SIR C.R.REDDY COLLEGE OF ENGINEERING, ELURU

Approved by AICTE, New Delhi & permanently affiliated to ANDHRA UNIVERSITY (Affiliated to JNTU Kakinada from the A.Y2017-2018)

DEPARTMENTOFINFORMATION TECHNOLOGY

III/IVB.TECHII SEMESTER

Regulation :: R16

Data Mining Labmanual

R1632126

Experiment 1

1)Aim: Demonstration of preprocessing on dataset student.arff

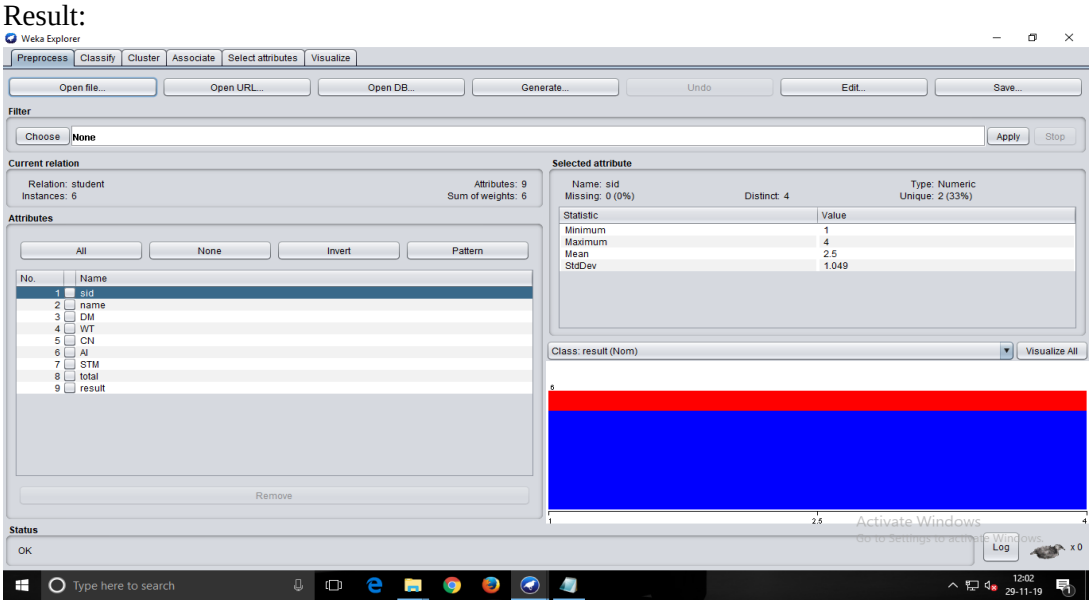
Data set

```
@relation student
@attribute sid numeric
@attribute name {usha,hari,rajesh,kiran,giri,manash}
@attribute DM numeric
@attribute WT numeric
@attribute CN numeric
@attribute AI numeric
@attribute STM numeric
@attribute total numeric
@attribute result {pass,fail}

@data
1,usha,60,55,45,50,40,250,pass
2,hari,60,55,45,40,40,240,pass
3,rajesh,60,55,40,50,40,240,pass
4,kiran,60,55,30,50,40,235,fail
2,giri,60,55,45,60,40,260,pass
3,manash,60,55,65,50,40,270,pass
```

Ex1:

Result:



Add:

NAME

weka.filters.unsupervised.attribute.Add

SYNOPSIS

An instance filter that adds a new attribute to the dataset. The new attribute will contain all missing values.

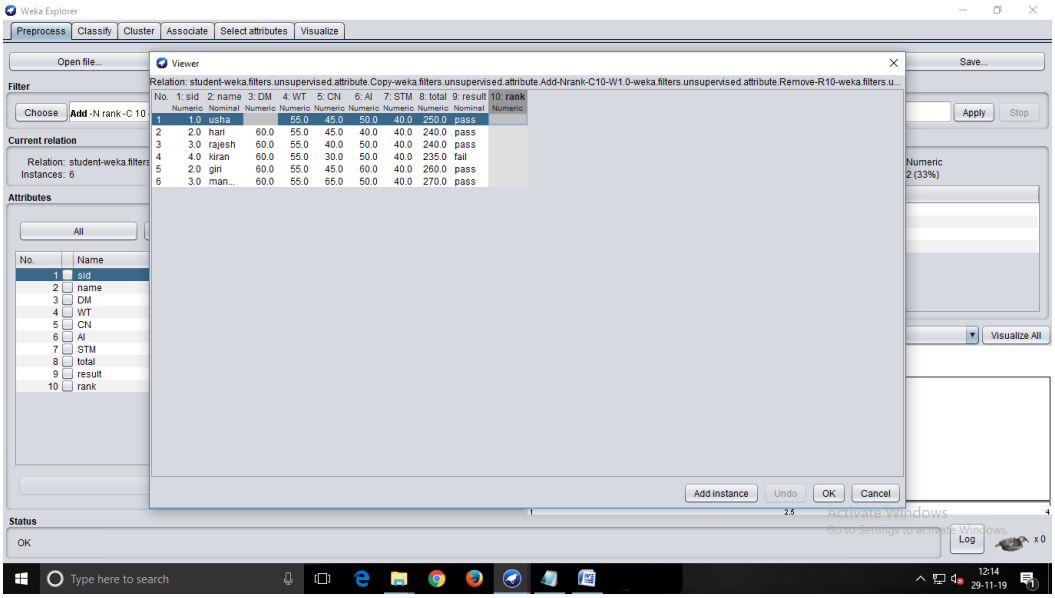
OPTIONS

nominalLabels -- The list of value labels (nominal attribute creation only). The list must be comma-separated, eg: "red,green,blue". If this is empty, the created attribute will be numeric.

debug -- If set to true, filter may output additional info to the console.

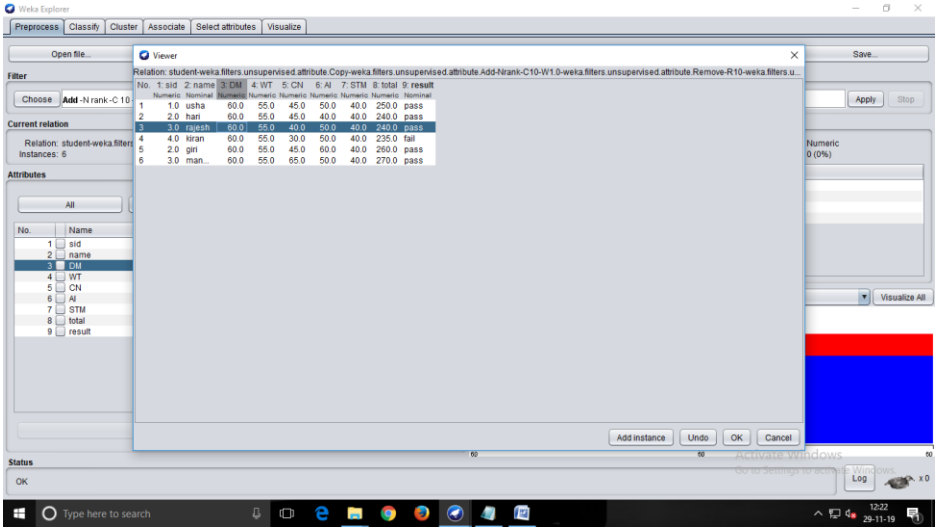
attributeName -- Set the new attribute's name.

attributeIndex -- The position (starting from 1) where the attribute will be inserted (first and last are valid indices).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built.
weight -- The weight for the new attribute.
dateFormat -- The format of the date values (see ISO-8601).
attributeType -- Defines the type of the attribute to generate.

