International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 37 العدد
Part 1 المجلد
المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

http://www.doi.org/10.62341/NAER1907

Received	**2025/08/20**	**تم استلام الورقة العلمية في**
Accepted	**2025/09/16**	**تم قبول الورقة العلمية في**
Published	**2025/09/18**	**تم نشر الورقة العلمية في**

Analyzing the Efficiency of a Data Mining Dataset in Weka Implementing an Automotive Dataset

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

Abstract

The Car manufacturing sector represents a major focus in the development of the automotive industry. In this research paper, a proposed data mining application for the automotive manufacturing sector is explained and tested. The dataset was retrieved from the machine learning repository at the University of California, Irvine. This research paper aims to create a more reliable classifier for future object classification. Classification is an important technique in data mining. It is a supervised learning process that involves classifying an object into one of the predefined classes based on its attributes. In this paper, we use a large database containing 7 attributes and 1,728 instances. We compare the results of a simple classification technique (using the J48 decision tree inference algorithm and MONK) with results based on different parameters using WEKA (Waikato Environment Knowledge Analysis), a data mining tool. The results of the experiment show a comparison between three algorithms to see which is the best and least error-prone algorithm. The physical characteristics of a car viz . Engine location ,price, how many doors, stroke, city fuel consumption, and other factors determine a vehicle's performance. Therefore, developing such a classification, although a huge undertaking, is absolutely essential in the car industry. Machine learning techniques can help integrate computer-based systems to predict vehicle quality and improve system efficiency. Classification models were trained using 214 datasets. The predicted values of 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.

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 37 العدد
Part 1 المجلد
http://www.doi.org/10.62341/NAER1907

تحليل فعالية مجموعة بيانات للتنقيب في البيانات في برنامج Weka باستخدام بيانات قطاع السيارات

نجاة عبد الله الصيد رحيل
كلية الآداب والعلوم بدر جامعة الزنتان – ليبيا
nagatrahil08@gmail.com

الملخص

يُعَدّ قطاع تصنيع السيارات من أبرز المحاور في تطوير صناعة المركبات. تتناول هذه الورقة البحثية تطبيقًا مقترحًا لتنقيب البيانات موجهًا لصناعة السيارات، حيث تم اختباره اعتمادًا على مجموعة بيانات مستخرجة من مستودع تعلم الآلة بجامعة كاليفورنيا، إيرفين. الهدف الرئيس من هذا البحث هو بناء مصنّف أكثر دقة وموثوقية لتصنيف الكائنات في المستقبل.التصنيف يُمثل إحدى أهم تقنيات تنقيب البيانات، وهو أسلوب تعلم تحت الإشراف يهدف إلى إسناد الكائن إلى إحدى الفئات المحددة مسبقًا وفقًا لخصائصه. في هذه الدراسة، استُخدمت قاعدة بيانات تضم 7 سمات و1,728 حالة. جرت مقارنة أداء خوارزمية بسيطة للتصنيف (شجرة القرار J48 وخوارزمية MONK) مع نتائج مبنية على معايير مختلفة باستخدام أداة WEKA (Waikato Environment for Knowledge Analysis)الخاصة بتنقيب البيانات. وأظهرت النتائج مقارنة بين ثلاث خوارزميات لتحديد الأكثر كفاءة والأقل عرضة للخطأ.إن الخصائص الفيزيائية للسيارة مثل: موقع المحرك، السعر، عدد الأبواب، الشوط، واستهلاك الوقود في المدينة، تُعَدّ عوامل أساسية تؤثر على أداء المركبة. ومن ثَمّ، فإن تطوير نظام تصنيف من هذا النوع، رغم تعقيده، يُمثل ضرورة في صناعة السيارات. يمكن لتقنيات تعلم الآلة أن تُسهم في دمج أنظمة حاسوبية للتنبؤ بجودة المركبات وتحسين كفاءة الأداء. تم تدريب نماذج التصنيف على 214 مجموعة بيانات، مع تقييم النتائج عبر أسلوب التحقق المتقاطع(10-fold cross-validation) ، ومن ثم تمت المقارنة بين النتائج.

الكلمات المفتاحية :تنقيب البيانات، تقنيات تعلم الآلة، J48، أشجار القرار، صناعة السيارات، WEKA

http://www.doi.org/10.62341/NAER1907

1. Introduction

What we see around us in our world is full of data. After compilation and organization, data, if we are lucky, becomes information. Information in today's networked world gets stored digitally and is provided in real time [1]. The issue lies in comprehension, assimilation, and use of this information to extract useful knowledge [2].Therefore; we need technologies to help us get through this ocean of data.

Data mining is an analysis method [3]. It is designed to explore data (large amounts of data - market or business focused) in search of recurring patterns and/or systematic relationships between all variables, and subsequently test the findings by applying the discovered patterns to new subsets of the data. The ultimate goal of data mining is prediction - and prediction data mining is the most common type of data mining and the easiest to apply commercially [4].

To begin with, in our paper we will first define our datasets and explain each of the algorithms we employed, what they do, and how we have implemented them. Then we shall present the results as well as the comparison between all the algorithms we employed.

Data Mining using WEKA.This manual/report adheres to an extensive template outlining data extraction and preprocessing tasks to be performed using WEKA. This manual is prepared for WEKA version 3.6. Interface components and modules may have changed in more recent versions of WEKA. We need to download the latest version of WEKA from the official WEKA site. The new release includes some additional GUI features and a wider package structure for the Java packages. Take note of these differences while you continue with the guide. Package structure differences are particularly significant when running Weka from the command line.

2. Definition of WEKA

Developed in New Zealand at UNIV of Waikato [5].

