Performance analysis of Data Mining dataset in Weka Implement car dataset

Nagat Esiad Rahel
Faculty of Art and Science
University Of El- Zintan .. Bader-Libya
nagatrahil08@gmail.com

Abstract

The "Car Manufacturing" sector occupies a prime position in the development of automobile industry .in this paper, a proposed data mining application in car manufacturing domain is explained and experimented. The dataset are retrieved from UCI machine learning repository. The purpose of this paper is to establish a classifier that is much more reliable in classification for future objects.

Classification is one important techniques of data mining. Classification is a supervised learning problem of assigning an object to one of several pre-defined categories based upon the attributes of the object. In this paper we make use of a large database containing 7 attributes and 1728 instances We compared results of simple classification technique (using the J48 Decision Tree Induction Algorithm and MONK) with the results, based upon various parameters using WEKA (Waikato Environment for Knowledge Analysis), a Data Mining tool. The results of the experimenter show comparative between three algorithm which the best algorithm and least error.

The physical characteristics of a car viz . engine-location , number of doors ,stroke , city-mpg ,price ,etc., are considered to determine the performance of a car .Hence development of such a classifier , though a voluminous task , is immensely essential in car manufacturing realm . Machine learning techniques can help in the integration of computer - based systems in predicting the quality of car and to improve the efficiency of the system .The classification models were trained by using 214 datasets .The predicted values for the classifiers were evaluated using 10 fold cross validation and the results were compared .

keywords

Data mining, Machine Learning Techniques, J48, Decision trees, Car market, WEKA classification

INTRODUCTION

When we see in our world is full of data. After compilation and organization, data, if we are lucky, becomes information. In today's interconnected world, information exists in electronic form that can be stored and transmitted instantly. Challenge is to understand, integrate, and apply information to generate useful knowledge “actionable intelligence" . So we need to use technique to help us about this volume of data.

Data Mining as an analytic process designed to explore data (usually large amounts of - typically business or market related - data) in search for consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has most direct business applications.

Firstly in my paper I will define my datasets and define each algorithm I have used and explained how it work and how I have applied them. after that I will discuss the results and compare between all algorithms that I have used.

Data Mining with WEKA This report/tutorial uses a detailed example to illustrate some of the basic data preprocessing and mining operations that can be performed using WEKA. It is based on WEKA version 3.6. Some of the interface elements and modules may have changed in the most current version of WEKA. You can download the most current version of WEKA from the WEKA Web site. The current version includes a few additional features in the GUI and has a more organized packaging structure for the Java components. You should pay attention to these differences as you go through the tutorial. The differences in packaging structure are particularly important when you are running WEKA from the command line.

• **What is WEKA?**

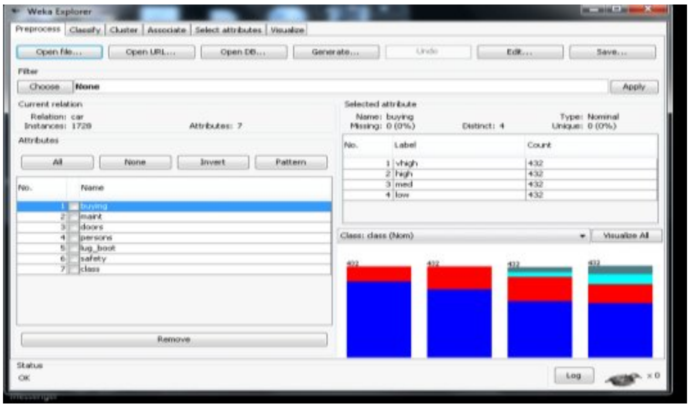
• Developed at UNIV of Waikato in New Zealand
• A collection of state-of-art machine earning algorithms and data pre-processing tools
• Provide implementation of
– Regression
– Classification
– Clustering
– Association rules
– Feature selection

Weka 3.6 : data mining software in java

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, Preprocessing ,classification , regression , clustering , association rules, and visualization.

Data Preprocessing in WEKA

The following guide is based WEKA version 3.6 Additional resources on WEKA, including sample data sets can be found from the official WEKA Web site.



This project illustrates some of the basic data preprocessing operations that can be performed using WEKA. The sample data set used for this project, unless otherwise indicated, is the "car data" available in arff format (car.arff).The data contains the following field.