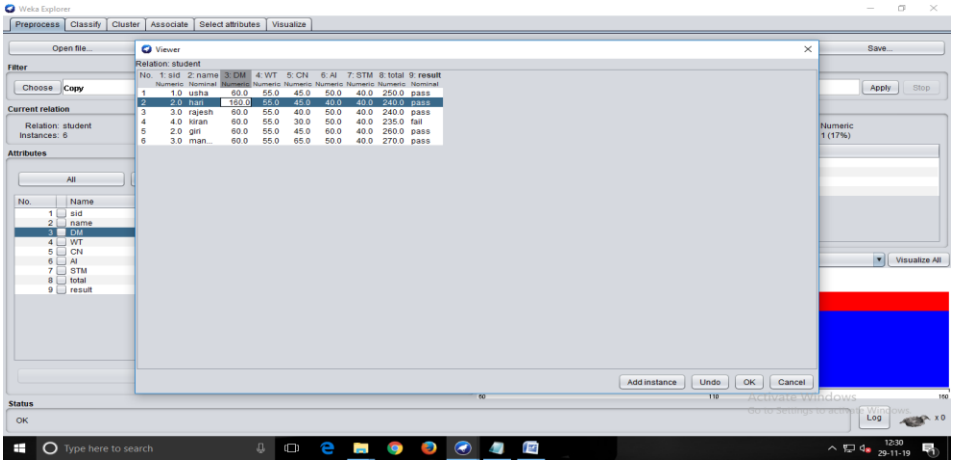


Remove:
NAME
weka.filters.unsupervised.attribute.Remove
SYNOPSIS
A filter that removes a range of attributes from the dataset. Will re-order the remaining attributes if invert matching sense is turned on and the attribute column indices are not specified in ascending order.
OPTIONS
debug -- If set to true, filter may output additional info to the console.
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use with caution to reduce runtime.)
attributeIndices -- Specify range of attributes to act on. This is a comma separated list of attribute indices, with "first" and "last" valid values. Specify an inclusive range with "-". E.g: "first-3,5,6-10,last".
invertSelection -- Determines whether action is to select or delete. If set to true, only the specified attributes will be kept; If set to false, specified attributes will be deleted.

missing attributes:



ErrornicData:

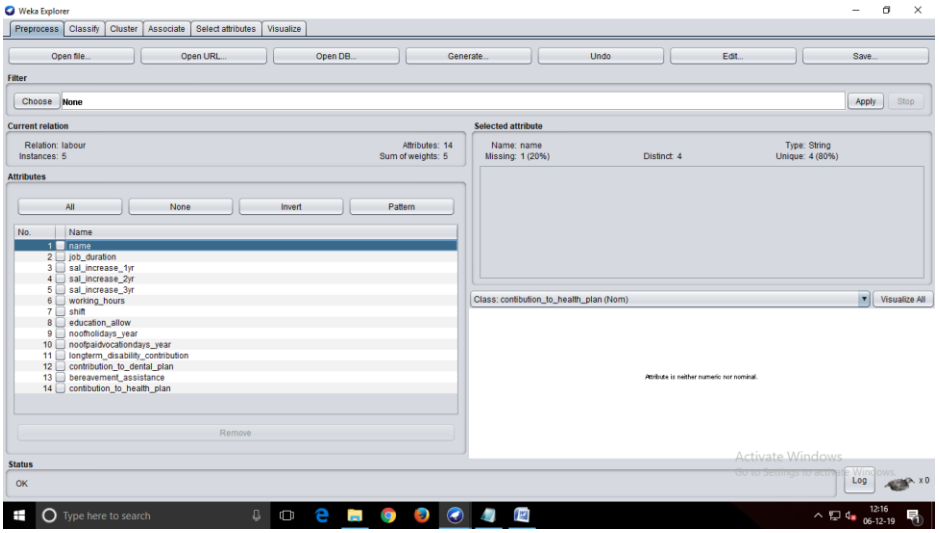


Experiment 2

2)Aim:Demonstration of preprocessing on dataset labor.arff

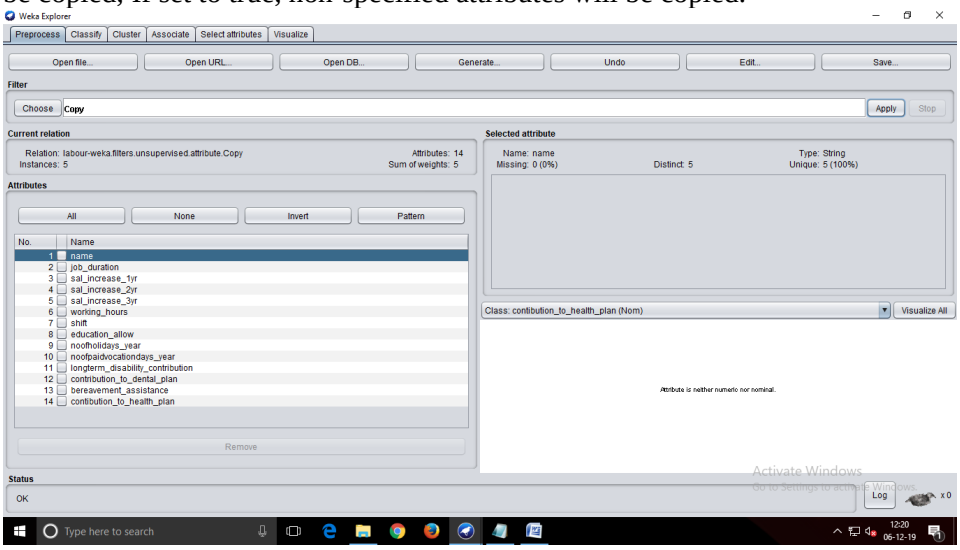
Data set
@relation labour
@attribute name string
@attribute job_duration numeric
@attribute sal_increase_1yr numeric
@attribute sal_increase_2yr numeric
@attribute sal_increase_3yr numeric
@attribute working_hours numeric
@attribute shift {day,night}
@attribute education_allow {yes,no}
@attribute noofholidays_year numeric
@attribute noofpaidvocationdays_year numeric
@attribute longterm_disability_contribution {yes,no}
@attribute contribution_to_dental_plan {none,half,full}
@attribute bereavement_assistance {yes,no}
@attribute contibution_to_health_plan {none,half,full}

@data
?,3,2,2,2,6,day,no,5,5,no,half,no,none
harish,2,2,2,3,5,night,no,5,4,no,none,no,none
ramesh,8,2,3,3,7,day,no,5,5,yes,full,yes,half
suresh,5,2,3,2,6,night,no,5,3,no,half,no,half
Naresh,5,2,2,2,6,day,no,5,4,no,half,no,half
Result:

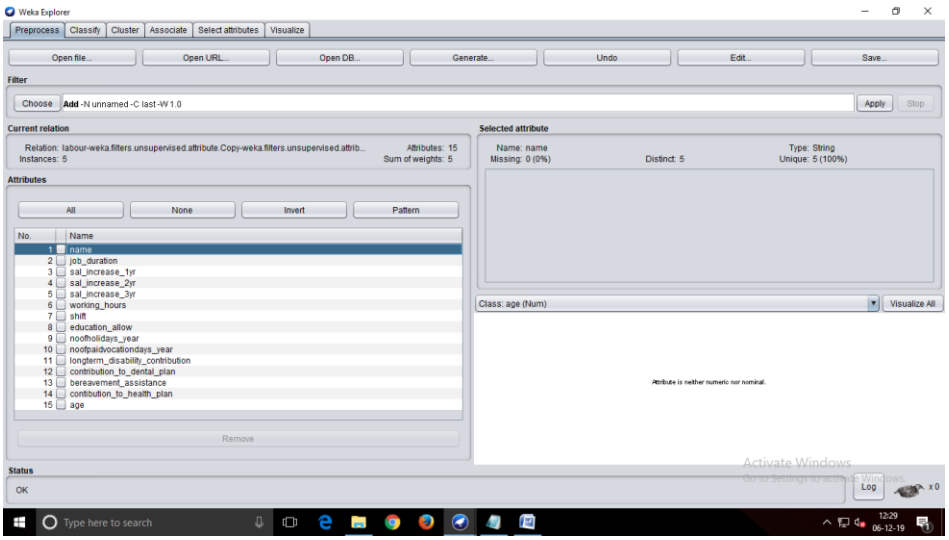


Copy:
NAME
weka.filters.unsupervised.attribute.Copy
SYNOPSIS

An instance filter that copies a range of attributes in the dataset. This is used in conjunction with other filters that overwrite attribute values during the course of their operation -- this filter allows the original attributes to be kept as well as the new attributes.
OPTIONS
debug -- If set to true, filter may output additional info to the console.
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use with caution to reduce runtime.)
attributeIndices -- Specify range of attributes to act on. This is a comma separated list of attribute indices, with "first" and "last" valid values. Specify an inclusive range with "-". E.g: "first-3,5,6-10,last".
invertSelection -- Sets copy selected vs unselected action. If set to false, only the specified attributes will be copied; If set to true, non-specified attributes will be copied.



