Comparison of Decision Tree Classifier and Bayes Classifier using WEKA

Vangala Bhavana
B.E 4/4, CSE Dept.
Vasavi College of Engineering
Telangana, India

T. Adilakshmi, PhD
Professor & Head of Department, CSE Dept.
Vasavi College of Engineering
Telangana, India

ABSTRACT

Data Mining is the process of locating potentially practical, interesting and previously unknown patterns from a big volume of data. It plays an important role in result orientation. Data mining can be used in each and every aspect of life. The same is similarly significant in other areas including sales/ marketing, revenue services, sports, health care and insurance etc. Classification is used to builds models from data with predefined classes as the model is used to classify new instance whose classification is not known. This paper compares the two famous algorithms called Bayesian and Decision tree algorithm and how it works on nominal and numerical data sets and demonstrates its results. The accuracy, precision, and classification errors are also measured to compare algorithm. WEKA tool has been used to perform the experiment.

General Terms

Algorithms, Evaluation, Results, Comparison, Data Set

Keywords

Data Mining, Classification, WEKA, Bayesian, Decision Tree.

1. INTRODUCTION

Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and it can be used for classifying newly available data. The term could cover any context in which some decision or forecast is made on the basis of presently available information. Classification procedures recognized method for repeatedly making such decisions in new situations. Here if we assume that problem is a concern with the construction of a procedure that will be applied to a continuing sequence of cases in which each new case must be assigned to one of a set of pre-defined classes on the basis of observed features of data. Creation of a classification procedure from a set of data for which the exact classes are known in advance is termed as pattern recognition or supervised learning. Contexts in which a classification task is fundamental include, for example, assigning individuals to credit status on the basis of financial and other personal information, and the initial diagnosis of a patient's disease in order to select immediate treatment while awaiting perfect test results. Some of the most critical problems arising in science, industry and commerce can be called as classification or decision problems. Three main historical strands of research can be identified: statistical, machine learning and neural network. All groups have some objectives in common. They have all attempted to develop procedures that would be able to handle a wide variety of problems and to be extremely general used in practical settings with proven success. There are two types of learning. They are supervised and unsupervised learning. The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations. The class labels of training data are unknown in unsupervised learning. Classification is said to have two steps:-

i. Model construction: describing a set of predetermined classes
ii. Model usage: for classifying future or unknown objects

We also have Lazy learners and Eager Learners in Classification. Lazy learners simply stores the training data without doing any further munging on it, till it gets the next test set. Eager learner's munges the training data as soon as it receives it.

Section two deals with the Classification algorithms and Section three deals with the Experimental Analysis which contains Algorithms explanation, Weka tool, Evaluation Measures, Results and comparison of Results. Section four deals with the conclusion and Future scope and Section 5 contains References.

2. CLASSFICATION ALGORITHMS

2.1 Decision Tree

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. The following decision tree is for the concept buy computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents test on an attribute. Each leaf node represents a class. The benefits of having a decision tree are as follows −It does not require any domain knowledge. It is easy to comprehend. The learning and classification steps of a decision tree are simple and fast.

Algorithm for Decision Tree Induction:

```
create a node N;
if tuples in D are all of the same class, C then
  return N as leaf node labeled with class C;
if attribute_list is empty then
  return N as leaf node with labeled
  with majority class in D;|| majority voting
apply attribute_selection_method(D, attribute_list)
to find the best splitting_criterion;
label node N with splitting_criterion;
if splitting_attribute is discrete-valued and
```

```
multiway splits allowed then  // no restricted to binary trees
attribute_list = splitting attribute; // remove splitting attribute
for each outcome j of splitting criterion
  // partition the tuples and grow subtrees for each partition
  let Dj be the set of data tuples in D satisfying outcome j; // a partition
  if Dj is empty then
    attach a leaf labeled with the majority
    class in D to node N;
  else
    attach the node returned by Generate
    decision tree(Dj, attribute list) to node N;
  end for
return N;
```