Car Dataset

Data Set Characteristics:	Multivariate	**Number of Instances:**	**1728**	**Area:**	**N/A**
Attribute Characteristics:	Categorical	**Number of Attributes:**	**7 with clacc**	**Date Donated**	**1997-06-01**
Associated Tasks:	Classification	**Missing Values?**	**No**	**Number of Web Hits:**	**112752**

Data Set Information:

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR car acceptability

. PRICE overall price

. buying buying price

. maint price of the maintenance

. TECH technical characteristics

. COMFORT comfort

. doors number of doors

. persons capacity in terms of persons to carry

. lug_boot the size of luggage boot

. safety estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR) , the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

3-2 Attribute Information:

Class Values:

unacc, acc, good, vgood

Attributes:

buying: vhigh, high, med, low.

maint: vhigh, high, med, low.

doors: 2, 3, 4, 5more.

persons: 2, 4, more.

lug_boot: small, med, big.

safety: low, med, high.

some details about car dataset

1. Title: Car Evaluation Database

2. Sources:

(a) Creator: Marko Bohanec

(b) Donors: Marko Bohanec (marko.bohanec@ijs.si)

Blaz Zupan (blaz.zupan@ijs.si)

(c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is derived, was first presented in M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988. Within machine-learning, this dataset was used for the evaluation of HINT (Hierarchy INduction Tool), which was proved to be able to completely reconstruct the original hierarchical model. This, together with a comparison with C4.5, is presented in B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR	car acceptability
. PRICE	overall price
. buying	buying price
. maint	price of the maintenance
. TECH	technical characteristics
. COMFORT	comfort
. doors	number of doors
. persons	capacity in terms of persons to carry

. lug_boot the size of luggage boot

. safety estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 1728
(instances completely cover the attribute space)

6. Number of Attributes: 6

7. Attribute Values:

buying	v-high, high, med, low
maint	v-high, high, med, low
doors	2, 3, 4, 5-more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	(3.993 %)
v-good	65	(3.762 %)

Information about the dataset

CLASSTYPE: nominal

CLASSINDEX: last

This part of car dataset

```
@relation car
@attribute buying {vhigh,high,med,low}
@attribute maint {vhigh,high,med,low}
@attribute doors {2,3,4,5more}
@attribute persons {2,4,more}
@attribute lug_boot {small,med,big}
@attribute safety {low,med,high}
@attribute class {unacc,acc,good,vgood}
@data
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc
vhigh,vhigh,2,2,med,med,unacc
vhigh,vhigh,2,2,med,high,unacc
```

Decision tree algorithm

For discrete attributes, the algorithm makes predictions based on the relationships between input columns in a dataset. It uses the values, known as states, of those columns to predict the states of a column that you designate as predictable. Specifically, the algorithm identifies the input columns that are correlated with the predictable column.

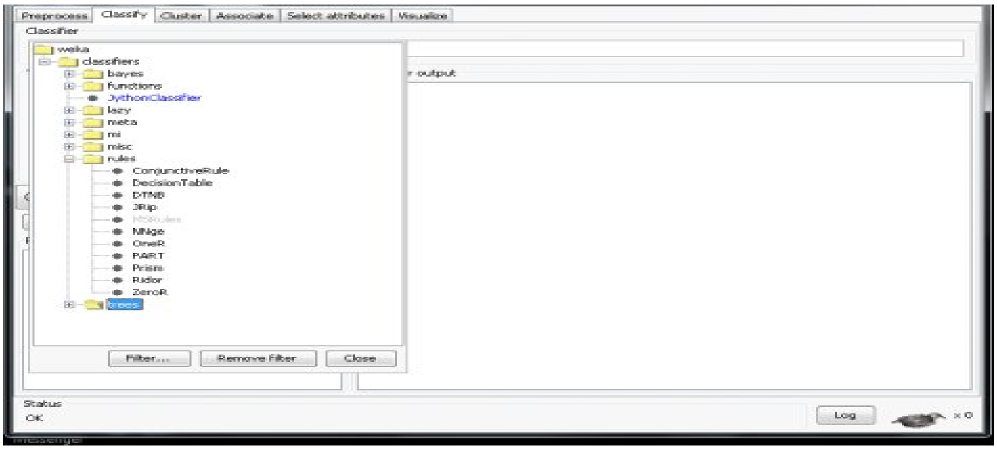
The J48 Decision Tree Induction Algorithm and MONK

The algorithm used by Weka and the MONK project is known as J48. J48 is a version of an earlier algorithm developed by J. Ross Quinlan, the very popular C4.5. Decision trees are a classic way to represent information from a machine learning algorithm, and offer a fast and powerful way to express structures in data.