A collection of state-of-art machine earning algorithms and data pre-processing tools.

Provide implementation of:

- Regression
- Classification
- Clustering
- Association rules
- Feature selection

http://www.doi.org/10.62341/NAER1907

Figure 1 : WEKA 3.6: data mining software in java

WEKA is a suite of machine learning algorithms to carry out data mining tasks. WEKA includes data preprocessing filters, association rule, regression, clustering, classification, and data visualization tools (figure 1).

3. Data Preprocessing in WEKA

The following tutorial is for WEKA 3.6. Any other material associated with WEKA, like datasets, we can obtain from the official WEKA webpage.

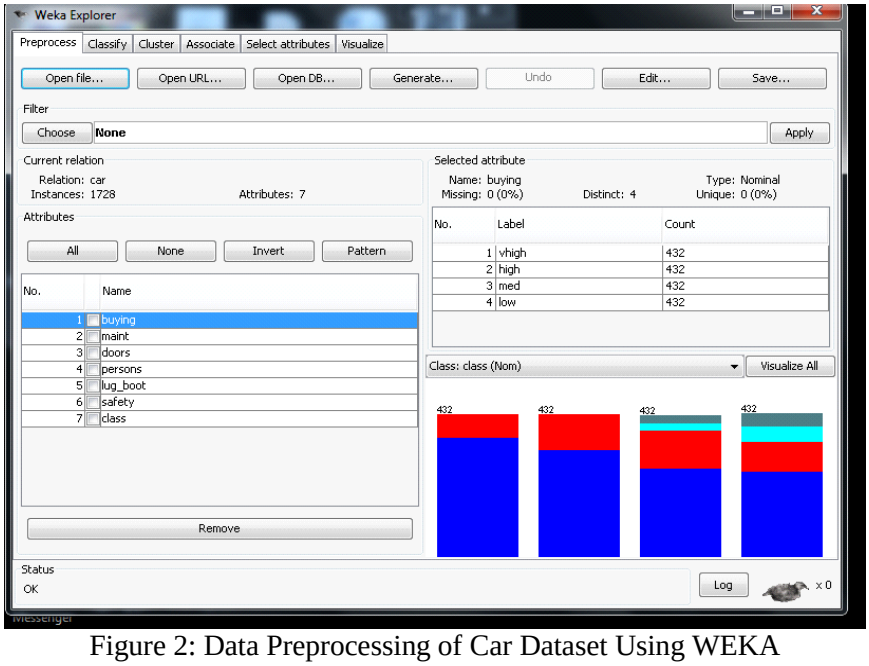

Figure 2: Data Preprocessing of Car Dataset Using WEKA

http://www.doi.org/10.62341/NAER1907

This project demonstrates the main data preprocessing which performed by WEKA (figure 2). We used sample indicated dataset in this project, like "car data" available in the arff format (car.arff). The data contains the below field (table 1).

Table 1: The Dataset of Car

Data Set Feature:	Multivariate	**Number of cases:**	**1728**	**Area:**	**N/A**
Attribute Features:	Conclusive	**Number of Attributes:**	**7 with class**	**Donation date**	**1997-06-01**
Associated Tasks:	Classification	**Lost Values?**	**No**	**Number of Web Hits:**	**112752**

4. Data Set Information

- **Attribute Information**

Class Values:

un-acc, acc , good , v-good

some details about car dataset:

1- Heading: The Car Estimate Database

2- Source:

(a) Creator: Marko Bohanec

(b) Donors: Marko Bohanec (marko.bohanec@ijs.si)
Blaz Zupan (blaz.zupan@ijs.si)

(c) Date: June, 1997

3- Previous use:

Hierarchical model of accuracy, from which the dataset has been derived, was firstly introduced in the paper by V. Bohanic and M. Rajkovic: Knowledge acquisition and interpretation for multi-attribute decision generation. In the 8th International Workshop on Skillful Systems and Their Applications, Avignon, France. pages 59-78, 1988. In machine learning research, this data set has been employed to verify the HINT (Hierarchical Induction Tools), which could reconstruct the main hierarchical structure in full. This and approchement to C4.5 is described in J. Demsar ,B. Zupan, M. Bohanec, B. Zupan, I. Bratko,: Machine Learning with Functional Analysis. ICML-97, Nashville, TN. 1997 (forthcoming).

4 - The automobile estimate database was token from a hierarchical decision model built in an effort to demonstrate DEX,

http://www.doi.org/10.62341/NAER1907

M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). According to the following concept structure,

The model will judge automobiles

- Car acceptance
- Price total price
- Purchase price
- Maintenance price.
- Technical characteristics TECH.
- Comfort COMFORT.
- Number of doors - doors.
- The ability of people to conceive.
- Lug_boot volume.
- Safety Estimate the safety of the car.

The input attributes are displayed in lower case letters:

The target idea (car) is a three-concept-intermediate model consisting of comfort ,technology, and price .All three concepts are part of the basic model .The car rating database contains instances with constructional information eliminated, hence it directly relates CAR to(6 input attributes): purchase, maintenance, doors, people, trunk, and safety.

With the main concept structure established, the database can particularly be useful to test constructive induction and structure discovery techniques.