Add:
NAME
weka.filters.unsupervised.attribute.Add
SYNOPSIS
An instance filter that adds a new attribute to the dataset. The new attribute will contain all missing values.
OPTIONS
nominalLabels -- The list of value labels (nominal attribute creation only). The list must be comma-separated, eg: "red,green,blue". If this is empty, the created attribute will be numeric.
debug -- If set to true, filter may output additional info to the console.
attributeName -- Set the new attribute's name.
attributeIndex -- The position (starting from 1) where the attribute will be inserted (first and last are valid indices).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built.
weight -- The weight for the new attribute.
dateFormat -- The format of the date values (see ISO-8601).
attributeType -- Defines the type of the attribute to generate.



String-to-Nominal:

NAME

weka.filters.unsupervised.attribute.StringToNominal

SYNOPSIS

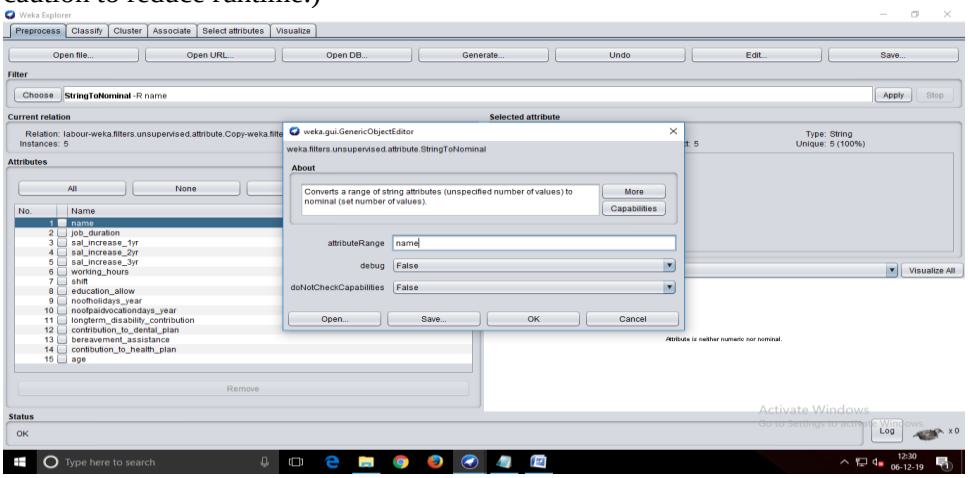
Converts a range of string attributes (unspecified number of values) to nominal (set number of values). You should ensure that all string values that will appear are represented in the first batch of the data.

OPTIONS

debug -- If set to true, filter may output additional info to the console.

attributeRange -- Sets which attributes to process ("first" and "last" are valid values and ranges and lists can also be used).

doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use with caution to reduce runtime.)



Normalize:

NAME

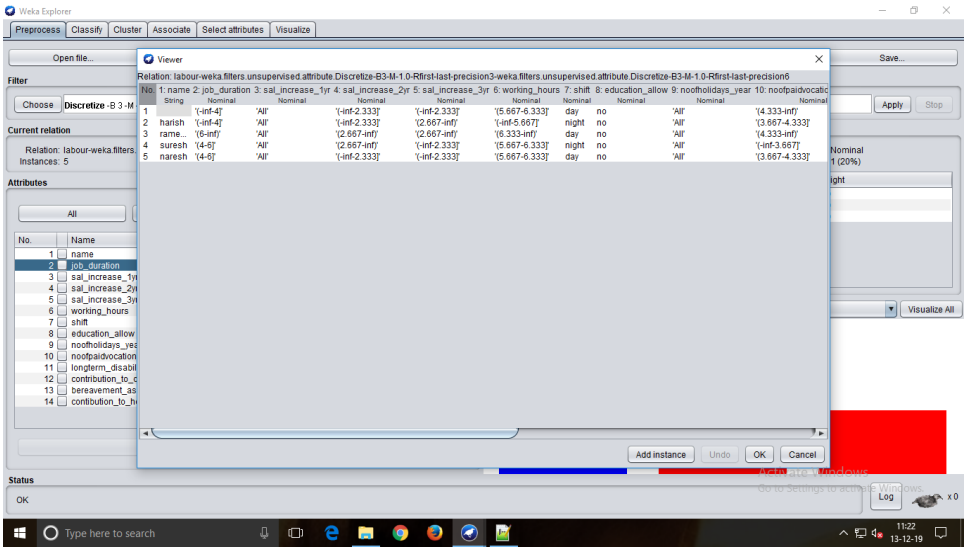
weka.filters.unsupervised.attribute.Normalize

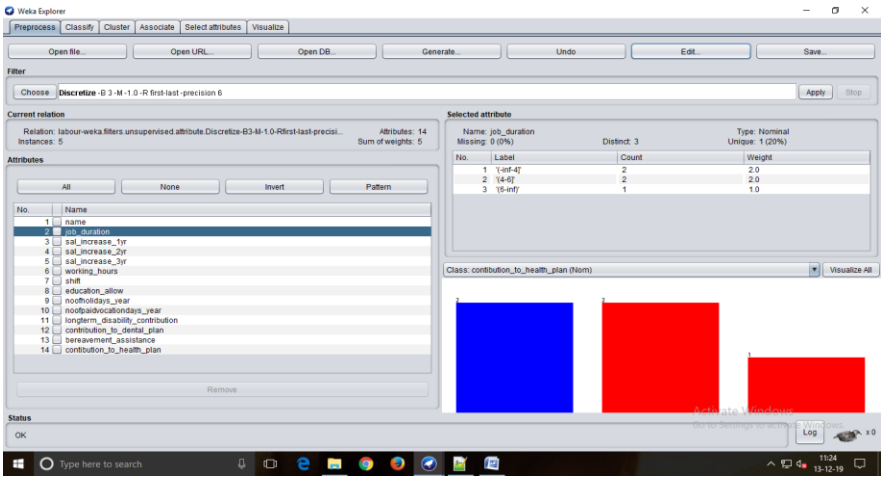
SYNOPSIS

Normalizes all numeric values in the given dataset (apart from the class attribute, if set). By default, the resulting values are in [0,1] for the data used to compute the normalization intervals. But with the scale and translation parameters one can change that, e.g., with scale = 2.0 and translation = -1.0 you get values in the range [-1,+1].

OPTIONS
debug -- If set to true, filter may output additional info to the console.
translation -- The translation of the output range (default: 0).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use with caution to reduce runtime.)
scale -- The factor for scaling the output range (default: 1).
ignoreClass -- The class index will be unset temporarily before the filter is applied.

Discretize:
NAME
weka.filters.unsupervised.attribute.Discretize
SYNOPSIS
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning. Skips the class attribute if set.
OPTIONS
spreadAttributeWeight -- When generating binary attributes, spread weight of old attribute across new attributes. Do not give each new attribute the old weight.
makeBinary -- Make resulting attributes binary.
debug -- If set to true, filter may output additional info to the console.