2.2 Bayesian Classification

2.2.1 Baye's Theorem

Bayes' Theorem is named after Thomas Bayes. There are two types of probabilities −

a) Posterior Probability [P (H/X)]
b) Prior Probability [P (H)] where X is data tuple and H is some hypothesis.
c) According to Bayes' Theorem, P (H/X) = P(X/H) P (H) / P(X)

2.2.2 Naïve Bayes

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Algorithm

Bayes theorem provides a way of calculating the posterior probability, *P* (*c*|*x*), from *P*(*c*), *P*(*x*), and *P* (*x*|*c*). Naive Bayes classifier assume that the effect of the value of a predictor (*x*) on a given class (*c*) is Independent of the values of other predictors. This assumption is called Class conditional independence.

Likelihood   Class Prior Probability

$$P(c\mid x)=\frac{P(x\mid c)P(c)}{P(x)}$$

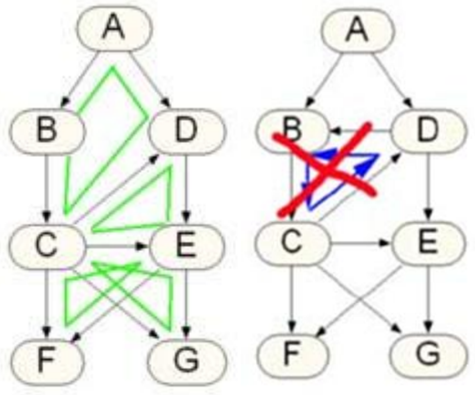
Posterior Probability   Predictor Prior Probability

$$P(c\mid X) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

2.2.3 Bayesian Belief Network

Bayesian Belief Networks specify joint conditional probability distributions. They are also known as Belief Networks, Bayesian Networks, or Probabilistic Networks. A Belief Network allows class conditional independencies to be defined between subsets of variables. It provides a graphical model of causal relationship on which learning can be performed. We can use a trained Bayesian Network for classification. There are two components that define a Bayesian Belief Network −

i. Directed acyclic graph
ii. A set of conditional probability tables



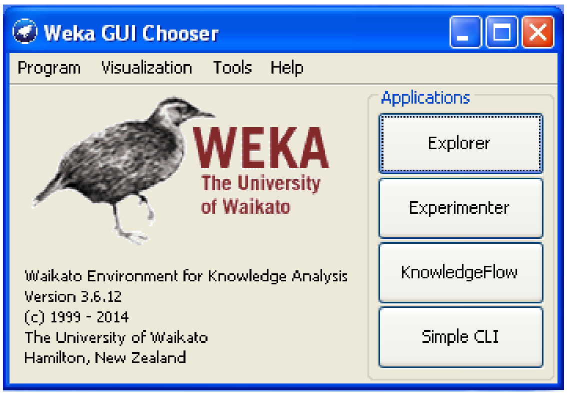
A Valid BaseNet   Not a BaseNet

3. EXPERIMENTAL ANALYSIS

3.1 Weka

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like this, and the bird sounds like this. Weka is open source software issued under the GNU General Public License.

Weka GUI Chooser



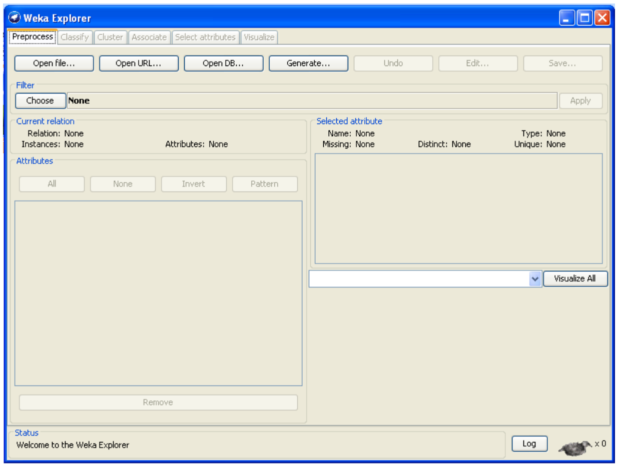
Waikato Environment for Knowledge Analysis
Version 3.6.12
(c) 1999 - 2014
The University of Waikato
Hamilton, New Zealand