5- The number of cases : (1728)
(cases cover the entire attribute space)

6-Number of the attributes :(6)

7-Values of attribute:

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- Losing the values of attribute - none

9- Class Division (Number of cases per class)

Class----- N ------- N[%]

V- GOOD -------- 65 (3.762 %)
GOOD---------- 69 (3.993 %)
ACC------- -------- 384(22.222 %)

http://www.doi.org/10.62341/NAER1907

UNACC----------1210 (70.023 %)

The data set Information below:

Type of class - Nominal

Index of class - Last

- **This part of the car dataset**

#relation car
#Attribute- buying {V-HIGH, HIGH,MED,LOW}
#Attribute- maint {V- HIGH, 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
V- HIGH,V- HIGH ,2 , 2 , low -small- unacc
V- HIGH,V- HIGH,2,2, med,small,unacc
V- HIGH,V- HIGH,2,2,small, high,unacc
V- HIGH,V- HIGH,2,2, low-med-unacc
V- HIGH,v- HIGH,2,2,med-unacc- med
V- HIGH,v- HIGH,2,2,med -unacc- high

5. Decision tree algorithm

For categorical attributes, the algorithm forecasts based on the predictability of the association among the input columns in the data. It forecasts the states of the column that it finds to be predictable using the values of these columns, or states. That is, the algorithm finds the input columns associated with the predictable column [6].

5.1. The J48 Decision Tree Induction Algorithm and MONK

The J48 algorithm used by WEKA and MONK is an implementation of the famous C4.5 algorithm already developed by J. Ross Quinlan [7]. The most representative form of machine learning algorithm information is the decision tree, which provides the fastest and easiest way to represent data structures [8].

5.2. Classification Via Decision Trees in WEKA

This project demonstrates the utilization of the C4.5 (J48) classifier in WEKA. The sample data set employed by this project, in database accessible in car.arff This report presumes appropriate data preprocessing has been accomplished. In this status the ID field is removed. After the algorithm of C4.5 was ready to handle the numeric attributes, thus nothing needed to partition any of these attributes.

http://www.doi.org/10.62341/NAER1907

WEKA has many applications for prediction and classification rules. The overall ideas to utilize it are similar. An improved version of the vehicle data will be used in this project to predict new cases using the C4.5 algorithm (figure 3).

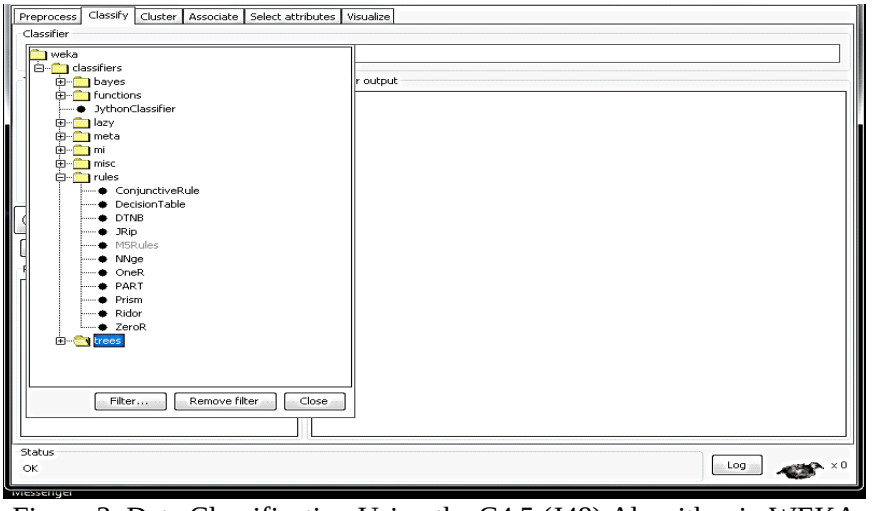

Figure 3: Data Classification Using the C4.5 (J48) Algorithm in WEKA

Then we choose the "Classification" category then click on the "Select button" to choose J48 classifier as shown in the image. Notice we say J48 (the C4.5 algorithm implementation) does not need to attribute splitting.
This is the outcome of employ class j48 in WEKA classifier in figure 4:

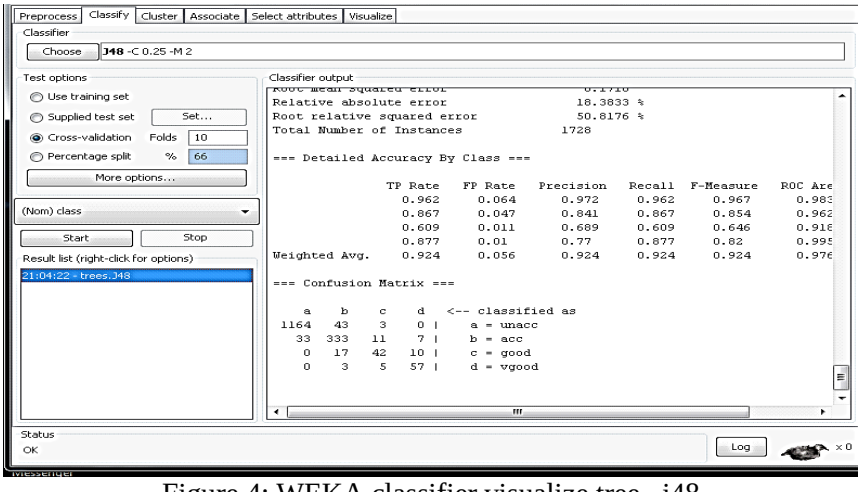

Figure 4: WEKA classifier visualize tree _j48

http://www.doi.org/10.62341/NAER1907

Let's also see a graphical representation of the classification tree. We can do this by right-clicking on the final result set and from the top of the menu, clicking on "Visualize Tree." Notice that by resizing the size of the window after choosing the various menu items from within the tree view, we can resize the presentation of the tree to give us a more readable copy (figure 5).