Classification via Decision Trees in WEKA

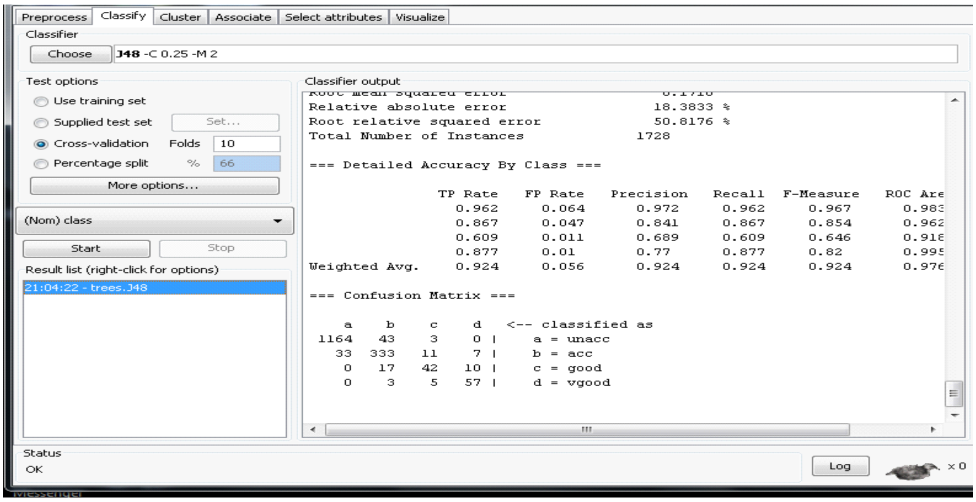
This project illustrates the use of C4.5 (J48) classifier in WEKA. The sample data set used for this project,in data base available in car.arff This document assumes that appropriate data preprocessing has been perfromed. In this case ID field has been removed. Since C4.5 algorithm can handle numeric attributes, there is no need to discretize any of the attributes.

WEKA has implementations of numerous classification and prediction algorithms. The basic ideas behind using all of these are similar. In this project we will use the modified version of the car data to classify new instances using the C4.5 algorithm (note that the C4.5 is implemented in WEKA by the classifier class: (figure)



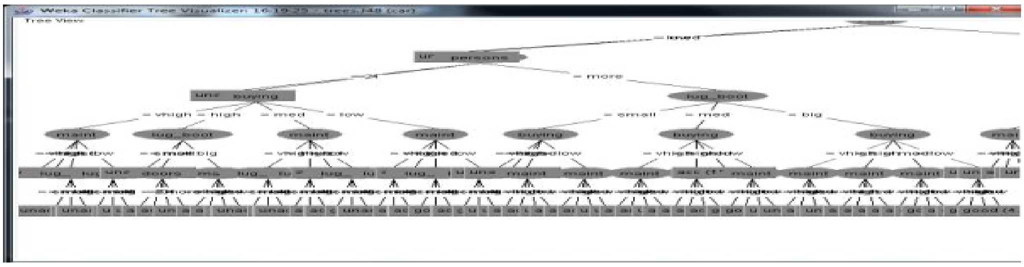
Next, we select the "Classify" tab and click the "Choose" button to select the J48 classifier, as depicted in Figures . Note that J48 (implementation of C4.5 algorithm) does not require discretization of attributes.

This is the result of using class j48 in weka classifier :



weka classifier visualize tree _j48

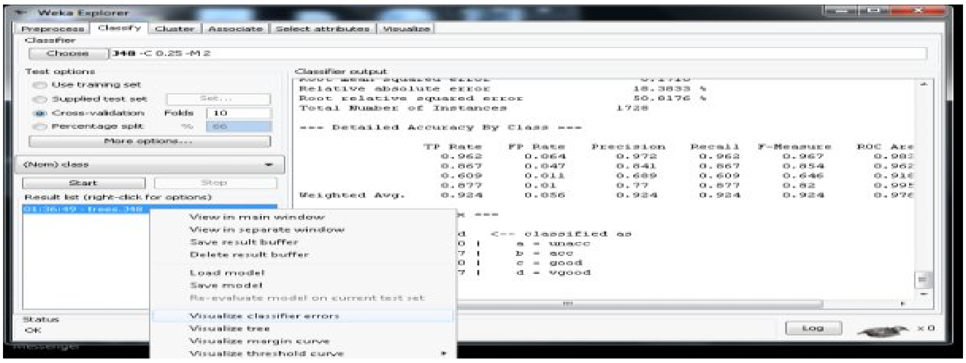
WEKA also let's us view a graphical rendition of the classification tree. This can be done by right clicking the last result set and selecting "Visualize tree" from the pop-up menu. The tree. Note that by resizing the window and selecting various menu items from inside the tree view we can adjust the tree view to make it more readable.