There are four buttons on Weka GUI Chooser

1. Explorer
2. Experimenter
3. Knowledge Flow
4. Simple CLI

Working with Explorer

Click on Explorer button on Weka GUI Chooser. Explorer window opens



Filter: None

Relation: None
Instances: None
Attributes: None

Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

WEKA Explorer option has Six tabs: 1. Preprocess- used to choose the data file to be used by the application 2. Classify- used to test and train different learning schemes on the preprocessed data file under experimentation 3. Cluster- used to apply different tools that identify clusters within the data file 4. Association- used to apply different rules to the data file that identify association within the data 5. Select attributes-used to apply different rules to reveal changes based on selected attributes inclusion or exclusion from the experiment 6. Visualize- used to see what the various manipulation produced on the data set in a 2D format, in scatter plot and bar graph output.

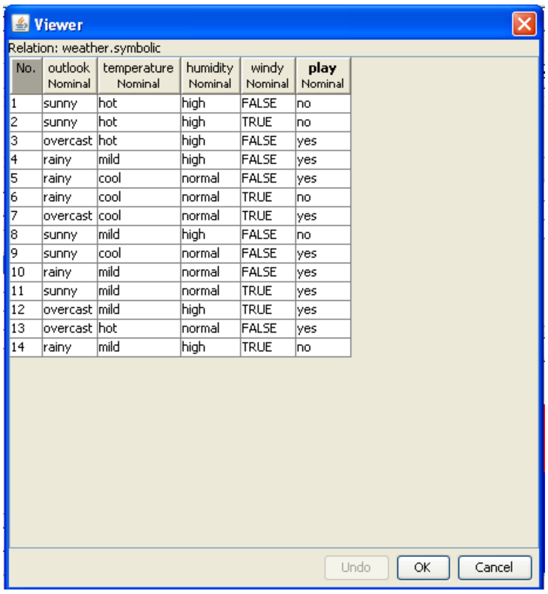
3.2 Data Set

These datasets are taken as sample data. This sample data can be numeric data or nominal data. There are process that convert numeric data into nominal data or categorical data.

a) Import a Data Set into the Explorer

b) Click on open file -> browse for the Dataset file (.arff file) -> select and click Open.

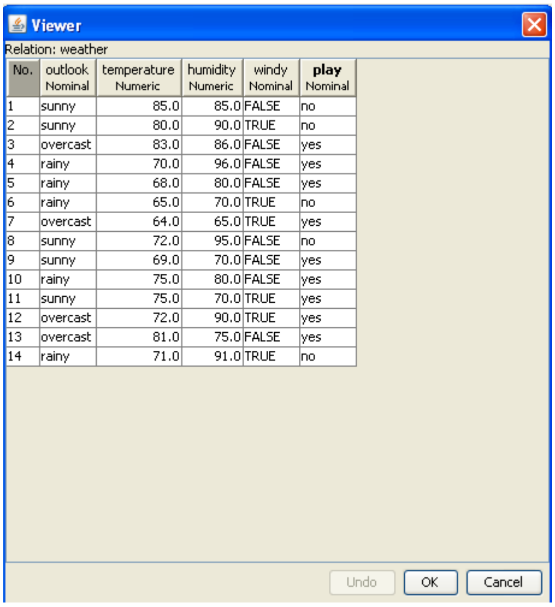
Nominal Data Set



Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Numeric Data set



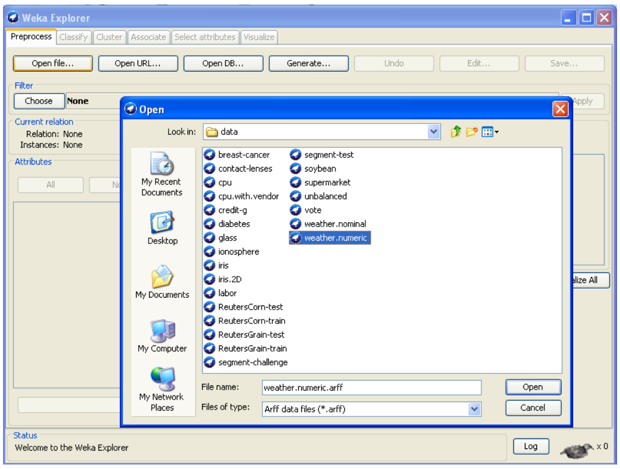
Relation: weather

No.	outlook Nominal	temperature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