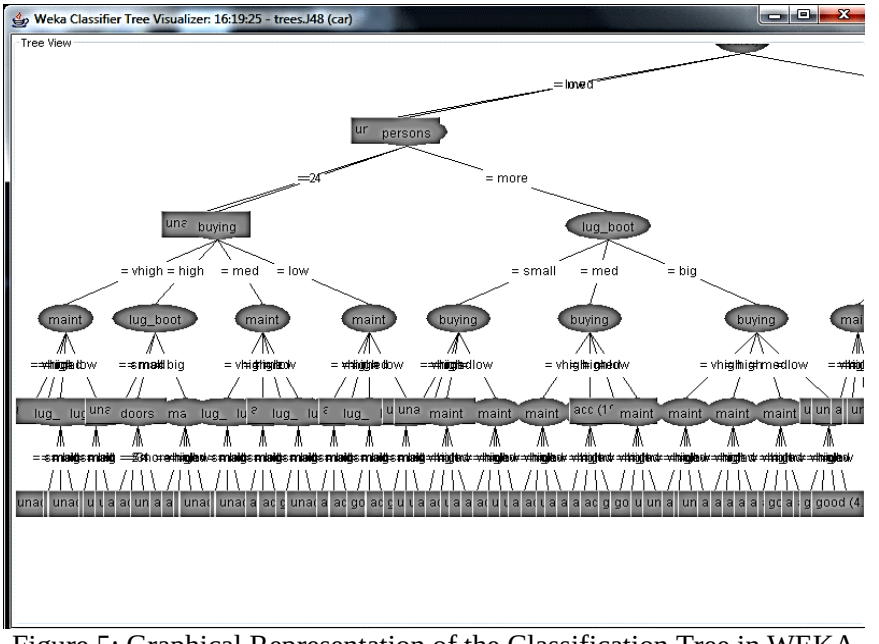

Figure 5: Graphical Representation of the Classification Tree in WEKA

5.3. Weka classifier visualize error tree j48

Naturally, in this assignment, we are interested in how our model predicts new cases. For this, we want to have a file with all new cases and their predicted class values by running the model. This is quite straightforward to do using the command line version of the WEKA classification tool. However, It can be done in the GUI-version using an "indirect" approach, as below.

To start, right-click the most recent set of results in the left-hand "Results List" panel. In the pop-up window that is displayed, click on the "View Workbook Errors" menu item. A new window containing a two-dimensional graph will be displayed. These steps and the window that is displayed are shown in Figure 6.

http://www.doi.org/10.62341/NAER1907

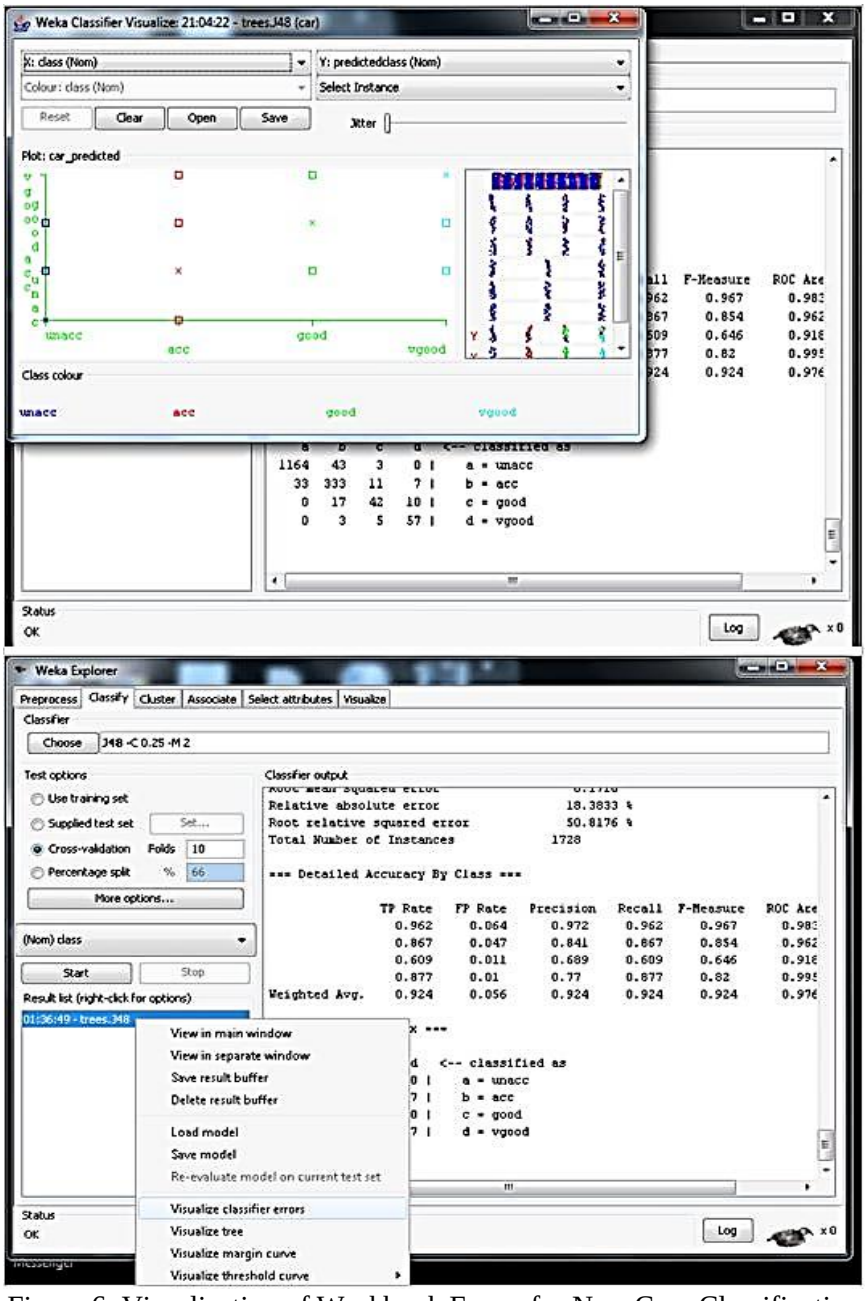

Figure 6: Visualization of Workbook Errors for New Case Classification in WEKA

6. Naive Bayes Algorithm

Microsoft Naive Bayes is a classification algorithm provided by Microsoft SQL Server Analysis Services for use in predictive modeling. It is referred to as Naive Bayes since the algorithm uses