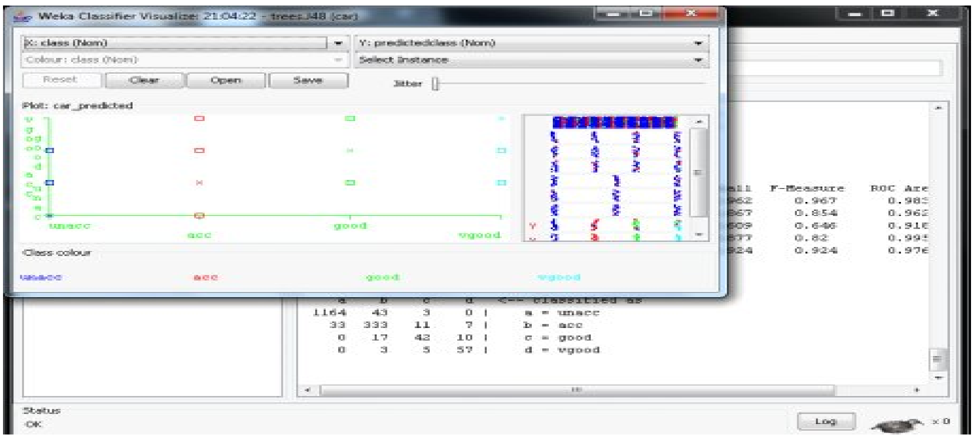
Weka classifier visualize error tree j48:

Of course, in this project we are interested in knowing how our model managed to classify the new instances. To do so we need to create a file containing all the new instances along with their predicted class value resulting from the application of the model. Doing this is much simpler using the command line version of WEKA classifier application. However, it is possible to do so in the GUI version using an "indirect" approach, as follows.

First, right-click the most recent result set in the left "Result list" panel. In the resulting pop-up window select the menu item "Visualize classifier errors". This brings up a separate window containing a two-dimensional graph. These steps and the resulting window are shown in Figures



The following figure shows the result of weka visualize error



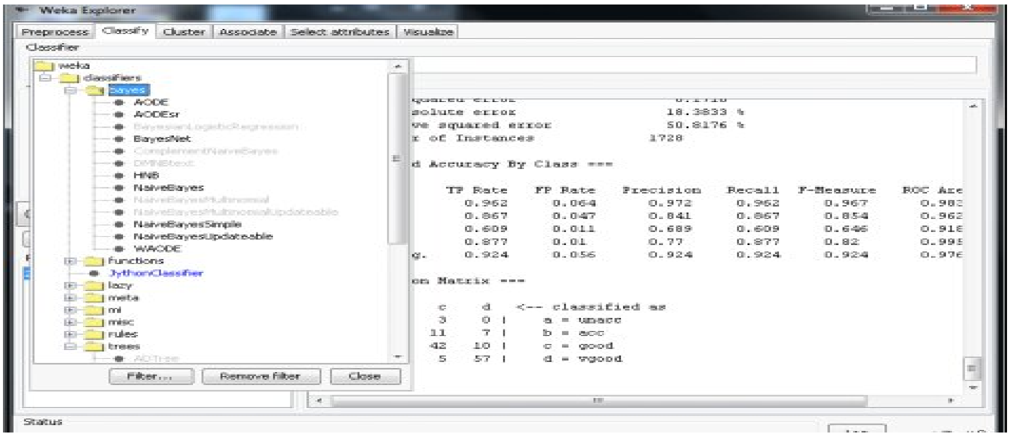
Naive Bayes Algorithm

The Microsoft Naive Bayes algorithm is a classification algorithm provided by Microsoft SQL Server Analysis Services for use in predictive modeling. The name Naive Bayes derives from the fact that the algorithm uses Bayes theorem but does not take into account dependencies that may exist, and therefore its assumptions are said to be naive.

