****DoRequest1.m**

**T.C.**

**SELÇUK ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

Uygulama Ödevi

*Bilgisayar Mühendisliği Bölümü*

*Computer Engineering Department*

*Veri Ön İşleme Teknikleri Dersi*

*Data Preprocessing Techniques CLASS*

*100% new algorithms*

*100% yeni algoritmalar*

*26.5.2017*

Hazırlayan: M. Abobakr Almostafa  
Okutman: Doç.Dr. Halife KODAZ

2016-2017 Bahar donemi

filename = 'bands.dat';

T = readtable(filename);

[**T**](file:///C:\Program%20Files\MATLAB\R2017a\help\matlab\ref\readtable.html?browser=F1help#outputarg_T?browser=F1help)**= readtable(**[**filename**](file:///C:\Program%20Files\MATLAB\R2017a\help\matlab\ref\readtable.html?browser=F1help#inputarg_filename?browser=F1help)**) creates a table by reading column oriented data from a file.**

**readtable determines the file format from the file extension:**

* **.txt, .dat, or .csv for delimited text files**
* **.xls, .xlsb, .xlsm, .xlsx, .xltm, .xltx, or .ods for spreadsheet files**

**readtable creates one variable in T for each column in the file and reads variable names from the first row of the file. By default, the variables created are double when the entire column is numeric, or cell arrays of character vectors when any element in a column is not numeric.**

for i=9:12

**here for the requested attribute we do the same calculations**

% take the attribute and covert to array

disp(['For Attribute Number ',num2str(i)])

disp('-------------------------------------------------------');

att = T(:,i);

X = table2array(att);

**A = table2array(**[**T**](file:///C:\Program%20Files\MATLAB\R2017a\help\matlab\ref\table2array.html?browser=F1help#inputarg_T)**) converts the table, T, to a homogeneous array, A.**

% Fisrt we see if we have outliers

disp('Fisrt we see if we have outliers')

X = sort(X);

Q1 = ClacQuartile(X,25);

Q3 = ClacQuartile(X,75);

IQR = Q3 - Q1;

LF = Q1- 1.5\* IQR; % LowerFence

UF = Q3+ 1.5\* IQR; % UpperFence

OutLiers = X(X<LF | X > UF)

**Calculating outliers to remove them from the data is very important to avoid unreasonable results**

% remove outliers if exist

if size(OutLiers,1) ~= 0

X = X(X>=LF & X <= UF);

end

disp('Outliers are removed')

For Attribute Number 9

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

Fisrt we see if we have outliers

OutLiers =

0

900

Outliers are removed

%Five Number Summary

disp('Five Number Summary')

Min = min(X)

Q1

Med = median(X)

Q3

Max = max(X)

**Already defined statistical functions were used because of the simple implementation of them.**

Five Number Summary

Min =

1000

Q1 =

1640

Med =

1800

Q3 =

2100

Max =

2600

disp('Other statistical measures')

Mean = mean(X)

Mode = mode(X)

IQR

Variance = var(X)

StandardDeviation = std(X)

**Again, they are easy to calculate.**

ther statistical measures

Mean =

1.8589e+03

Mode =

1800

IQR =

460

Variance =

9.6142e+04

StandardDeviation =

310.0670

figure

boxplot(X);

**As an example we show the result of the first two attributes for boxplot**

title(['For Attribute Number ',num2str(i)])

end



**DoRequest2.m**

filename = 'bands.dat';

T = readtable(filename);

for i=9:12

figure

% take the attribute and covert to array

disp(['For Attribute Number ',num2str(i)])

disp('-------------------------------------------------------');

att = T(:,i);

X = table2array(att);

% Fisrt we see if we have outliers

disp('Fisrt we see if we have outliers')

X = sort(X);

Q1 = ClacQuartile(X,25);

Q3 = ClacQuartile(X,75);

IQR = Q3 - Q1;

LF = Q1- 1.5\* IQR; % LowerFence

UF = Q3+ 1.5\* IQR; % UpperFence

OutLiers = X(X<LF | X > UF)