Applying Discretize to convert Numeric Dataset to Categorical Dataset

1. Import a Dataset containing numeric Data items into the explorer.

   Example: Weather-Numeric.arff



Filter: None

Relation: None
Instances: None

Look in: data

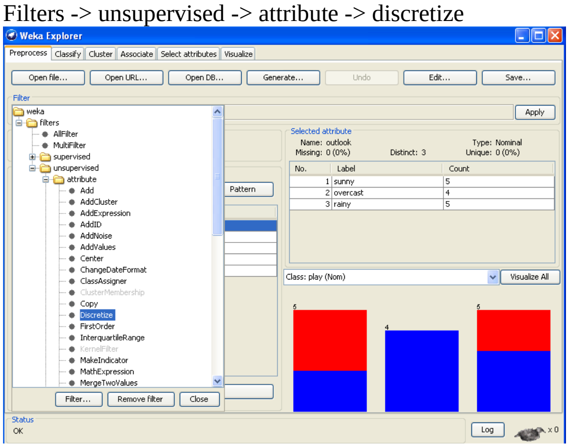
breast-cancer
contact-lenses
cpu
cpu.with.vendor
credit-g
diabetes
glass
ionosphere
iris
iris.2D
labor
ReutersCorn-test
ReutersCorn-train
ReutersGrain-test
ReutersGrain-train
segment-challenge
segment-test
soybean
supermarket
unbalanced
vote
weather.nominal
weather.numeric

File name: weather.numeric.arff
Files of type: Arff data files (*.arff)

Welcome to the Weka Explorer

2. From Filters, Choose filter Discretize.

   Filters -> unsupervised -> attribute -> discretize



weka
filters
AllFilter
MultiFilter
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddValues
Center
ChangeDateFormat
ClassAssigner
ClusterMembership
Copy
Discretize
FirstOrder
InterquartileRange
KernelFilter
MakeIndicator
MathExpression
MergeTwoValues

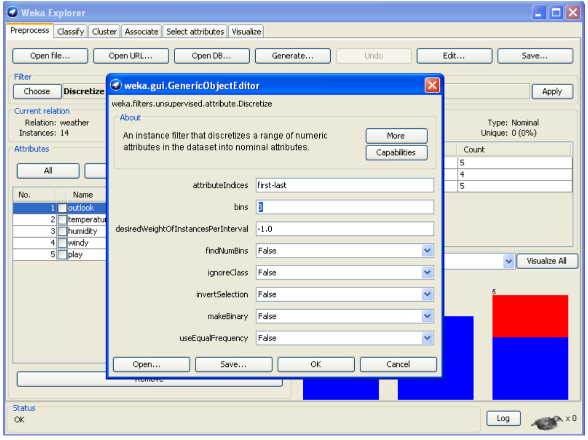
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom)

OK

3. Click on the filter name in the explorer to open Generic Object Browser to Edit Parameter values



weka.gui.GenericObjectEditor

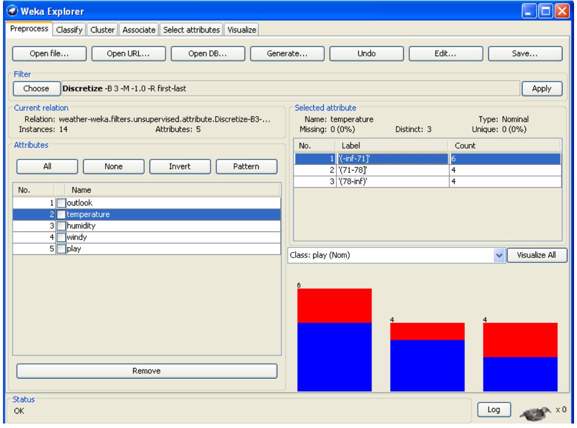
weka.filters.unsupervised.attribute.Discretize

About: An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices	first-last
bins	3
desiredWeightOfInstancesPerInterval	-1.0
findNumBins	False
ignoreClass	False
invertSelection	False
makeBinary	False
useEqualFrequency	False

Relation: weather
Instances: 14

4. Click Apply button to apply filter on the Dataset to convert it to categorical Dataset and find the data changed to categorical data items.