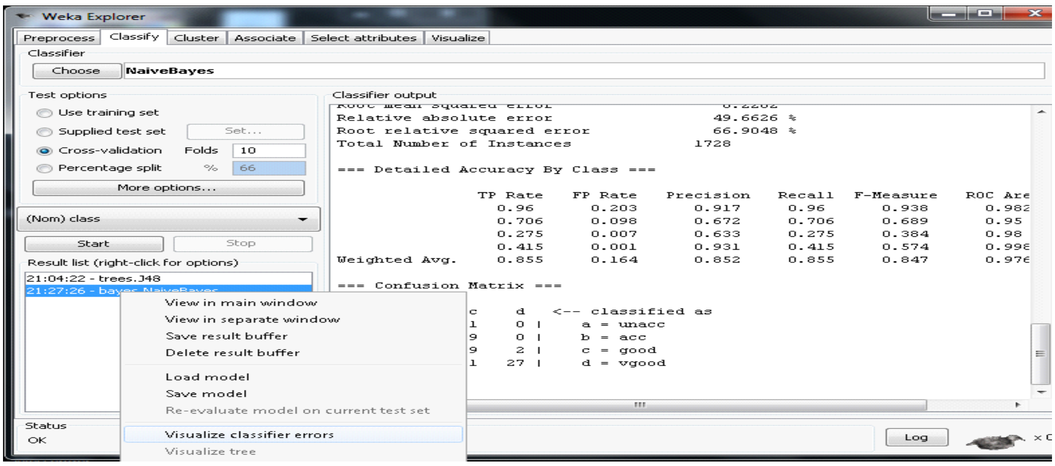
Class Naïve Bayes

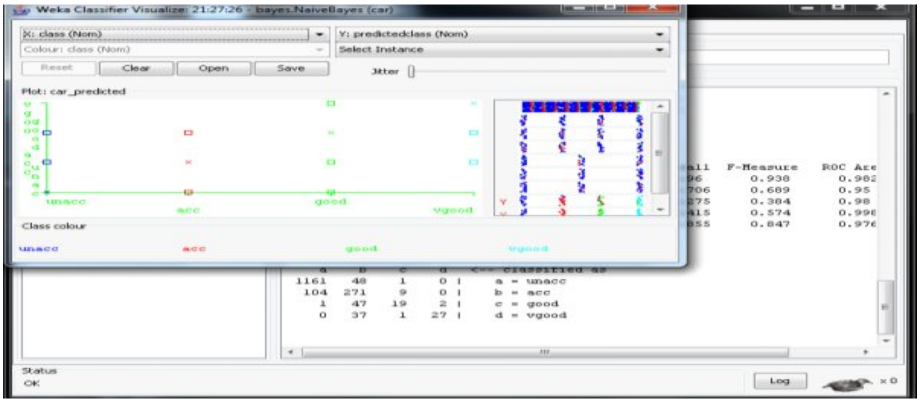
Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an Update able Classifier (which in typical usage are initialized with zero training instances) if you need the Update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

How the naive bayes algorithm used



The result of naive bayse class as the following





This figure modify the error rate:

Weka classifiers lazy Algorithm

Class LBR:

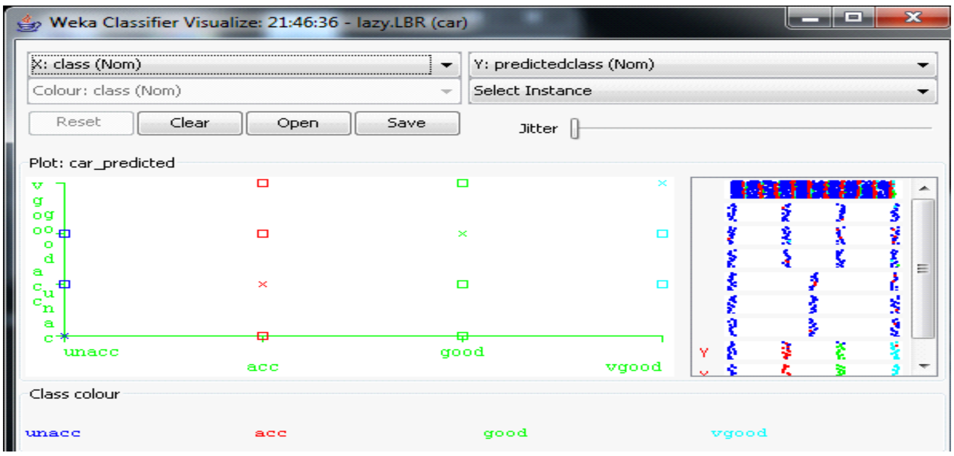
Lazy Bayesian Rules implement a lazy learning approach to lessening the attribute-independence assumption of naive Bayes. For each object to be classified, LBR selects a set of attributes for which the attribute independence assumption should not be made.

All remaining attributes are treated as independent of each other given the class and the selected set of attributes. LBR has demonstrated very high accuracy. Its training time is low but its classification time is high due to the use of a lazy methodology. This implementation does not include

caching, that can substantially reduce classification time when multiple classifications are performed for a single training set. For more information.