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
Discretize -B 3 -M -1.0 -R first-last -precision 6
Apply
Stop
Current relation
Relation: labour-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-Rfirst-last-precisi...
Instances: 5
Attributes: 14
Sum of weights: 5
Selected attribute
Name: job_duration
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 1 (20%)
No. Label Count Weight
1 '(-inf-4]' 2 2.0
2 '(4-6]' 2 2.0
3 '(6-inf]' 1 1.0
Attributes
All
None
Invert
Pattern
No. Name
1 name
2 job_duration
3 sal_increase_1yr
4 sal_increase_2yr
5 sal_increase_3yr
6 working_hours
7 shift
8 education_allow
9 noofholidays_year
10 noofpaidvocationdays_year
11 longterm_disability_contribution
12 contribution_to_dental_plan
13 bereavement_assistance
14 contibution_to_health_plan
Remove
Class: contibution_to_health_plan (Nom)
Visualize All
Status
OK
Log
x 0
Type here to search
11:24
15-12-19

Experiment 3

3)Demonstration of Association rule process on dataset contact lenses.arff using apriori algorithm
Data set

```
@relation contact-lenses
@attribute age {young, pre-presbyopic,presbyopic}
@attribute spectacle-prescrip {myope, hypermetrope}
@attribute astigmatim {no,yes}
@attribute tear-prod-rate {reduced, normal}
@attribute contact-lenses {soft,hard,none}
@data
young,myope,no,reduced,none
young,myope,no,normal,soft
young,myope,yes,reduced,none
young,myope,yes,normal,hard
young,hypermetrope,no,reduced,none
young,hypermetrope,no,normal,soft
young,hypermetrope,yes,reduced,none
young,hypermetrope,,yes,normal,hard
pre-presbyopic,myope,no,reduced,none
pre-presbyopic,myope,no,normal,soft
pre-presbyopic,myope,yes,reduced,none
pre-presbyopic,myope,yes,normal,soft
pre-presbyopic,myope,yes,reduced,none
pre-presbyopic,myope,yes,normal,hard
pre-presbyopic,hypermetrope,no,reduced,none
pre-presbyopic,hypermetrope,no,normal,soft
pre-presbyopic,hypermetrope,,yes,reduced,none
pre-presbyopic,hypermetrope,yes,normal,none
presbyopic,myope,no,normal,none
presbyopic,myope,yes,reduced,none
presbyopic,myope,yes,normal,hard
presbyopic,hypermetrope,no,reduced,none
presbyopic,hypermetrope,no,,normal,soft
Presbyopic,hypermetrope,yes,reduced,none
```

NAME
weka.associations.Apriori

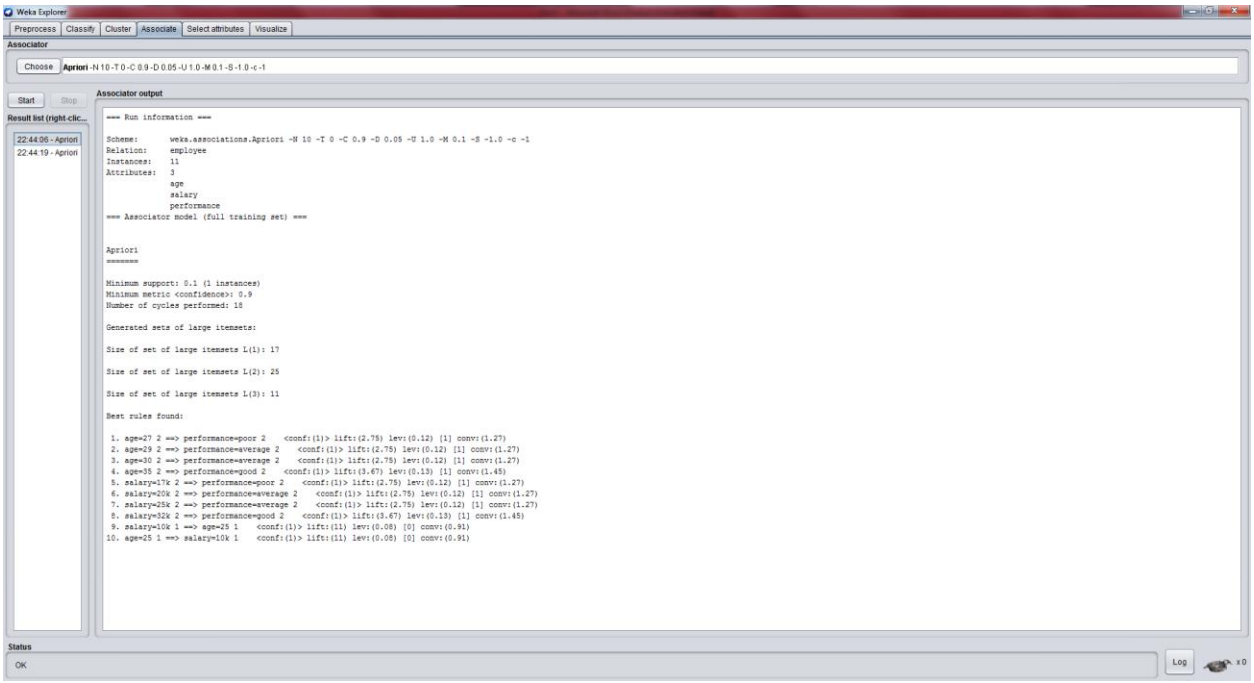
SYNOPSIS
Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence.
The algorithm has an option to mine class association rules. It is adapted as explained in the second reference.

OPTIONS
minMetric -- Minimum metric score. Consider only rules with scores higher than this value.
verbose -- If enabled the algorithm will be run in verbose mode.
numRules -- Number of rules to find.
lowerBoundMinSupport -- Lower bound for minimum support.
classIndex -- Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
outputItemSets -- If enabled the itemsets are output as well.
car -- If enabled class association rules are mined instead of (general) association rules.
doNotCheckCapabilities -- If set, associator capabilities are not checked before associator is built (Use with caution to reduce runtime).
removeAllMissingCols -- Remove columns with all missing values.
significanceLevel -- Significance level. Significance test (confidence metric only).
treatZeroAsMissing -- If enabled, zero (that is, the first value of a nominal) is treated in the same way as a missing value.

Steps:

1. Open the datafile in wekaexplorer.It is presumed that the required data fields have been discretized.In this example it is age attribute.
2. Clicking on the associate tab will bring up the interface for association rule algorithm.
3. We will use apriori algorithm. This is the default algorithm.
4. In order to change the parameters for the run we click on the text box immediately to the right of the chosen button.

Result:



Experiment 4

4)Aim:Demonstration of Association rule process on dataset test.arff using apriori algorithm.
Data set
@relation attribute
@attribute bread{y,n}
@attribute jelly{y,n}
@attribute butter{y,n}
@attribute milk{y,n}
@attribute sugar{y,n}
@data
yyy n n
y n y n n
y n y y n
y n n y y
yy n y n

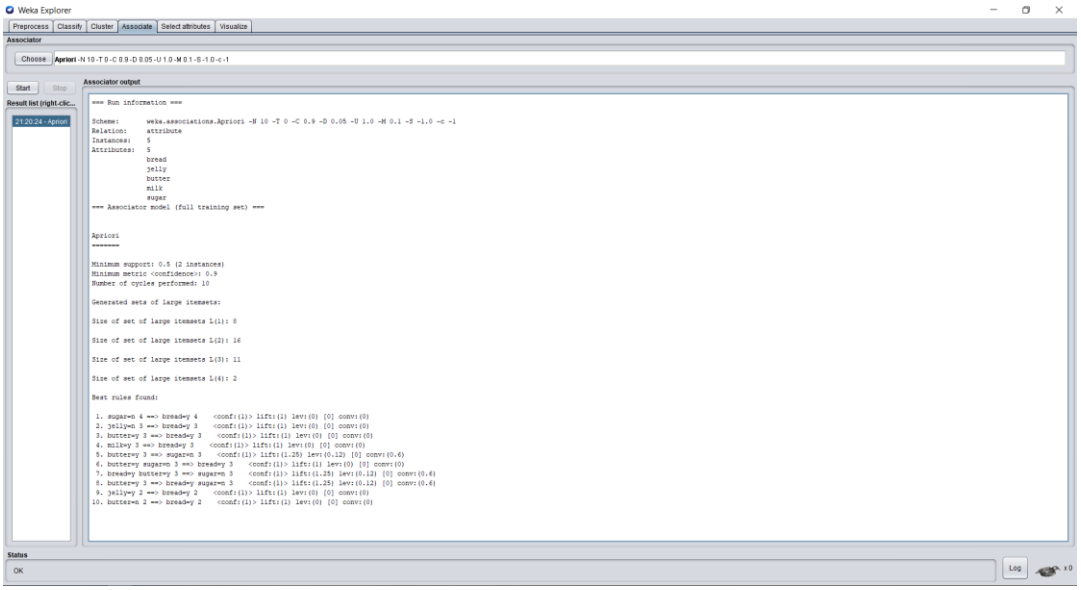
NAME
weka.associations.Apriori

SYNOPSIS
Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence.
The algorithm has an option to mine class association rules. It is adapted as explained in the second reference.

