Project 4 notebook

Meenakshi Nagarajan

Nov 3, 2017

Dataset – Occupancy detection

https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+

Background of data

The dataset used in this study is obtained from the UCI Machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+). Used to find out the occupancy status.

Dataset Characteristics: Multivariate, Time-Series

Attribute characteristics: Real

Date Donated: 2016/ 02/29

Number of instances: 20560

Number of Attributes: 7

Missing values: N/A

Attributes

date time year-month-day hour:minute:second

Temperature, in Celsius

Relative Humidity, %

Light, in Lux

CO2, in ppm

Humidity Ratio, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air

Occupancy, 0 or 1, 0 for not occupied, 1 for occupied status

```
#Meenakshi Nagarajan
#nagarajan.12@wright.edu
library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

#Load the data into 'df.occupancy'
df.occupancy <- read.csv("/Users/meenakshinagarajan/Desktop/Datamining/Occupa
ncy.csv", header=TRUE)
df.occupancy$date <- NULL
str(df.occupancy)

## 'data.frame':    8143 obs. of  6 variables:
##  $ Temperature  : num  23.2 23.1 23.1 23.1 23.1 ...
##  $ Humidity     : num  27.3 27.3 27.2 27.2 27.2 ...
##  $ Light        : num  426 430 426 426 426 ...
##  $ CO2          : num  721 714 714 708 704 ...
##  $ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
##  $ Occupancy    : int  1 1 1 1 1 1 1 1 1 1 ...
```

The data frame says all are numerical attributes. Therefore the data needs some preparation before transforming this to transaction data.

Identify levels to convert numerical variables into factors

```
for(i in 1:5){
wfact=cut(df.occupancy[,i],pretty(df.occupancy[,i],3))
print(colnames(df.occupancy)[i])
print(table(wfact))
}

## [1] "Temperature"
## wfact
## (19,20] (20,21] (21,22] (22,23] (23,24]
##    2728    2701    1632    1006      71
## [1] "Humidity"
## wfact
## (10,20] (20,30] (30,40]
##    1983    4004    2156
## [1] "Light"
## wfact
##         (0,500]     (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03]
```

```
##            2733             248               1               1
## [1] "CO2"
## wfact
##         (0,500]     (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03]
##            5566            1603             750             183
## (2e+03,2.5e+03]
##              41
## [1] "HumidityRatio"
## wfact
## (0.002,0.003] (0.003,0.004] (0.004,0.005] (0.005,0.006] (0.006,0.007]
##          1641          3274          2512           525           191
```

Divide the variables into categories

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.4.2
```

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

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

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

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

```
df.occupancy[[ "Temperature"]] <- ordered(cut(df.occupancy[[ "Temperature"]],
c(19,20,21,22,23,24)),labels = c("Very Low","Low","Medium","High","Very-High"
))
head(df.occupancy$Temperature)
```

```
## [1] Very-High Very-High Very-High Very-High Very-High Very-High
## Levels: Very Low < Low < Medium < High < Very-High
```

```
df.occupancy[[ "Humidity"]] <- ordered(cut(df.occupancy[[ "Humidity"]], c(10,
20,30,40)),labels = c("Low","Medium","High"))
head(df.occupancy$Humidity)
```

```
## [1] Medium Medium Medium Medium Medium Medium
## Levels: Low < Medium < High
```