This is the result and visualize error of lazy

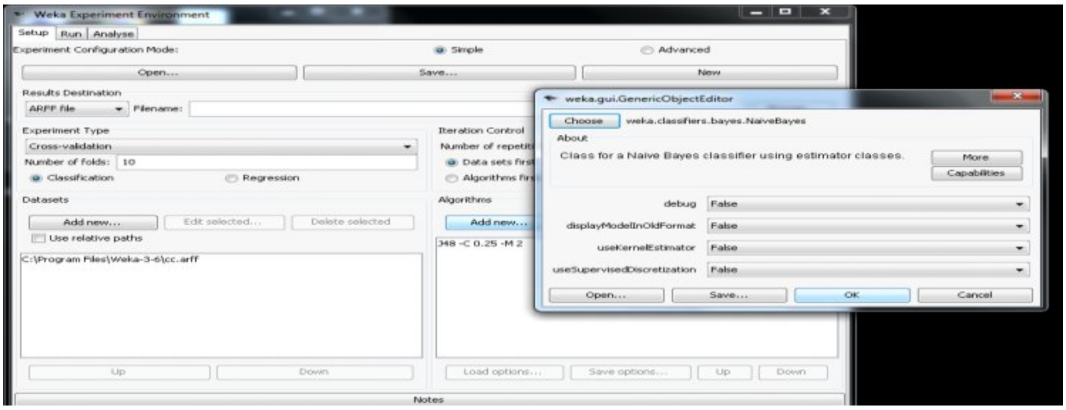
algorithm class LBR



From the results of each algorithm we can see the results And choose the best algorithm according to the T rate and F rate as the following table:

Algorithms'	Tree(J48)	naivebaye	Lazr(LBR)
TP Rate	**0.924**	**0.855**	**0.942**
FP Rate	**0.056**	**0.164**	**0.047**
Precision	**0.924**	**0.852**	**0.942**
Recall	**0.924**	**0.855**	**0.942**
F-Measure	**0.924**	**0.847**	**0.94**
Roc Area	**0.976**	**0.976**	**0.992**
Class	**acc**	**V good**	**acc**

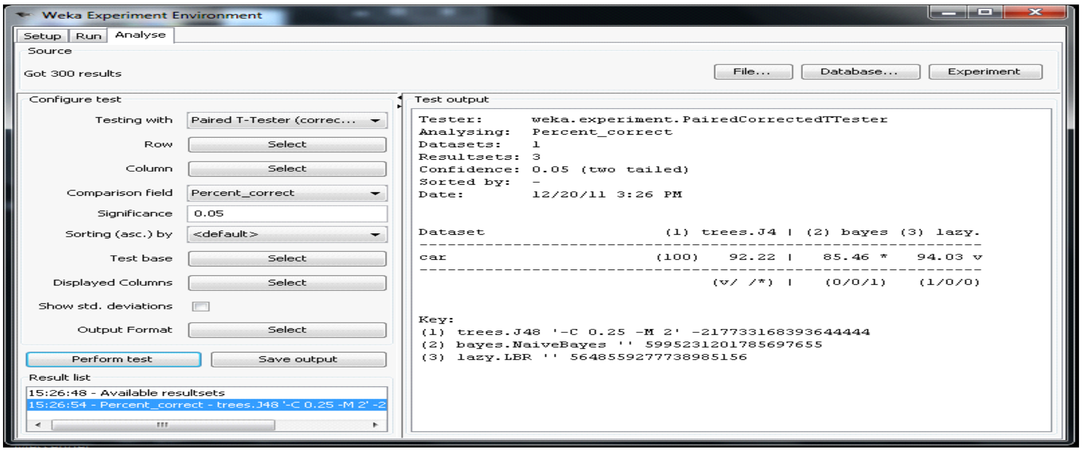
weka experimenter:



This figure shows us how apply weka experiment and how choose the algorithms that we have used .

Results discussion

This is the results of running weka experiment



when we use weka experiment it will shows us very accurate and clear results according of percent correct as a following :

Tester: weka .experiment t.PairedCorrectedTTester

Analyzing: Percent correct

Datasets: 1

Result sets: 3

Confidence: 0.05 (two tailed)

Sorted by: -

Date: 12/19/11 4:09 PM

Dataset (1) trees.J4 | (2) bayes (3) lazy.

----

car (100) 92.22 | 85.46 * 94.03 v

(v/ /*) | (0/0/1) (1/0/0)

Key:

(1) trees.J48 '-C 0.25 -M 2' -217733168393644444

(2) bayes.NaiveBayes '' 5995231201785697655

(3) lazy.LBR '' 5648559277738985156

According of weka experiment results we can say that the lazy algorithm have very good result then this algorithms is the best in comparison to others to be sure we can see to in weka classifier visualize .

weka knowledge flow Environment:

The Knowledge Flow presents a "data-flow" inspired interface to Weka. The user can select Weka components from a tool bar, place them on a layout canvas and connect them together in order to form a "knowledge flow" for processing and analyzing data. At present, all of Weka's classifiers and filters are available in the Knowledge Flow along with some extra tools.

The Knowledge Flow can handle data either incrementally or in batches (the Explorer handles batch data only). Of course learning from data incrementally requires a classifier that can be updated on an instance by instance basis. Currently in Weka there are five classifiers that can handle data incrementally: Naïve BayesUpdateable, IB1, IBk, LWR (locally weighted regression).