OPTIONS
minMetric -- Minimum metric score. Consider only rules with scores higher than this value.
verbose -- If enabled the algorithm will be run in verbose mode.
numRules -- Number of rules to find.
lowerBoundMinSupport -- Lower bound for minimum support.
classIndex -- Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
outputItemSets -- If enabled the itemsets are output as well.
car -- If enabled class association rules are mined instead of (general) association rules.
doNotCheckCapabilities -- If set, associator capabilities are not checked before associator is built (Use with caution to reduce runtime).
removeAllMissingCols -- Remove columns with all missing values.
significanceLevel -- Significance level. Significance test (confidence metric only).
treatZeroAsMissing -- If enabled, zero (that is, the first value of a nominal) is treated in the same way as a missing value.

Steps:

1. Open the datafile in wekaexplorer.It is presumed that the required data fields have been discretized.In this example it is age attribute.
2. Clicking on the associate tab will bring up the interface for association rule algorithm.
3. We will use apriori algorithm. This is the default algorithm.
4. In order to change the parameters for the run we click on the text box immediately to the right of the chosen button.



Experiment 5

5)Demonstration of classification rule process on dataset student.arff using j48 algorithm.
Data set

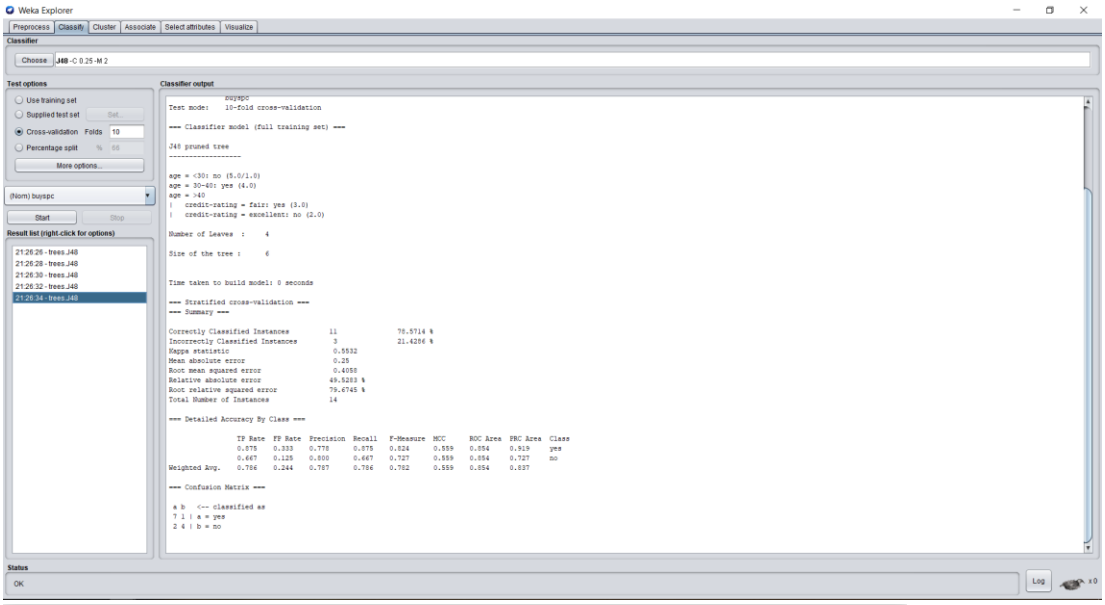
```
@relation student
@attribute age {<30,30-40,>40}
@attribute income {low,medium,high}
@attribute student {yes,no}
@attribute credit-rating {fair,excellent}
@attribute buyspc {yes,no}
@data
%
<30, high, no, fair, no
<30, high, no, excellent, no
30-40, high, no, fair, yes
>40, medium, no, fair, yes
>40, low, yes, fair, yes
>40, low, yes, excellent, no
30-40, low, yes, excellent, yes
<30, medium, no, fair, no
<30, low, yes, fair, no
>40, medium, yes, fair, yes
<30, medium, yes, excellent, yes
30-40, medium, no, excellent, yes
30-40, high, yes, fair, yes
>40, medium, no, excellent, no
%
```

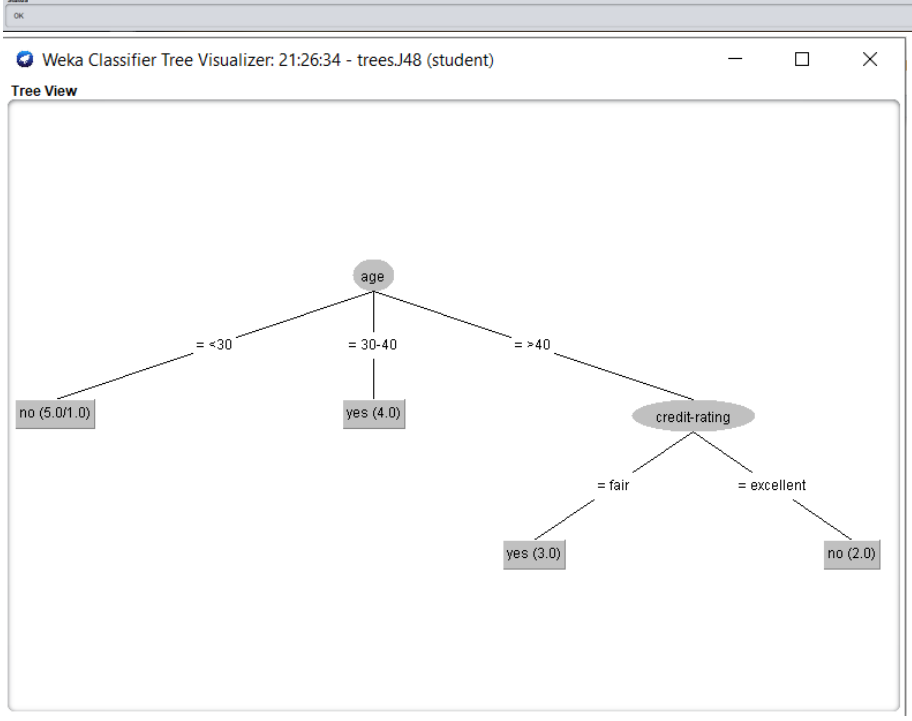
NAME
weka.classifiers.trees.J48

SYNOPSIS
Class for generating a pruned or unpruned C4.5 decision tree.
OPTIONS
seed -- The seed used for randomizing the data when reduced-error pruning is used.
unpruned -- Whether pruning is performed.
confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).
numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.
reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.
useLaplace -- Whether counts at leaves are smoothed based on Laplace.
doNotMakeSplitPointActualValue -- If true, the split point is not relocated to an actual data value. This can yield substantial speed-ups for large datasets with numeric attributes.

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose the J48 algorithm in classify.
3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.
5. Generate a tree by clicking visualize tree on that particular attribute.





Experiment 6

6)Demonstration of classification rule process on dataset employee.arff using j48 algorithm.
Data set