Filter: **Discretize** -B 3 -M -1.0 -R first-last

Current relation
Relation: weather-weka.filters.unsupervised.attribute.Discretize-B3-...
Instances: 14 Attributes: 5

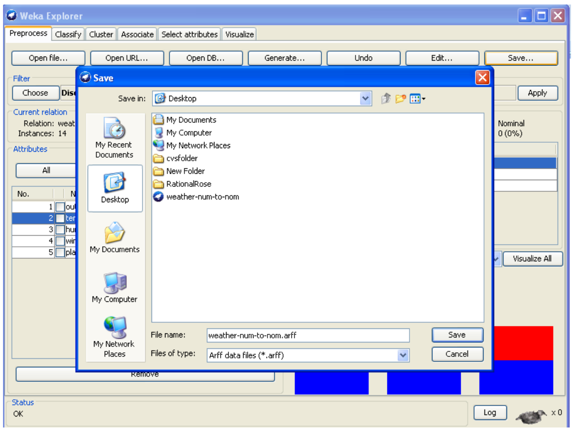
No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Selected attribute
Name: temperature Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count
1	'(-inf-71]'	6
2	'(71-78]'	4
3	'(78-inf)'	4

Class: play (Nom)

5. To give the meaningful names to the categories (labels) generated in the step 4, save the dataset into a text file, and change the names.
   a) Click on save button in the explorer
   b) Select a location, give a filename and click to save.
   c) Go to the file and it in any text editor to change the Label names.
6. Save the Dataset

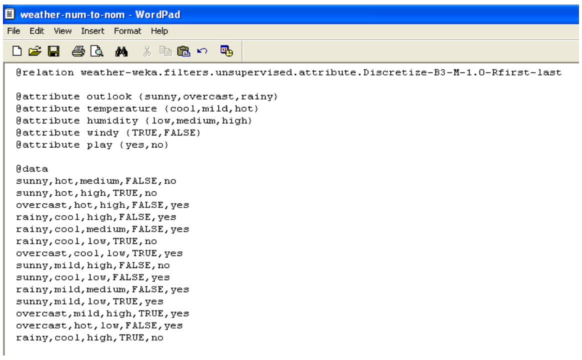


Save in: Desktop

My Documents
My Computer
My Network Places
cvsfolder
New Folder
RationalRose
weather-num-to-nom

File name: weather-num-to-nom.arff
Files of type: Arff data files (*.arff)

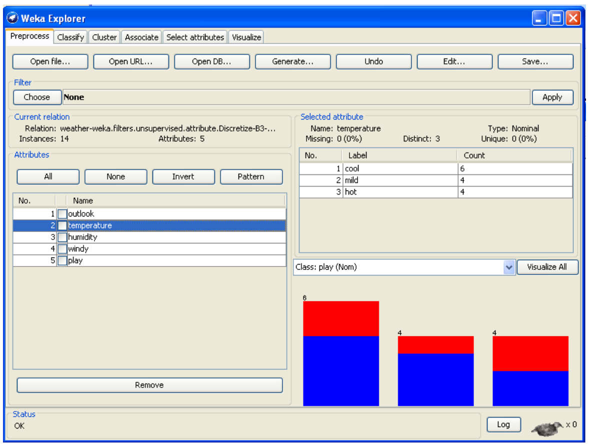
7. Open in an Editor (WordPad)
8. Edit Data and Save



weather-num-to-nom - WordPad

```
@relation weather-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-Rfirst-last

@attribute outlook {sunny,overcast,rainy}
@attribute temperature {cool,mild,hot}
@attribute humidity {low,medium,high}
@attribute windy {TRUE,FALSE}
@attribute play {yes,no}

@data
sunny,hot,medium,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,cool,high,FALSE,yes
rainy,cool,medium,FALSE,yes
rainy,cool,low,TRUE,no
overcast,cool,low,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,low,FALSE,yes
rainy,mild,medium,FALSE,yes
sunny,mild,low,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,low,FALSE,yes
rainy,cool,high,TRUE,no
```