Features of the Knowledge Flow:

- intuitive data flow style layout
- process data in batches or incrementally
- process multiple batches or streams in parallel! (each separate flow executes in its own thread)
- chain filters together
- view models produced by classifiers for each fold in a cross validation
- visualize performance of incremental classifiers during processing (scrolling plots of classification accuracy, RMS error, predictions etc)

Components available in the KnowledgeFlow:

Evaluation:

- Training Set Maker - make a data set into a training set
- Test Set Maker - make a data set into a test set
- Cross Validation Fold Maker - split any data set, training set or test set into folds
- Train Test Split Maker - split any data set, training set or test set into a training set and a test set
- Class Assigner - assign a column to be the class for any data set, training set or test set
- Class Value Picker - choose a class value to be considered as the "positve" class. This is useful when generating data for ROC style curves (see below).
- Classifier Performance Evaluator - evaluate the performance of batch trained/tested classifiers
- Incremental Classifier Evaluator - evaluate the performance of incrementally trained classifiers

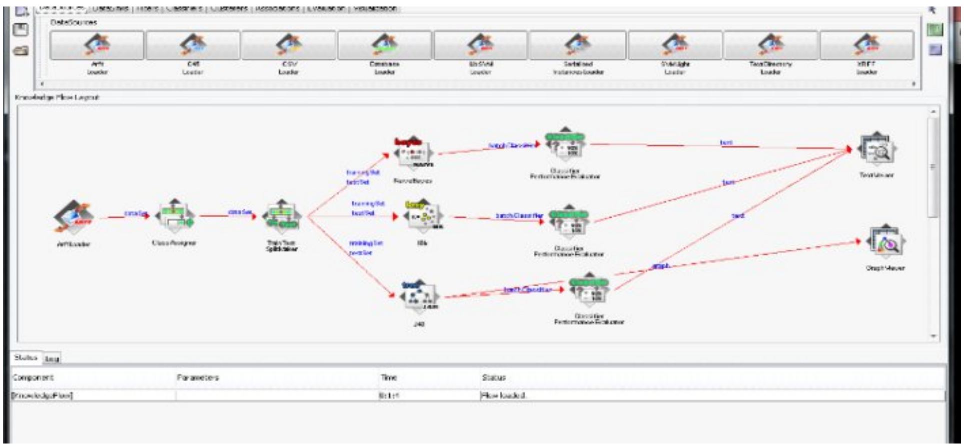
- Prediction Appended - append classifier predictions to a test set. For discrete class problems, can either append predicted class labels or probability distributions.
- Visualization:
- Data Visualized - component that can pop up a panel for visualizing data in a single large 2D scatter plot
- Scatter Plot Matrix - component that can pop up a panel containing a matrix of small scatter plots (clicking on a small plot pops up a large scatter plot)
- Attribute Summarizer - component that can pop up a panel containing a matrix of histogram plots - one for each of the attributes in the input data
- Model Performance Chart - component that can pop up a panel for visualizing threshold (i.e. ROC style) curves.
- Text Viewer - component for showing textual data. Can show data sets, classification performance statistics etc.
- Graph Viewer - component that can pop up a panel for visualizing tree based models
- Strip Chart - component that can pop up a panel that displays a scrolling plot of data (used for viewing the online performance of incremental classifiers)

  Filters: All of Weka's filters are available

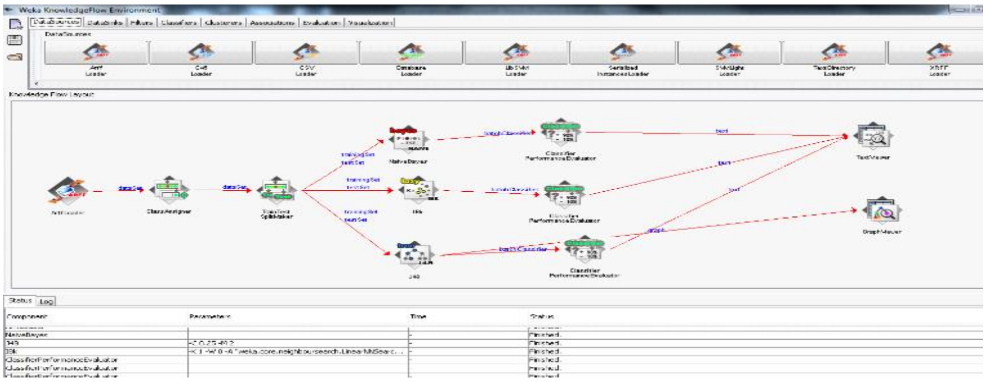
  Classifiers: All of Weka's classifiers are available