http://www.doi.org/10.62341/NAER1907

Bayes' theorem without considering possible dependencies, hence the assumptions are naïve [9].

6.1. Class Naïve Bayes

A simple Bayesian classifier that uses estimator classes. The accuracy values of the numerical estimator are chosen based on an analysis of the training data. For this reason, this classifier is not an updatable classifier (which is typically initialized without any training instances). If you need an updateable classifier, use the Naive Bayes updateable classifier. A classifier capable of updating Naive Bayes will use a default precision of 0.1 for numerical properties when the constructor classifier is called without any training [10].

6.2. Naïve Bayes Algorithm Explained

How does the Naive Bayes algorithm work?

The Naïve Bayes algorithm is based on Bayes' Theorem, where the probability of a sample belonging to a certain class is calculated based on the values of its features, assuming that the features are independent from each other (which is why it is called "Naïve") [11].

- The probability of each class is first calculated from the training data.
- When new data is introduced, the probabilities of it belonging to each class are computed using these prior values.
- Finally, the sample is classified into the class that achieves the highest probability (figure 7).

http://www.doi.org/10.62341/NAER1907

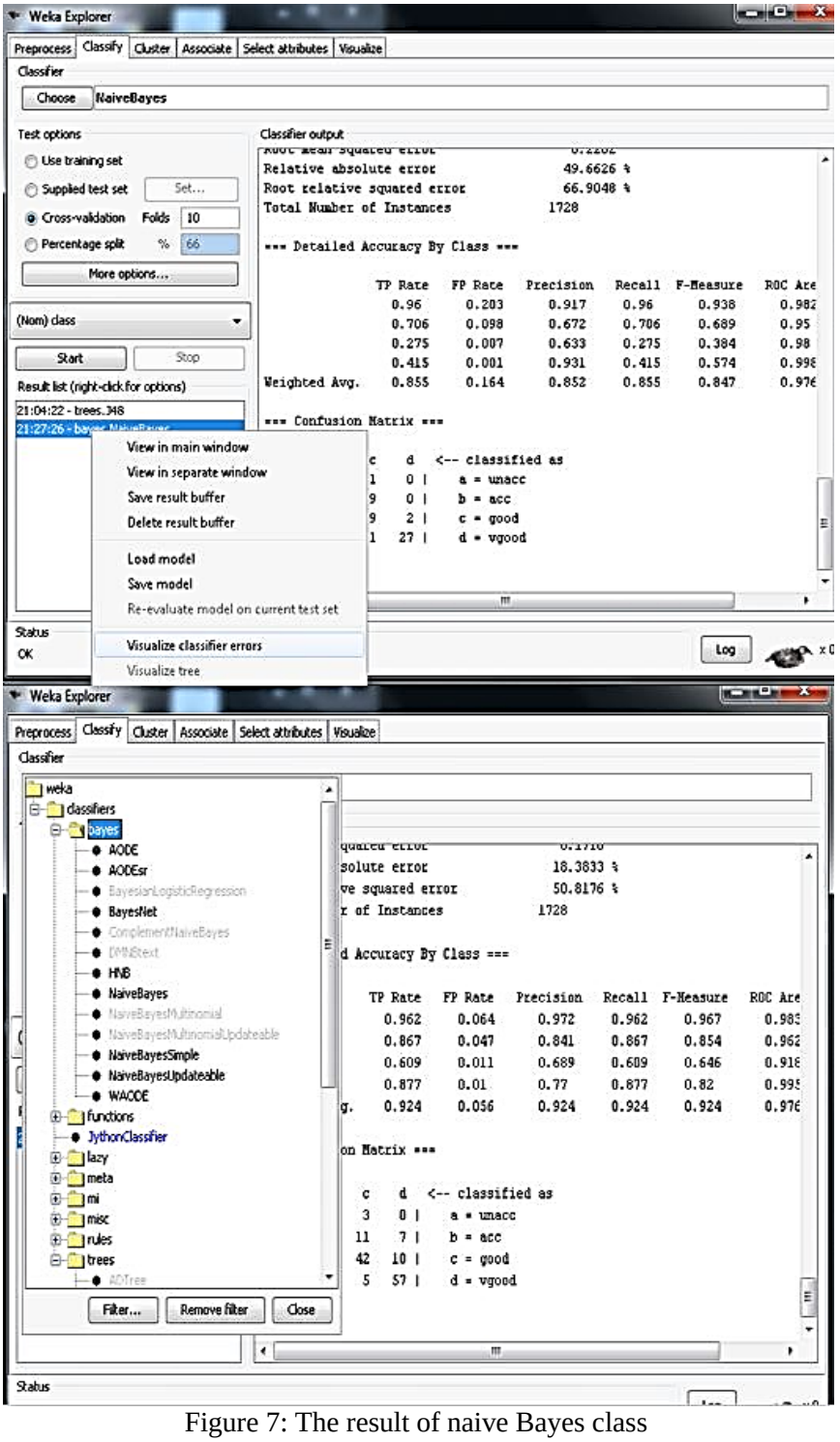

Figure 7: The result of naive Bayes class

http://www.doi.org/10.62341/NAER1907

The figure 8 modify the error rate:

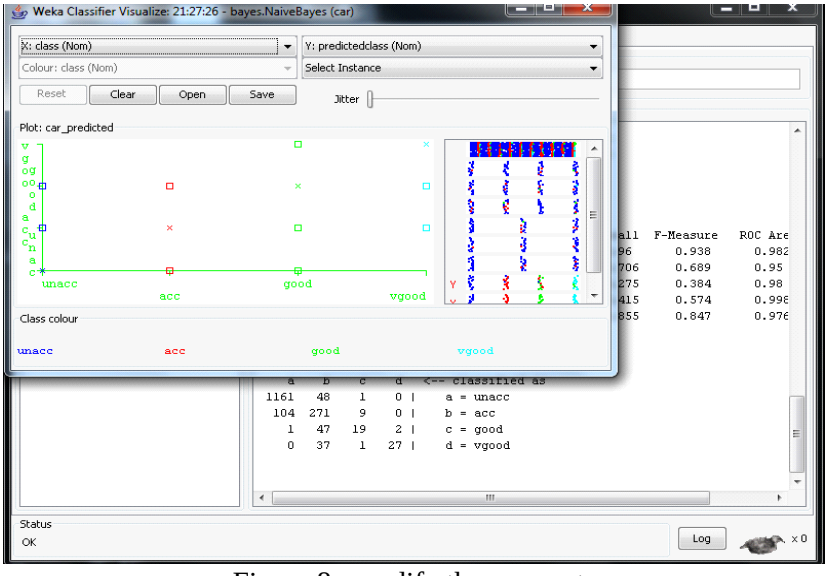

Figure 8 : modify the error rate

7. WEKA classifiers lazy Algorithm Class LBR

Lazier Bayes rules apply a lazy learning interface to reduce the assumption of feature independence for naive Bayes rules. LBR chooses a set of attributes for which attribute independence should not be assumed for any object to be classified [12].

All other attributes will be treated as independent between the selected attribute set and class. LBR has shown to have excellent accuracy [13]. Its training time is low and classification time is high due to the fact that it uses a passive strategy [14]. This does not count the utilization of caching, which has the advantage of saving significantly on classification time while performing several classifications of the same training set [15]. For additional information, see:

The figure 9 show the result and visualize error of lazy algorithm class LBR.

http://www.doi.org/10.62341/NAER1907

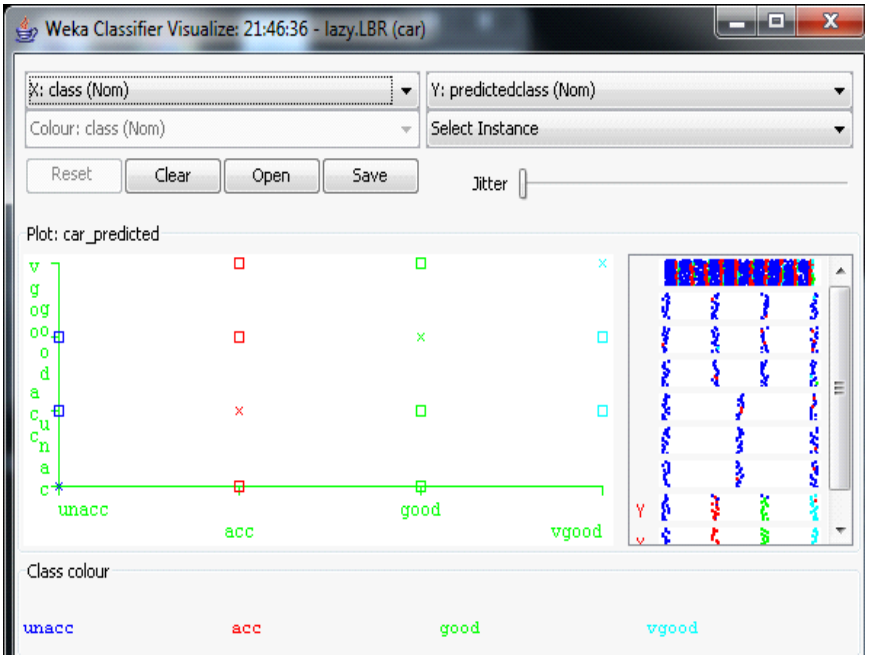

Figure 9: the result and visualize error of lazy algorithm class LBR

8. Comparison of Algorithms Based on T-rate and F-rate

From the outcomes of each algorithm we can see the outcomes and select the best algorithm according to the T rate and F rate as the following table 2:

Table 2 : Comparison of Algorithms

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

http://www.doi.org/10.62341/NAER1907

9. Results discussion

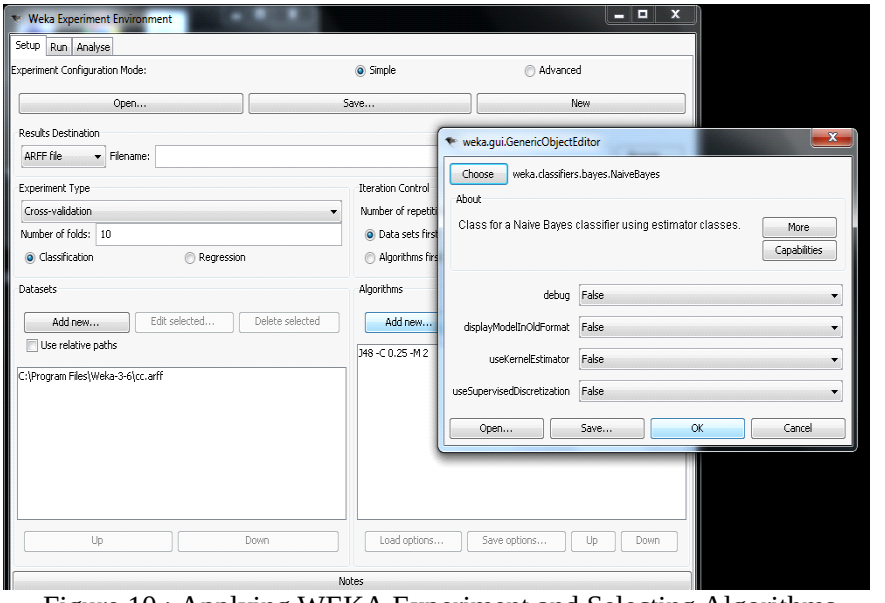

Figure 10 : Applying WEKA Experiment and Selecting Algorithms

The figure 10 shows us how apply weka experiment and how choose the algorithms that we have used.
This is the results of running WEKA experiment (figure 11) :

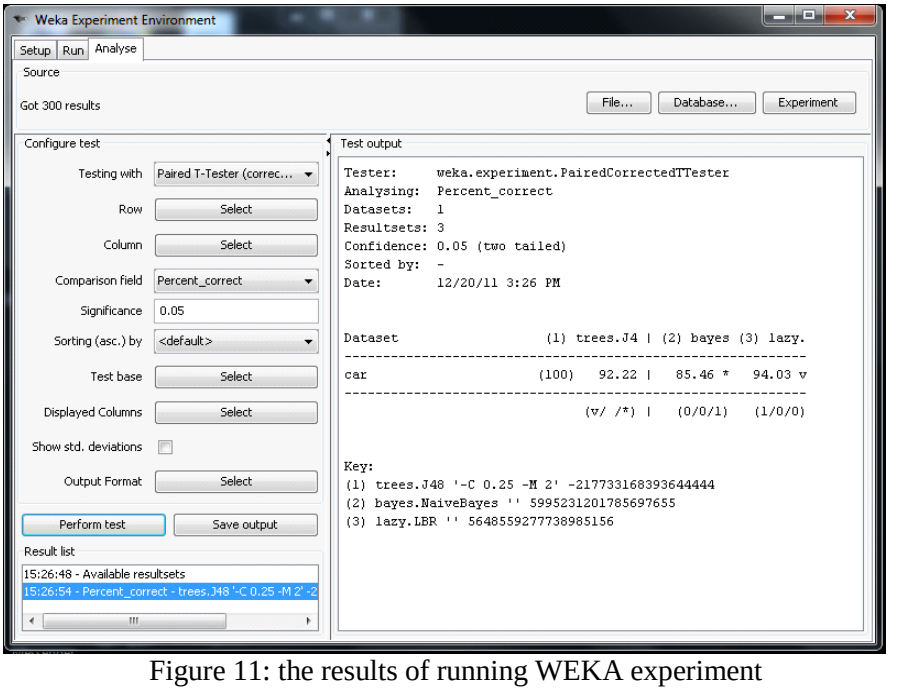

Figure 11: the results of running WEKA experiment

http://www.doi.org/10.62341/NAER1907

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

A test : WEKA. experiment t. Paired Corrected T Tester
Analyzing: Percent correct
Datasets: 1
Result sets: 3
Dependability : 0.05 - (two- tailed)
Arranged by :
Date: 12/19/11 6:09 AM
Dataset: (1) bayes | (2) lazy (3) trees.J4.

car	(100)	92.22	85.46 *	94.03 v

(v/ /*) | (0/0/1) (1/0/0)
Key:
(1) trees-J48 ' -M 2' - C 0.25 - (217733168393644444).
(2) bayes-Naive-Bayes -(5995231201785697655).
(3) lazy-LBR (5648559277738985156)

Based on WEKA experiment result we can say that the lazy algorithm have very good result then this algorithms is the best among others to make us sure we can view to in WEKA classifier visualize (figure 12).

