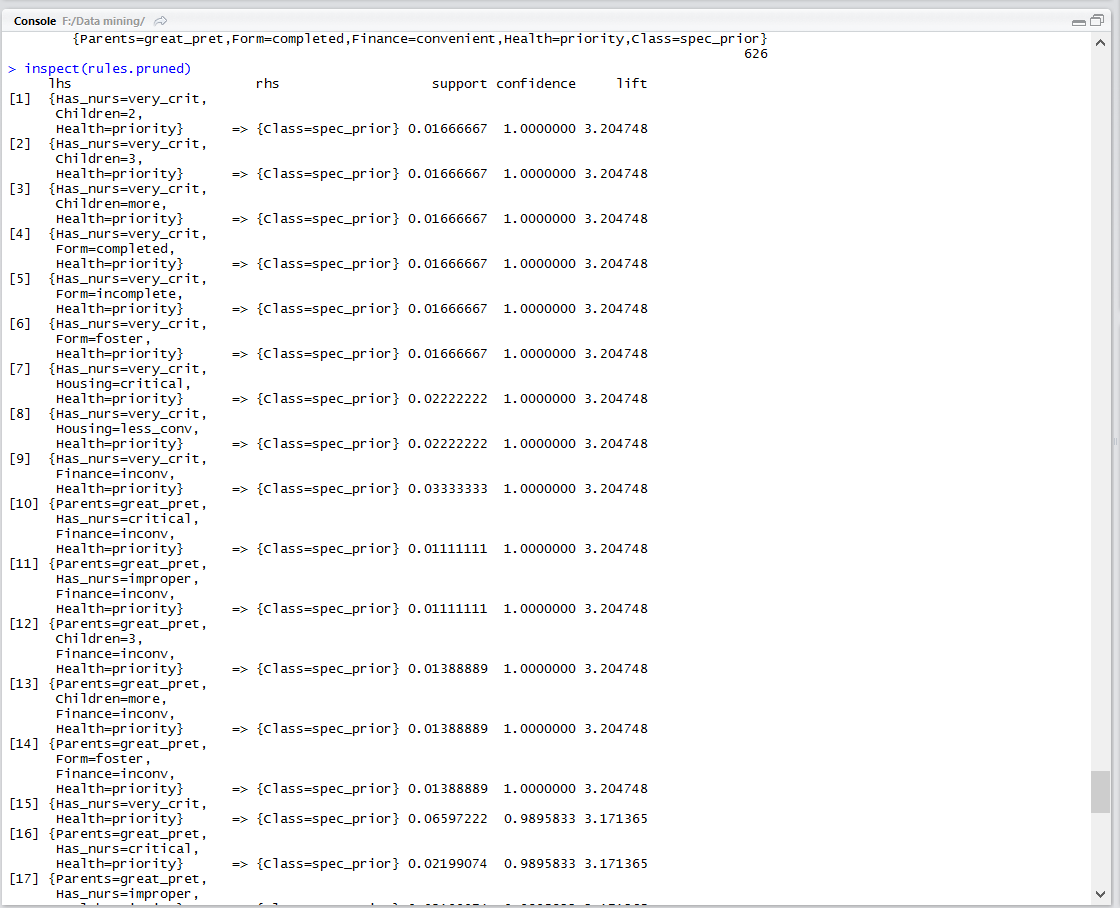
>write.xlsx(rdundant\_rules, "F:/Data mining/redundant\_rules.xlsx")

