Exercise 6

Matthew McAvoy

September 13, 2016

Tried loading .arff fie, but was unsuccessful. Answered questions by using csv file.

```
url <- "https://vincentarelbundock.github.io/Rdatasets/csv/texmex/liver.csv"
liver_data <- read.csv(url, header=TRUE)
```

To Discretize on a feature, I will use ALT.M

To discretize on an equal number of observations, we can use below.

```
disALT.M <- with(liver_data, cut(ALT.M, breaks=quantile(ALT.M, probs=seq(0, 1, by=0.20)), include.lowes
liver_data$disALT.M <- disALT.M
summary(liver_data$disALT.M)
```

```
##   [2,12]  (12,16]  (16,19]  (19,25] (25,324]
##      139      152       88      106      121
```

To discretize on equal length, we can use below.

```
diseALT.M <- with(liver_data, cut(ALT.M, 5, include.lowest=TRUE))
liver_data$diseALT.M <- diseALT.M
summary(liver_data$diseALT.M)
```

```
## [1.68,66.4]  (66.4,131]   (131,195]   (195,260]   (260,324]
##         599           4           0           1           2
```

To use a topdown approach, we need to load the discretization package and then try it with a discretization algorithm. The Method I use is CAIM (class-attribute interdepedence maximization) to achieve the lowest number of intervals and the highest class-atribute interdependency. [1]

[1] Kurgan L., Cios K. CAIM Discretization Algorithm. Uni of Colorado at Denver. https://ai2-s2-pdfs.s3.amazonaws.com/4c2f/b94f58af8d18a7d80f8dd384957718806ae5.pdf

```
library(discretization)
topdown(liver_data, method = 1)
```

```
## Warning in is.na(r): is.na() applied to non-(list or vector) of type 'NULL'

## Warning in Ops.factor(xo[ci], xo[ci + 1]): '+' not meaningful for factors

## Warning in is.na(r): is.na() applied to non-(list or vector) of type 'NULL'

## Warning in Ops.factor(xo[ci], xo[ci + 1]): '+' not meaningful for factors
```

```
## [[1]]
## [1]   1.0 482.5 483.5 514.5 606.0
##
## [[2]]
## [1]  15.0  65.5 120.0 126.0 129.0
##
## [[3]]
## [1]   4.0  19.5  22.5  23.5 198.0
##
## [[4]]
## [1]   5.0  14.5  15.5  16.5 104.0
##
## [[5]]
## [1]  2.7360  7.4385 22.8285 23.0850 27.5310
##
## [[6]]
## [1]   1.0 171.5 212.0 293.0 341.0
##
## [[7]]
## [1]   2.0  68.5 148.5 236.5 324.0
##
## [[8]]
## [1]   6.0  40.5  91.5 133.5 250.0
##
## [[9]]
## [1]  3.2490 13.4235 29.2410 32.6610 42.7500
##
## [[10]]
## [1] 1 4
##
## [[11]]
## [1] 1 5
```

This discretized the whole data. Looking at item 7 which corresponds to ALT.M, we see it separated similarly to how discretization on equal length output.

Unfortunately, there doesn't seem to be a package to perform a bottomup discretization.