% remove outliers if exist

if size(OutLiers,1) ~= 0

X = X(X>=LF & X <= UF);

end

disp('Outliers are removed')

% min-max normalizasyon

Min = min(X);

Max = max(X);

New\_Min = 0;

New\_Max = 1;

X\_MinMax = ((X - Min )/ (Max - Min) ) \* (New\_Max- New\_Min) + New\_Min;

**Here we use the rule**

disp('min-max normalization is Done ');

disp(X\_MinMax');

For Attribute Number 9

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

Fisrt we see if we have outliers

OutLiers =

0

900

Outliers are removed

min-max normalization is Done

Columns 1 through 6

0 0.0625 0.1250 0.1563 0.1563 0.1563

Mean = mean(X);

SD = std(X);

X\_ZScore = (X-Mean)/SD;

disp('Z-Score normalization is Done ');

disp(X\_ZScore')

disp(X\_ZScore')

Columns 1 through 6

-0.6059 -0.6059 -0.6059 -0.6059 -0.6059 -0.6059

Columns 7 through 12

-0.6059 -0.6059 -0.6059 -0.6059 -0.6059 -0.6059

disp('Applaying n equal-width ...');

disp('Number of Bins and width ')

n = 5

Width = (Max-Min)/n

Applaying n equal-width ...

Number of Bins and width

n =

5

Width =

320

X = sort(X);

for j = 1 : n

Bin = X (X >= Min+ (j-1)\*Width & X< Min+ j \* Width);

histogram(Bin)

hold on

end

end

the histogram for the first bin here



**DoRequest3.m**

filename = 'bands.dat';

T = readtable(filename);

disp('Claculating Antropy for the classification attribute ...');

att = T(:,20);

XT = table2array(att);

XT = string(XT);

**We convert to string so we can compare the class values**

N = size(XT,1);

count1 = size( find(XT == 'band'),1);

prop1 = (count1)/N;

count2 = size( find(XT == 'noband'),1);

prop2 = (count2)/N;

HT = - prop1 \* log2(prop1) - prop2 \* log2(prop2);

**Since we have two clases**

disp(' Antropy for the classification attribute H(T)');

HT

Claculating Antropy for the classification attribute ...

Antropy for the classification attribute H(T)

HT =

0.9506

for i=9:12

HNew\_X = 0; %the antropy for the current attribute ... All Bin Mean Values

% take the attribute and covert to array

disp(['For Attribute Number ',num2str(i)])

disp('-------------------------------------------------------');

att = T(:,i);

X = table2array(att);

% Fisrt we see if we have outliers

disp('Fisrt we see if we have outliers')

X = sort(X);

Q1 = ClacQuartile(X,25);

Q3 = ClacQuartile(X,75);

IQR = Q3 - Q1;

LF = Q1- 1.5\* IQR; % LowerFence

UF = Q3+ 1.5\* IQR; % UpperFence

OutLiers = X(X<LF | X > UF)

% remove outliers if exist

if size(OutLiers,1) ~= 0

X = X(X>=LF & X <= UF);

end

disp('Outliers are removed')

For Attribute Number 9

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

Fisrt we see if we have outliers

OutLiers =

0

900

Outliers are removed

New\_X = table2array(att);

if size(OutLiers,1) ~= 0

New\_X = New\_X(New\_X>=LF & New\_X <= UF);

end

disp('Applaying n equal-width ...');

disp('Number of Bins and width ')

n = 3

Min = min(X);

Max = max(X);

Width = (Max-Min)/n

for j = 1 : n

Bin = X (X >= Min+ (j-1)\*Width & X< Min+ j \* Width);

disp('Smoothing By Mean')

Mean = mean (Bin);

indexs = find(ismember( New\_X,Bin));

New\_X(indexs) = Mean;

Applaying n equal-width ...