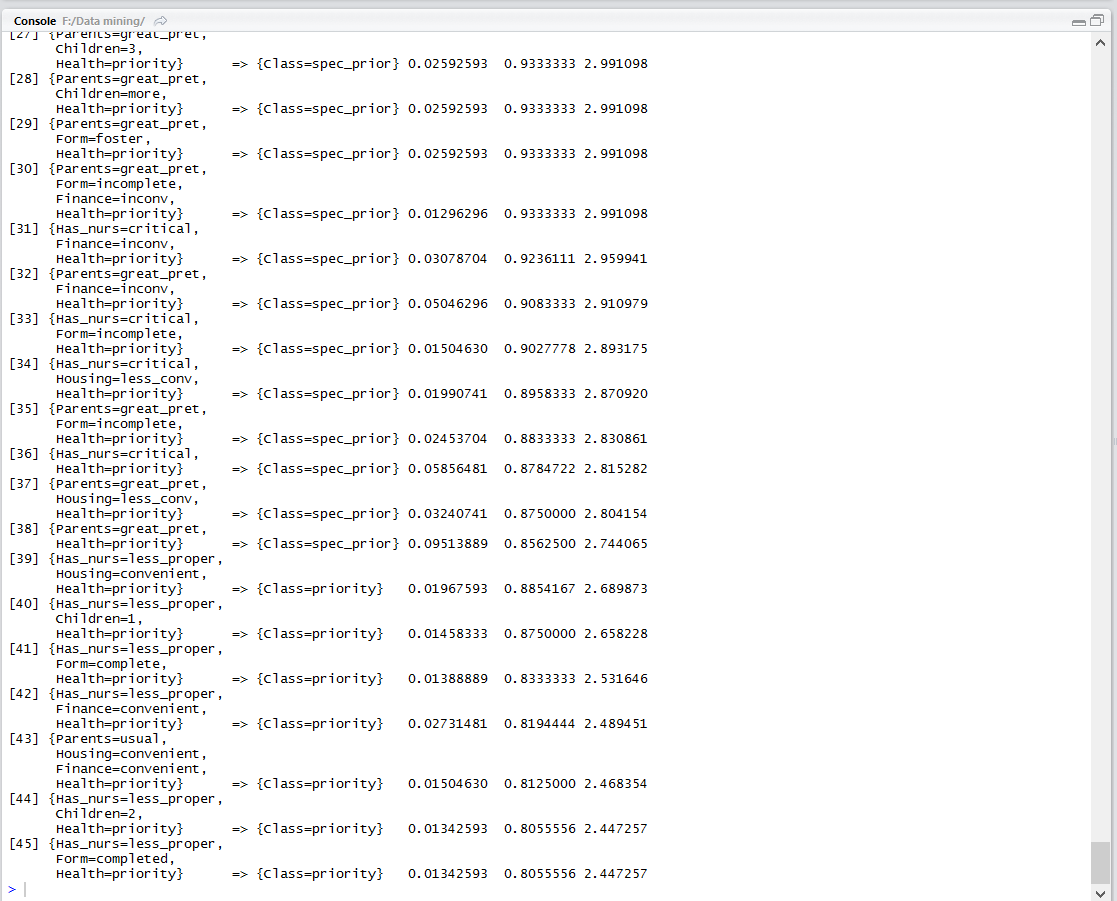
**3] Remove redudant rules**

**# 3.Remove redundant rules**

>rules.pruned<-rules.sorted[!redundant]

>inspect(rules.pruned)