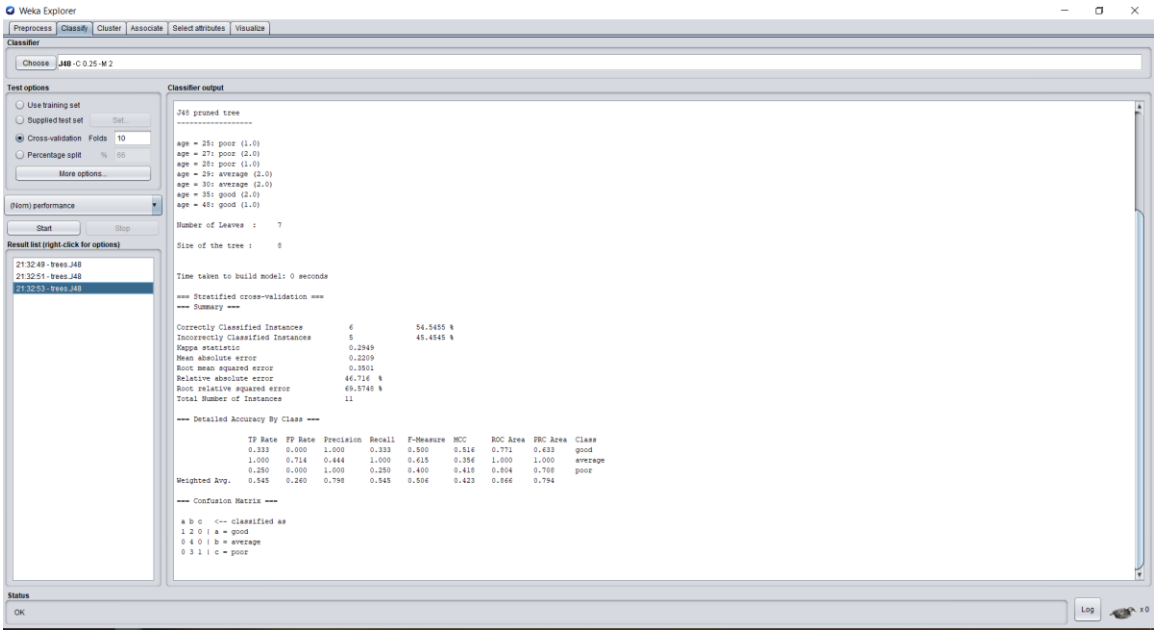
```
@relation employee
@attribute age{25,27,28,29,30,35,48}
@attribute salary{10k,15k,17k,20k,25k,30k,35k,32k}
@attribute performance{good,average,poor}
@data
25 10k poor
27 15k poor
27 17k poor
28 17k poor
29 20k average
30 25k average
29 25k average
30 20k average
35 32k good
35 35k good
48 32k good
```

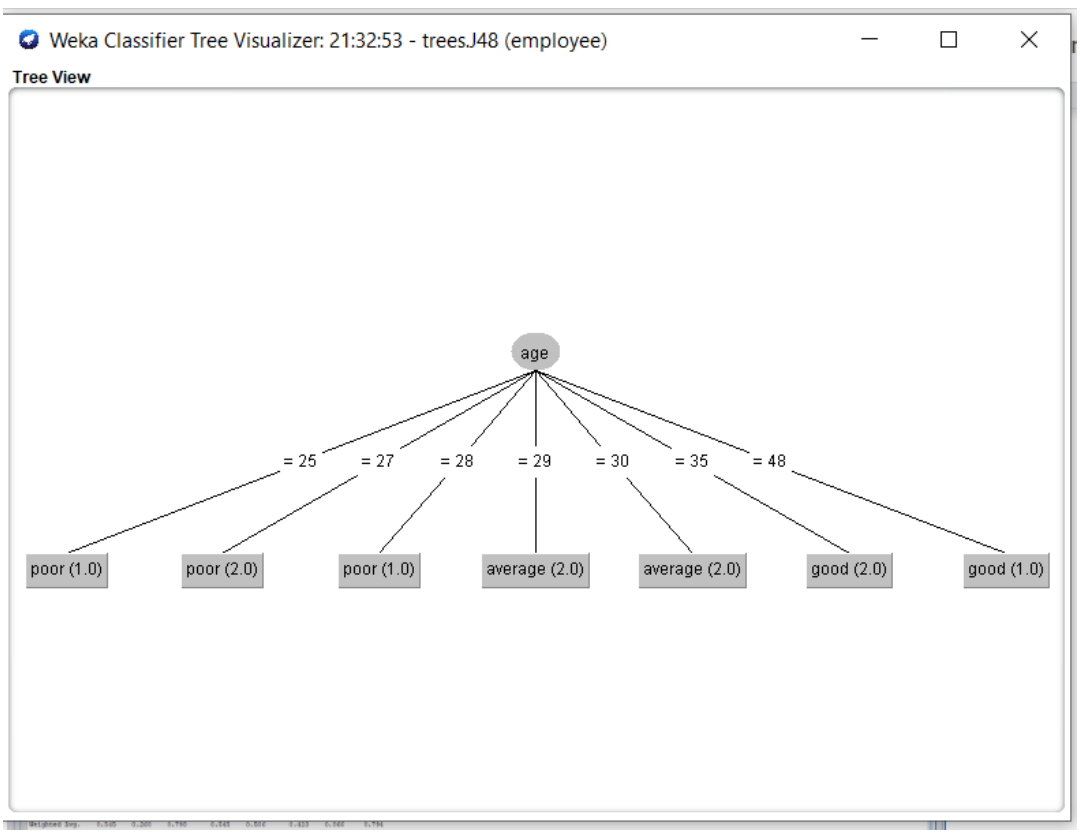
NAME
weka.classifiers.trees.J48

SYNOPSIS
Class for generating a pruned or unpruned C4.5 decision tree.
OPTIONS
seed -- The seed used for randomizing the data when reduced-error pruning is used.
unpruned -- Whether pruning is performed.
confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).
numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.
reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.
useLaplace -- Whether counts at leaves are smoothed based on Laplace.
doNotMakeSplitPointActualValue -- If true, the split point is not relocated to an actual data value. This can yield substantial speed-ups for large datasets with numeric attributes.

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose the J48 algorithm in classify.
3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.
5. Generate a tree by clicking visualize tree on that particular attribute.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) performance
Start
Stop
Result list (right-click for options)
21:32:49 - trees.J48
21:32:51 - trees.J48
21:32:53 - trees.J48
Classifier output
J48 pruned tree
---
age = 25: poor (1.0)
age = 27: poor (2.0)
age = 28: poor (1.0)
age = 29: average (2.0)
age = 30: average (2.0)
age = 35: good (2.0)
age = 48: good (1.0)
Number of Leaves : 7
Size of the tree : 8
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 6 54.5455 %
Incorrectly Classified Instances 5 45.4545 %
Kappa statistic 0.2949
Mean absolute error 0.2209
Root mean squared error 0.3501
Relative absolute error 46.716 %
Root relative squared error 69.5748 %
Total Number of Instances 11
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.333 0.000 1.000 0.333 0.500 0.516 0.771 0.633 good
1.000 0.714 0.444 1.000 0.615 0.356 1.000 1.000 average
0.250 0.000 1.000 0.250 0.400 0.418 0.804 0.708 poor
Weighted Avg. 0.545 0.260 0.798 0.545 0.506 0.423 0.866 0.794
=== Confusion Matrix ===
a b c <-- classified as
1 2 0 | a = good
0 4 0 | b = average
0 3 1 | c = poor
Status
OK
Log


Weka Classifier Tree Visualizer: 21:32:53 - trees.J48 (employee)
Tree View
age
= 25
= 27
= 28
= 29
= 30
= 35
= 48
poor (1.0)
poor (2.0)
poor (1.0)
average (2.0)
average (2.0)
good (2.0)
good (1.0)

Experiment 7

7)Aim:Demonstration of classification rule process on dataset employee.arff using Id3 algorithm.
Data set

```
@relation employee
@attribute age{25,27,28,29,30,35,48}
@attribute salary{10k,15k,17k,20k,25k,30k,35k,32k}
@attribute performance{good,average,poor}
@data
25 10k poor
27 15k poor
27 17k poor
28 17k poor
29 20k average
30 25k average
29 25k average
30 20k average
35 32k good
35 35k good
48 32k good
```

NAME
weka.classifiers.trees.Id3

SYNOPSIS
Class for constructing an unpruned decision tree based on the ID3 algorithm. Can only deal with nominal attributes. No missing values allowed. Empty leaves may result in unclassified instances.
OPTIONS
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.