```
df.occupancy[[ "HumidityRatio"]] <- ordered(cut(df.occupancy[[ "HumidityRatio
"]], c(0.002,0.003,0.004,0.005,0.006,0.007)),labels = c("Very Low","Low","Med
ium","High","Very-High"))
df.occupancy[[ "Light"]] <- ordered(cut(df.occupancy[[ "Light"]], c(0,500,1e+
03,1.5e+03,2e+03)),labels = c("Max light","Min light","Medium","Very Low"))
df.occupancy[[ "CO2"]] <- ordered(cut(df.occupancy[[ "CO2"]], c(0,500,1e+03,1
```

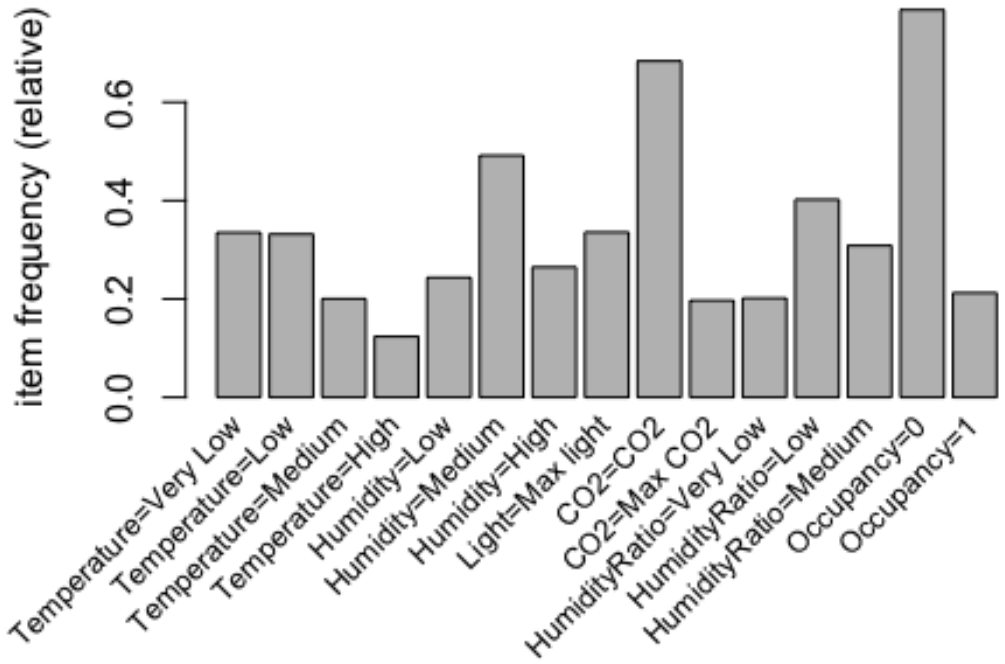
```
.5e+03,2e+03,2.5e+03)),labels = c("CO2","Max CO2","Medium CO2","Low CO2","Ver
y Low"))
head(df.occupancy$HumidityRatio)

## [1] Medium Medium Medium Medium Medium Medium
## Levels: Very Low < Low < Medium < High < Very-High

df.occupancy$Occupancy<-as.factor(df.occupancy$Occupancy)
```

Coercing into transactions

```
Occupancy<-as(df.occupancy,"transactions")
```

Plot to display most important items

```
itemFrequencyPlot(Occupancy, support = 0.1, cex.names=0.8)
```



Find all the rules with minimum support of 1% and confidence of 0.6

```
rules <- apriori(Occupancy,parameter = list(support = 0.01, confidence = 0.6)
)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 81
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[24 item(s), 8143 transaction(s)] done [0.00s].
## sorting and recoding items ... [20 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [1109 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Rules for not occupied and occupied with lift measure greater than 1

```
rulesNotOccupied<-subset(rules,subset=rhs %in% "Occupancy=0" & lift>1)
rulesOccupied<-subset(rules,subset=rhs %in% "Occupancy=1" & lift>1)
```

Compare rules for both sets with highest confidence

```
inspect(head(rulesNotOccupied,n=3,by="confidence"))
```

```
##     lhs                          rhs              support confidence     li
ft count
## [1] {Temperature=High,
##      HumidityRatio=Very Low} => {Occupancy=0} 0.01940317          1 1.2695
67   158
## [2] {Temperature=High,
##      Humidity=Low}           => {Occupancy=0} 0.02456097          1 1.2695
67   200
## [3] {Temperature=High,
##      CO2=CO2}                => {Occupancy=0} 0.02456097          1 1.2695
67   200
```

```
inspect(head(rulesOccupied,n=3,by="confidence"))
```

```
##     lhs                           rhs              support confidence     l
ift count
## [1] {Light=Max light,
##      CO2=Low CO2}              => {Occupancy=1} 0.01891195          1 4.709
659   154
```

```
## [2] {Light=Max light,
##      HumidityRatio=Very-High} => {Occupancy=1} 0.02149085          1 4.709
659   175
## [3] {Light=Max light,
##      CO2=Low CO2,
##      HumidityRatio=Very-High} => {Occupancy=1} 0.01645585          1 4.709
659   134
```

From the rules, we see that, when there is Max light and Low co2, the occupancy status is 1 and the status is 0 when the temperature is high