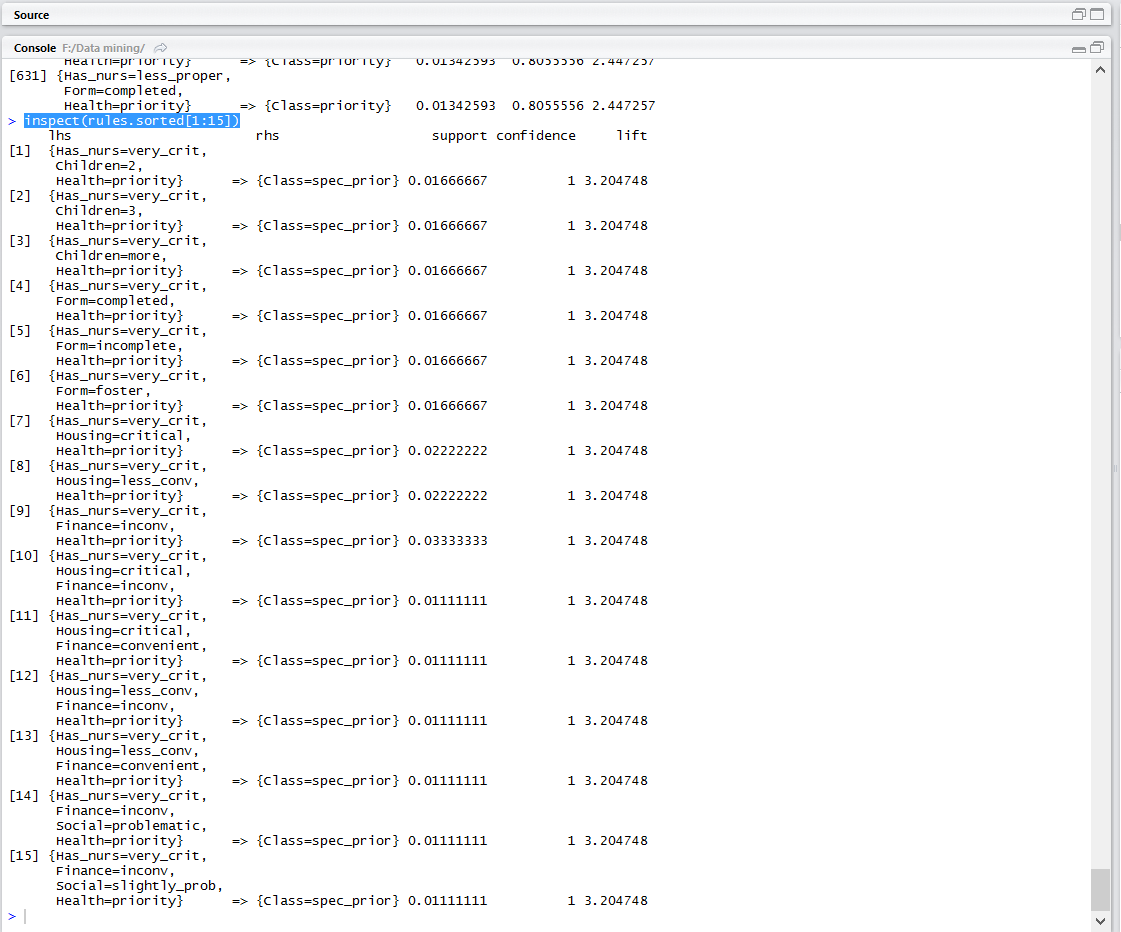


>rules\_pruned<-as(rules.pruned,"data.frame")

>write.xlsx(rules\_pruned, "F:/Data mining/pruned\_rules.xlsx")

**4] Interpret your results**

>inspect(rules.sorted[1:15])



Rule 2: {Has\_nurs=very\_crit,Health=priority} => {Class=spec\_prior}

Rule 10: {Has\_nurs=very\_crit,Housing=critical,Finance=inconv,Health=priority,Class=spec\_prior}

Rule 11: {Has\_nurs=very\_crit,Housing=critical,Finance=convenient,Health=priority,Class=spec\_prior}

Rule 12: {Has\_nurs=very\_crit,Housing=less\_conv,Finance=inconv,Health=priority,Class=spec\_prior}

Rule 13: {Has\_nurs=very\_crit,Housing=less\_conv,Finance=convenient,Health=priority,Class=spec\_prior}

Rule 14: {Has\_nurs=very\_crit,Finance=inconv,Social=problematic,Health=priority,Class=spec\_prior}

Rule 15: {Has\_nurs=very\_crit,Finance=inconv,Social=slightly\_prob,Health=priority,Class=spec\_prior}

**Rules #10 to Rule #15 provide no extra knowledge in addition to rule #2, since rule #2 tells us that all Class= spec\_prior.**