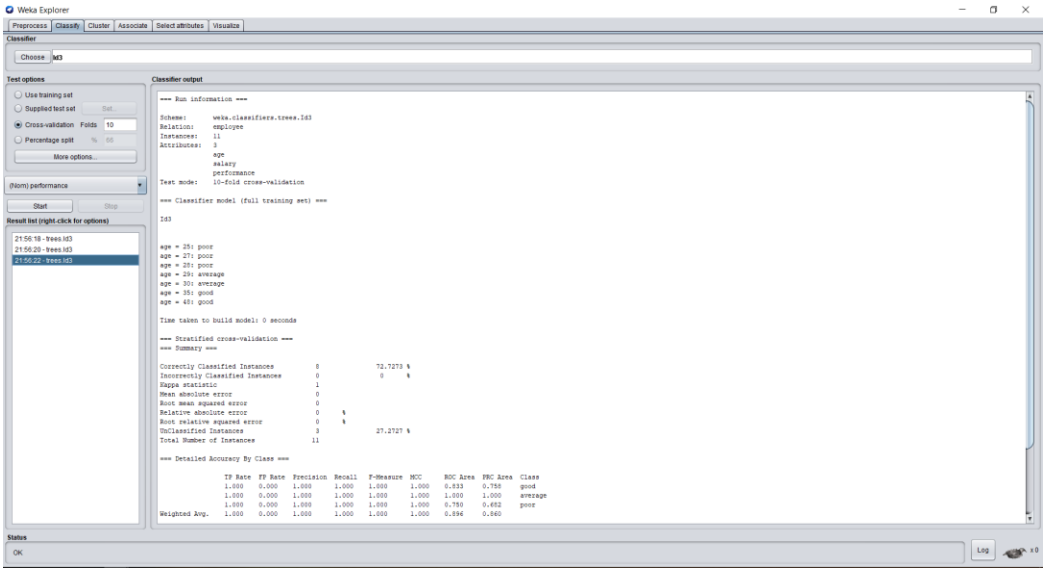
batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.

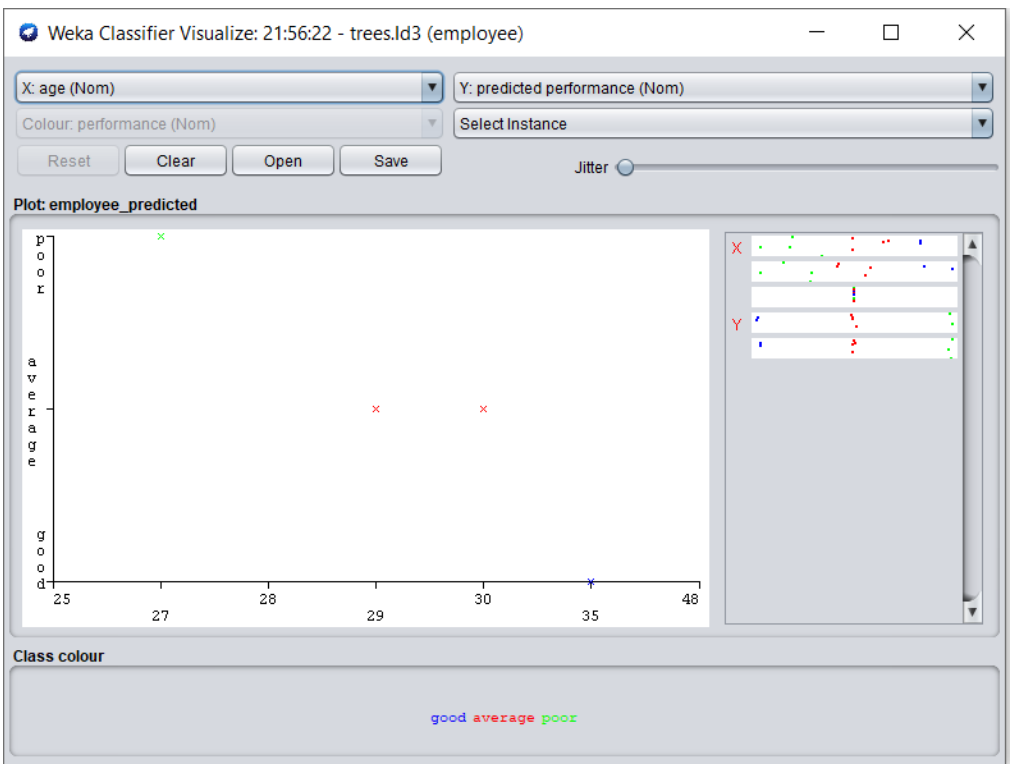
debug -- If set to true, classifier may output additional info to the console.

doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose Id3.
3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Id3
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
More options...
(Nom) performance
Start
Stop
Result list (right-click for options)
21:56:18 - trees.Id3
21:56:20 - trees.Id3
21:56:22 - trees.Id3
Classifier output
=== Run information ===
Scheme: weka.classifiers.trees.Id3
Relation: employee
Instances: 11
Attributes: 3
age
salary
performance
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Id3
age = 25: poor
age = 27: poor
age = 28: poor
age = 29: average
age = 30: average
age = 35: good
age = 48: good
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 8 72.7273 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 %
Root relative squared error 0 %
UnClassified Instances 3 27.2727 %
Total Number of Instances 11
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 0.000 1.000 1.000 1.000 1.000 0.833 0.758 good
1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 average
1.000 0.000 1.000 1.000 1.000 1.000 0.750 0.652 poor
Weighted Avg. 1.000 0.000 1.000 1.000 1.000 1.000 0.894 0.860
Status
OK
Log


Weka Classifier Visualize: 21:56:22 - trees.Id3 (employee)
X: age (Nom)
Y: predicted performance (Nom)
Colour: performance (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: employee_predicted
poor
average
good
25
27
28
29
30
35
48
X
Y
Class colour
good average poor

Experiment 8

8)Aim:Demonstration of classification rule process on dataset employee.arff using naive bayes algorithm.

Data set
@relation employee

@attribute age{25,27,28,29,30,35,48}

@attribute salary{10k,15k,17k,20k,25k,30k,35k,32k}

@attribute performance{good,average,poor}

@data

25 10k poor

27 15k poor

27 17k poor

28 17k poor

29 20k average

30 25k average

29 25k average

30 20k average

35 32k good

35 35k good

48 32k good

NAME

weka.classifiers.bayes.NaiveBayes

SYNOPSIS

Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances) -- if you need the UpdateableClassifier functionality, use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

OPTIONS

useKernelEstimator -- Use a kernel estimator for numeric attributes rather than a normal distribution.

numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.

batchSize -- The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.

debug -- If set to true, classifier may output additional info to the console.

displayModelInOldFormat -- Use old format for model output. The old format is better when there are many class values. The new format is better when there are fewer classes and many attributes.

doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime).

useSupervisedDiscretization -- Use supervised discretization to convert numeric attributes to nominal ones.

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose NaiveBayes from Bayes.
3. Click on start for each attribute to apply algorithm on data.
4. Discover the highest percentage of correctly classified instances.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose **NaiveBayes**

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation Folds 10
- Percentage split % 66

More options...

(Nom) performance

Start | Stop

Result list (right-click for options)

17:19:35 - bayes.NaiveBayes
17:19:40 - bayes.NaiveBayes
17:19:42 - bayes.NaiveBayes

Classifier output

```
  [total]          11.0    12.0    12.0



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          10               90.9091 %
Incorrectly Classified Instances         1                9.0909 %
Kappa statistic                          0.8625
Mean absolute error                      0.2899
Root mean squared error                  0.3171
Relative absolute error                 61.3111 %
Root relative squared error             63.0158 %
Total Number of Instances               11

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     good
                 1.000    0.143    0.800      1.000    0.889      0.828    1.000     1.000     average
                 0.750    0.000    1.000      0.750    0.857      0.810    1.000     1.000     poor
Weighted Avg.    0.909    0.052    0.927      0.909    0.908      0.868    1.000     1.000

=== Confusion Matrix ===
```

Status

OK

Log x 0

Start a web search ENG 17:20 06-02-20

Experiment 9

9)Demonstration of clustering rule process on dataset iris.arff using simple k-means

Data set

```
@RELATION iris

@ATTRIBUTE sepallength	REAL

@ATTRIBUTE sepalwidth	REAL
@ATTRIBUTE petallength	REAL
@ATTRIBUTE petalwidth	REAL
@ATTRIBUTE class	{Iris-setosa,Iris-versicolor,Iris-virginica}
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
```

NAME

weka.clusterers.SimpleKMeans

SYNOPSIS

Cluster data using the k means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean.