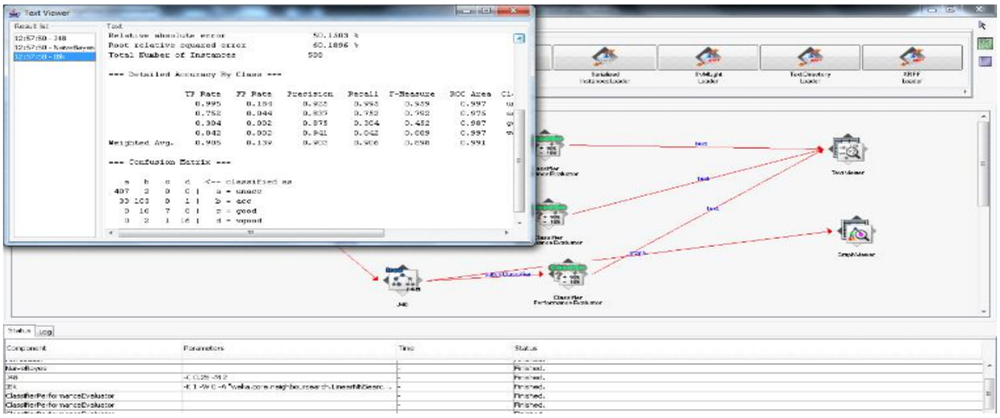
  Data Sources: All of Weka's loaders are available

  when we applied knowledge flow using weka classifier algorithms it shows us this results

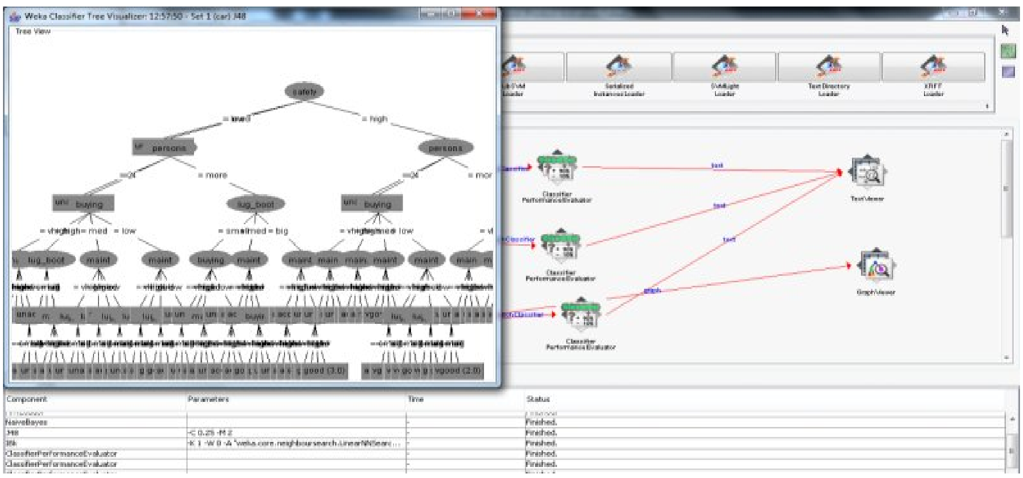


when we make start loading it will shows us this figure and results ,because all classifier algorithms are finished and we can see that no error in our results .





from the text results we can say that the lazy algorithm has very good result then this algorithms is the best in comparison to the others.



This figure shows us the graph result of using j48

References :

Data mining:practical Machin learning Tools and Technique with java implementation ,by l.H.witten and E.frank ,morgan kanfmann publisher ,2000.

Gupte,S,Masoud ,O,Martin ,R.F.K.Papanikolopoulos ,N.P.Detection and Classifica-tion.

D.Michie ,Methodologies from Machine Learning in Data analysis and Software ,Computer Journal ,Vol .34,No.6,1991,pp.559-565.

N.Kerdprasop,and K. Kerdpraso,"Moving data mining tools toward a business intelligence system ",Enformatika ,,Vol.19,pp.117-122,2007.

Yoshida,T.,Mohottala,S.,Kagesawa , M.,lkeuchi,K.:Vehicle Classification System with Local-Feature Based Algorithm using CG Model images. IEICE Trans.,Vol.E00-A,No.12,December 2002.

www.thesai.org/.../Paper%204-... - United State

www.boirefillergroup.com/....KDD_CONFERENE_
PAPER_AUG2006.pdf

www.dcc.fc.up.pt/~ricroc/aulas/0708/atdmlp/material/paper_dmbiz06.pdf

www.ecmlpkdd2006.org/ws-pdmaec.pdf

http://www.linkedin.com/in/federicocesconi

www.linkedin .com/in/federicocesconi

www.eecs.northwestern.ed/~yingliu/papers/pdcs.pdf

_www.eecs.northwestern.ed/~yingliu/papers/pdcs.pdf

www.ics.**uci**.edu/~mlearn/

http://www.cs.waikato.ac.nz/ml/weka/