9. Load the Edited file again into the Explorer .



Filter: **None**

Current relation
Relation: weather-weka.filters.unsupervised.attribute.Discretize-B3-...
Instances: 14 Attributes: 5

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Selected attribute
Name: temperature Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count
1	cool	6
2	mild	4
3	hot	4

Class: play (Nom)

3.3 Evaluation Measures

3.3.1 Model Evaluation and Selection

Evaluation metrics: How can we measure accuracy? Other metrics to consider? Use validation test set of class-labeled tuples instead of training set when assessing accuracy Methods for estimating a classifier's accuracy: Holdout method, random subsampling, Cross-validation, Bootstrap.

3.3.2 Classifier Evaluation Metrics Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

Classifier Accuracy or recognition rate: percentage of test set tuples that are correctly classified.

Accuracy = (TP + TN)/All

Error rate = (FP + FN)/All

Where TP is True Positives, FP is False Positives, FN is False Negatives, and TN is True Negatives.

Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

Recall: completeness – what % of positive tuples did the classifier label as positive? Perfect score is 1.0.n precision & recall

$$recall = \frac{TP}{TP + FN}$$

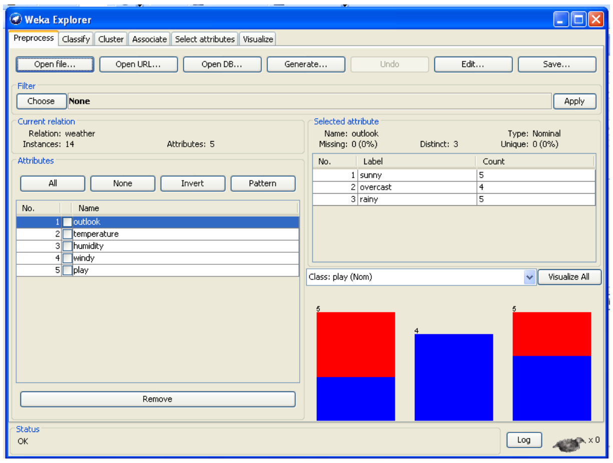
***F* measure (*F*$_1$or*F*-score)**: harmonic mean of precision and recall.

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

3.4 Results

3.4.1 Decision Tree Classifier:

1) Import Weather-numeric Dataset into the Explorer



Filter: None

Relation: weather
Instances: 14 Attributes: 5

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

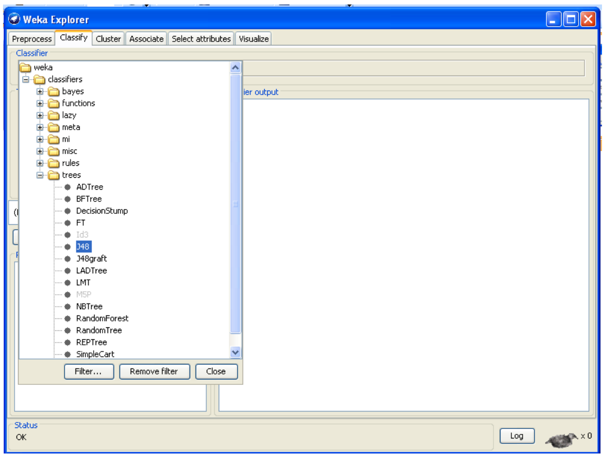
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