Number of Bins and width

n =

3

Width =

533.3333

Smoothing By Mean

disp('Calculating the antropy for the generated categorical value ... Bin Mean Value');

count1 = size( find(XT(indexs) == 'band'),1);

BinN = size(Bin,1);

prop1 = (count1)/BinN;

count2 = size( find(XT(indexs) == 'noband'),1);

prop2 = (count2)/BinN;

HBin = - prop1 \* log2(prop1) - prop2 \* log2(prop2)

HBin =

0.9885

%Calculating the antropy for the current attribute ... All Bin Mean Values

prop = BinN/N;

HNew\_X = HNew\_X + prop \* HBin;

end

disp('Calculating the antropy for the current attribute ... All Bin Mean Values');

HNew\_X

HNew\_X =

0.9172

disp('Calculating the GAIN for the current attribute ...');

Gain = HT - HNew\_X

Gain =

0.0334

end

**the same operation is repeated here but for n = 4**

**number of bins as it is requested in the homework**

for i=9:12

HNew\_X = 0; %the antropy for the current attribute ... All Bin Mean Values

% take the attribute and covert to array

disp(['For Attribute Number ',num2str(i)])

disp('-------------------------------------------------------');

att = T(:,i);

X = table2array(att);

% Fisrt we see if we have outliers

disp('Fisrt we see if we have outliers')

X = sort(X);

Q1 = ClacQuartile(X,25);

Q3 = ClacQuartile(X,75);

IQR = Q3 - Q1;

LF = Q1- 1.5\* IQR; % LowerFence

UF = Q3+ 1.5\* IQR; % UpperFence

OutLiers = X(X<LF | X > UF)

% remove outliers if exist

if size(OutLiers,1) ~= 0

X = X(X>=LF & X <= UF);

end

disp('Outliers are removed')

New\_X = table2array(att);

if size(OutLiers,1) ~= 0

New\_X = New\_X(New\_X>=LF & New\_X <= UF);

end

disp('Applaying n equal-width ...');

disp('Number of Bins and width ')

n = 4

Min = min(X);

Max = max(X);

Width = (Max-Min)/n

for j = 1 : n

Bin = X (X >= Min+ (j-1)\*Width & X< Min+ j \* Width);

disp('Smoothing By Mean')

Mean = mean (Bin);

indexs = find(ismember( New\_X,Bin));

New\_X(indexs) = Mean;

disp('Calculating the antropy for the generated categorical value ... Bin Mean Value');

count1 = size( find(XT(indexs) == 'band'),1);

BinN = size(Bin,1);

prop1 = (count1)/BinN;

count2 = size( find(XT(indexs) == 'noband'),1);

prop2 = (count2)/BinN;

HBin = - prop1 \* log2(prop1) - prop2 \* log2(prop2)

%Calculating the antropy for the current attribute ... All Bin Mean Values

prop = BinN/N;

HNew\_X = HNew\_X + prop \* HBin;

end

disp('Calculating the antropy for the current attribute ... All Bin Mean Values');

HNew\_X

disp('Calculating the GAIN for the current attribute ...');

Gain = HT - HNew\_X

end

**as an example we show the results of the 12th attribute …**

For Attribute Number 12

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

Fisrt we see if we have outliers

OutLiers =

6

6

6

6

6

8

8

8

8

8

8

8

8

10

10

10

10

10

10

10

12

12

12

12

16

Outliers are removed

Applaying n equal-width ...

Number of Bins and width

n =

4

Width =

1.2500

Smoothing By Mean

Calculating the antropy for the generated categorical value ... Bin Mean Value

HBin =

0.9248

Smoothing By Mean

Calculating the antropy for the generated categorical value ... Bin Mean Value

HBin =

0.7793

Smoothing By Mean

Calculating the antropy for the generated categorical value ... Bin Mean Value

HBin =

0.9852

Smoothing By Mean

Calculating the antropy for the generated categorical value ... Bin Mean Value

HBin =

0.8256

Calculating the antropy for the current attribute ... All Bin Mean Values

HNew\_X =

0.8067

Calculating the GAIN for the current attribute ...

Gain =

0.1439