OPTIONS

seed -- The random number seed to be used.

displayStdDevs -- Display std deviations of numeric attributes and counts of nominal attributes.

numExecutionSlots -- The number of execution slots (threads) to use. Set equal to the number of available cpu/cores

canopyMinimumCanopyDensity -- If using canopy clustering for initialization and/or speedup this is the minimum T2-based density below which a canopy will be pruned during periodic pruning

dontReplaceMissingValues -- Replace missing values globally with mean/mode.

debug -- If set to true, clusterer may output additional info to the console.

numClusters -- set number of clusters

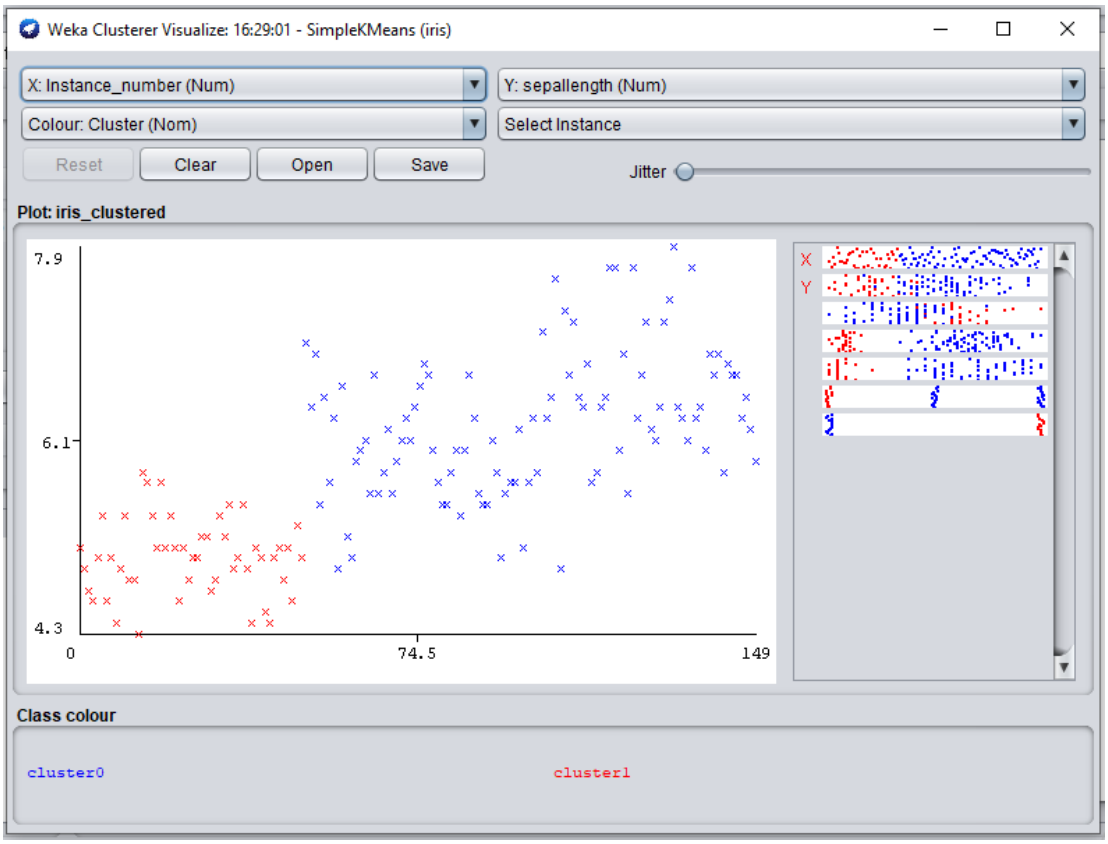
doNotCheckCapabilities -- If set, clusterer capabilities are not checked before the clusterer is built (Use with caution to reduce runtime).

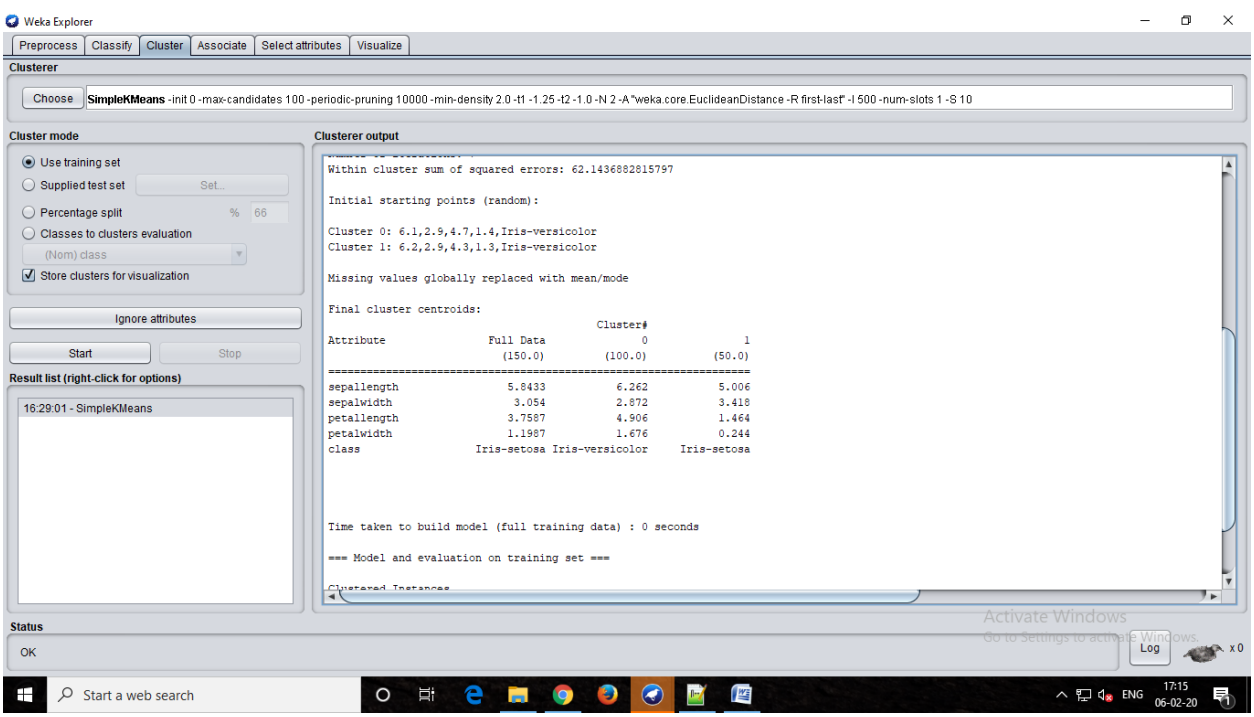
maxIterations -- set maximum number of iterations

Steps:

1. Open the datafile in weka explorer and click on cluster.
2. Select the simple k-means algorithm by clicking the choose.
3. Start the algorithm to generate cluster output.

4. Right click on simple k-means and select visualize cluster assignments.





Experiment 10

10)Demonstration of clustering rule process on dataset student.arff using simple k-means.

Data set

```
@relation student
@attribute sid numeric
@attribute name {usha,hari,rajesh,kiran,giri,manash}
@attribute DM numeric
@attribute WT numeric
@attribute CN numeric
@attribute AI numeric
@attribute STM numeric
@attribute total numeric
@attribute result {pass,fail}

@data
1,usha,60,55,45,50,40,250,pass
2,hari,60,55,45,40,40,240,pass
3,rajesh,60,55,40,50,40,240,pass
4,kiran,60,55,30,50,40,235,fail
2,giri,60,55,45,60,40,260,pass
3,manash,60,55,65,50,40,270,pass
```

NAME

weka.clusterers.SimpleKMeans

SYNOPSIS

Cluster data using the k means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean.

OPTIONS

seed -- The random number seed to be used.

displayStdDevs -- Display std deviations of numeric attributes and counts of nominal attributes.

numExecutionSlots -- The number of execution slots (threads) to use. Set equal to the number of available cpu/cores

canopyMinimumCanopyDensity -- If using canopy clustering for initialization and/or speedup this is the minimum T2-based density below which a canopy will be pruned during periodic pruning

dontReplaceMissingValues -- Replace missing values globally with mean/mode.

debug -- If set to true, clusterer may output additional info to the console.

numClusters -- set number of clusters

doNotCheckCapabilities -- If set, clusterer capabilities are not checked before the clusterer is built (Use with caution to reduce runtime).

maxIterations -- set maximum number of iterations

Steps:

1. Open the datafile in weka explorer and click on cluster.
2. Select the simple k-means algorithm by clicking the choose.
3. Start the algorithm to generate cluster output.
4. Right click on simple k-means and select visualize cluster assignments.