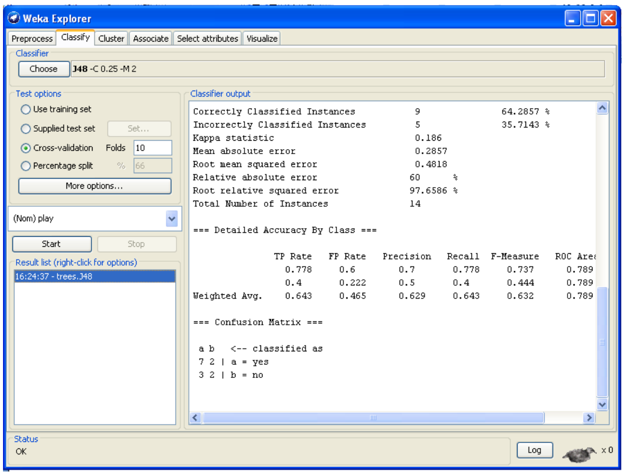
Class: play (Nom)

OK

1. Click on the Classify Tab in the Explorer.
2. Click on Choose button to choose a Classifier from the list of Classifiers available
   a) From Tree Classifier category, select ‘J48 ‘ classifier



b) Click on Apply button to Apply the J48 classifier on the Dataset.



J48 -C 0.25 -M 2

Folds 10

% 66

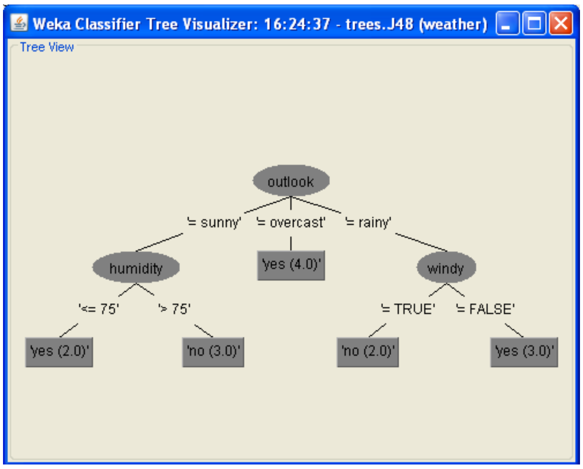
(Nom) play

16:24:37 - trees.J48

```
Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area
                 0.778     0.6       0.7         0.778     0.737      0.789
                 0.4       0.222     0.5         0.4       0.444      0.789
Weighted Avg.    0.643     0.465     0.629       0.643     0.632      0.789

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

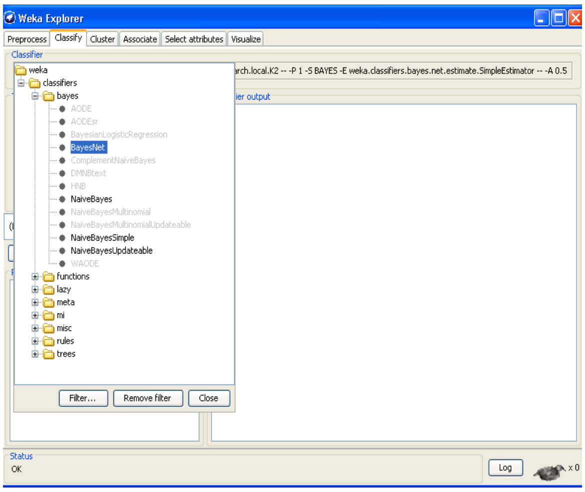
OK

c) Right click on the result list item to view the Tree Right-click -> visualize tree.



3.4.2 Bayes Classification Algorithm

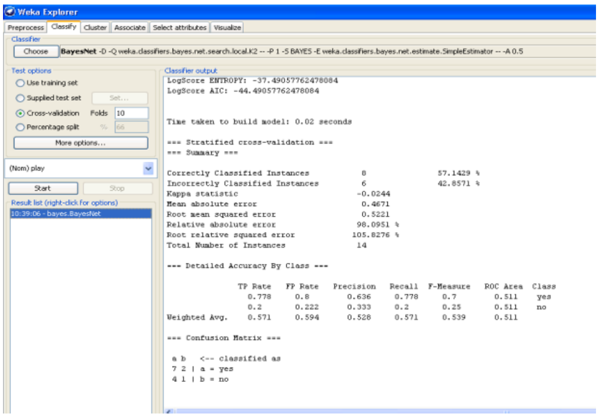
1. Choose the algorithm Bayes Net from Bayes Classifiers



arch.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

OK

2. Apply the algorithm and apply on the data set.



Classifier: BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) play

Result list (right-click for options)
10:39:06 - bayes.BayesNet

Classifier output