**When a rule (such as #10) is a super rule of another rule (#2) as above, and the former has the same or a lower lift, the former rule (#10, #11, #12, #13, #14, #15) are redundant. The same or a lower lift, the former rule.**

**Rule 2 is {Has\_nurs=very\_crit,Health=priority} , and all these attributes are there in other rules, therefore other rules become redundant.**

**Hence, it can be concluded that redundant data rules can be removed from the data set, as rules that are obtained after pruning will serve the purpose.**

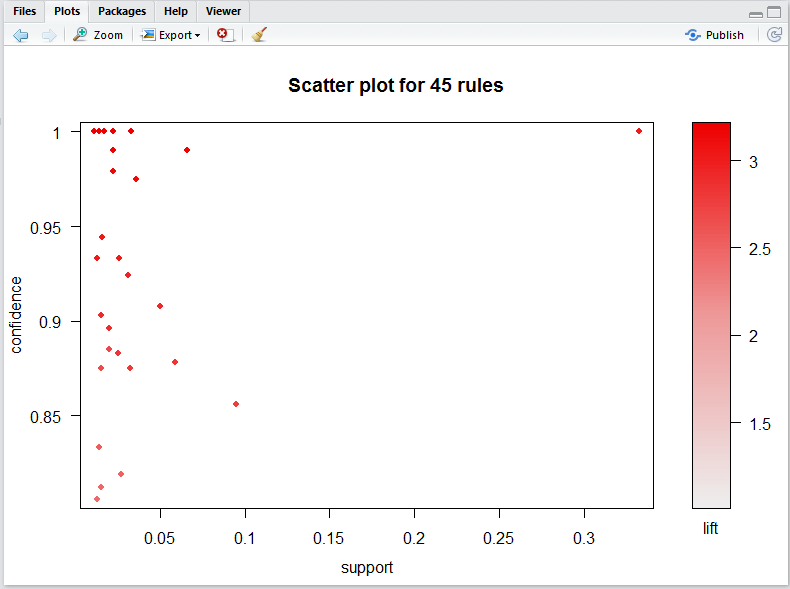
**Total redundant rules generated:585**

**Rules got after pruning the data :45**

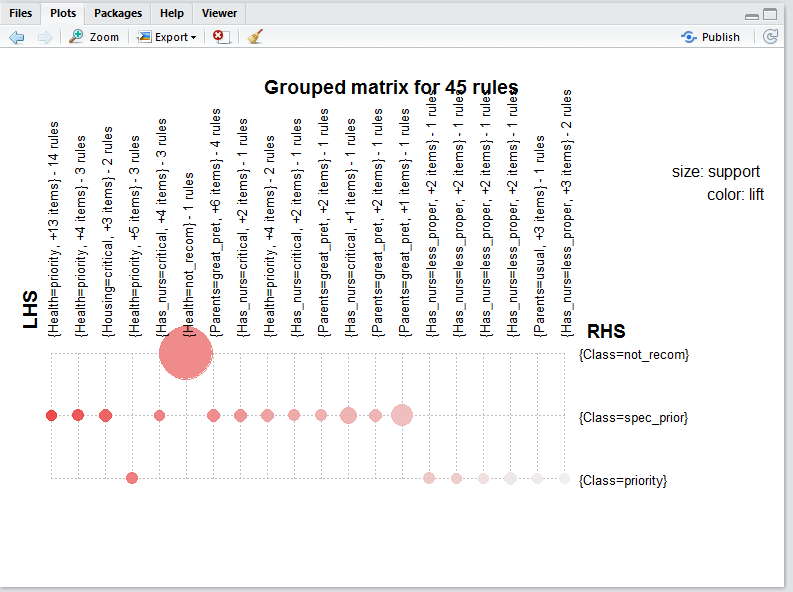
**5] Graphs:**

>library(arulesViz)

> plot(rules.pruned)



> plot(rules.pruned,method="grouped")



> plot(rules.pruned,method="paracoord",control = list(recorder=TRUE))