```
LogScore ENTROPY: -37.49057762478084
LogScore AIC: -44.49057762478084


Time taken to build model: 0.02 seconds

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

Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                         -0.0244
Mean absolute error                      0.4671
Root mean squared error                  0.5221
Relative absolute error                 98.0951 %
Root relative squared error            105.8276 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.778     0.8       0.636     0.778     0.7         0.511    yes
                 0.2       0.222     0.333     0.2       0.25        0.511    no
Weighted Avg.    0.571     0.594     0.528     0.571     0.539       0.511

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 4 1 | b = no
```

3.5 Comparison of Results

Decision Trees are very flexible, easy to understand, and easy to debug. They will work with classification problems and regression problems. So if you are trying to predict a categorical value like (red, green, up, down) or if you are trying to predict a continuous value like 2.9, 3.4 etc. Decision Trees will handle both problems. Probably one of the coolest things about Decision Trees is they only need a table of data and they will build a classifier directly from that data without needing any up front design work to take place. To some degree properties that don't matter won't be chosen as splits and will get eventually pruned so it's very tolerant of nonsense. To start its set it and forget it.

Bayes requires you build a classification by hand. There's no way to just toss a bunch of tabular data at it and have it pick the best features it will use to classify. Picking which features matter is up to you. Decisions trees will pick the best features for you from tabular data. If there were a way for Bayes to pick features you'd be getting close to using the same techniques that make decision trees work like that. Give this fact that means you may need to combine Bayes with other statistical techniques to help guide you towards what features best classify and that could be using decision trees. Bayes will answer as a continuous classifier. There are techniques to adapt it to categorical prediction however they will answer in terms of probabilities like (A 90%, B 5%, C 2.5% D 2.5%) Bayes can perform quite well, and it doesn't over fit nearly as much so there is no need to prune or process the network. That makes them simpler algorithms to implement. However, they are harder to debug and understand because it's all probabilities getting multiplied 1000's of times so you have to be careful to test its doing what you expect. Bayes does quite well when the training data doesn't contain all possibilities so it can be very good with low amounts of data. Decision trees work better with lots of data compared to Bayes.

4. CONCLUSION

Decision tree mainly deals with the Yes or No type of results and it limits the output to certain extent only. Bayesian model is generally useful if it helps us to greater understand the world we are modeling, and if it allows us to make useful predictions about how the world will behave. It is often easier to experiment with the model as compared to reality. As with all analytic methods, there are also limitations of the decision tree method that users must be aware of. The main disadvantage is that it can be subject to over fitting and under fitting, particularly when using a small data set. This problem can limit the generalizability and robustness of the resultant models. Another potential problem is that strong correlation between different potential input variables may result in the selection of variables that improve the model statistics but are not causally related to the outcome of interest. Bayesian nets are one such way. Because a Bayes net only relates nodes that are probabilistically related by some sort of causal dependency, an enormous saving of computation can result. There is no need to store all possible configurations of states, all possible worlds, if you will. All that is needed to store and work with is all possible combinations of states between sets of related parent and child nodes (families of nodes, if you will). This makes for a great saving of table space and computation. (Of course, some models are still too large for today's Bayes net algorithms. A second reason Bayesian nets are proving so useful is that they are so adaptable.

5. REFERENCES

[1] C. Apte and S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997

[2] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984

[4] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998

[5] P. K. Chan and S. J. Stolfo. Learning arbiter and combinertrees from partitioned data for scaling machine learning. KDD'95

[6] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 2000.

[7] J. Magidson. The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection. In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994.

[8] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ : A fast scalable classifier for data mining. EDBT'96.

[9] T. M. Mitchell. Machine Learning. McGraw Hill, 1997.

[10] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.