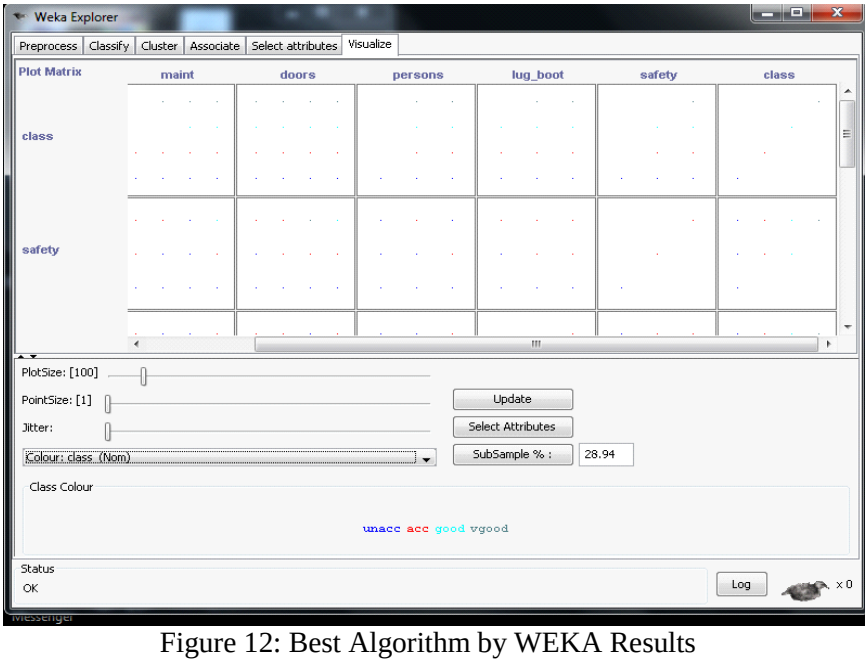

Figure 12: Best Algorithm by WEKA Results

http://www.doi.org/10.62341/NAER1907

10. WEKA knowledgment Inflow Environment

Knowledge - Flow provides WEKA with a dataflow interface. One can choose WEKA's elements from the toolbar, Put them on a sketchpad, and connect them form (Knowledgment Inflow) for processing and analyzing data. All of WEKA's filters and classifiers are now in the (Knowledgment Inflow) with some other tools.

Also it can process data incrementally or in batches (The explorer processes the data one by one). Therefore, incremental learning from data needs a classifier that can make progress on a case-by-case basis. this time WEKA has five classifiers it can process data progressively : LWR (locally weighted regression), Naïve Bayes Updateable, IB1, IBk.

Advantages of the knowledgment inflow:

- axiomatic data flow pattern planning.
- process data in gradually or batches.
- Process a lot of streams or batches in parallel! (every sporadic flow is executed by its own - line.)
- Series filters together.
- Display the models which generated by the classifiers for every tuck in validation process.
- During processing, conceive the work of incremental classifiers (scroll charts of classification predictions, RMS error ,accuracy, etc).

The Components Available Within Knowledgment Inflow:

Evaluation:

- Training Set Builder - Convert the dataset in the training set.
- Test Set Maker - Convert dataset to test set.
- Cross-validation tuck generator - divided any training set ,test set or dataset, into tucks.
- Train Test Split Maker - split any test set ,data set or training set into a test set and training set.
- Class assigner will assign the column to give a class for each training set ,data set or test set .
- Class Value Picker - select the class value then it considered as "positive" class. This is beneficial for generating data when ROC style turning (see below).

http://www.doi.org/10.62341/NAER1907

- Classifier Performance Evaluator - evaluate the efficiency of batch tested/ trained classifiers.
- Incremental Classifier Evaluator - evaluate the efficiency from increasingly trained classifiers.
- Prediction Appended - append the classifier predictions for the test set . For separate classes issues , either can append prophesy probability distributions or class labels .
- Visualization:
- Data Visualized - a component will display a panel to visualizing data in one large 2D scatter-plot.
- Scatter-plot Matrix - a component can show a panel containing the matrix of small scatter-plots (the click the a small plot will show a large scatter-plot).
- Attribute Summarizer - a component wich can display a panel including an array of the histograms a plot for every attribute within input data.
- Model Performance Chart - A component wich can display a panel to illustrate the threshold (i.e. ROC style) curves.
- Text Viewer - A component for showing text data that can show the classification performance statistics ,data sets etc.
- Graph Viewer - a component wich can display a panel to visualizing the tree based forms.
- Bar Chart - A component that can show a panel displaying a scrolling chart of data (used to display the performance of growing classifiers online).

Filters: are available in every Weka's filters.

Classifiers: are available in every Weka's classifiers.

Data Sources: are available in every Weka's loaders .

When we applied knowledge flow using weka classifier algorithms it shows us this results in figure 13:

http://www.doi.org/10.62341/NAER1907

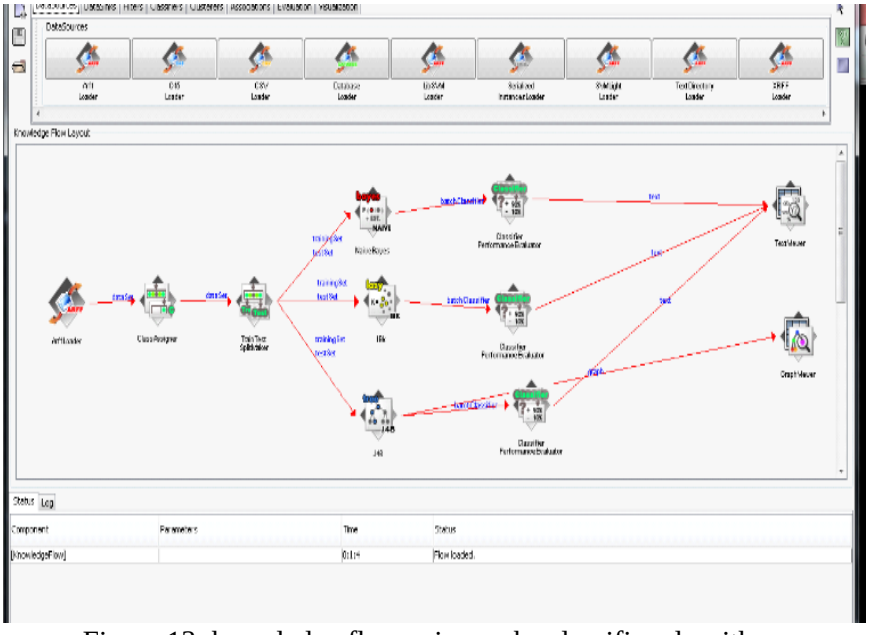

Figure 13: knowledge flow using weka classifier algorithms

When we make start loading it will shows us this figure 14 the results, because all classifier algorithms are finished and we can see that no error in our results.

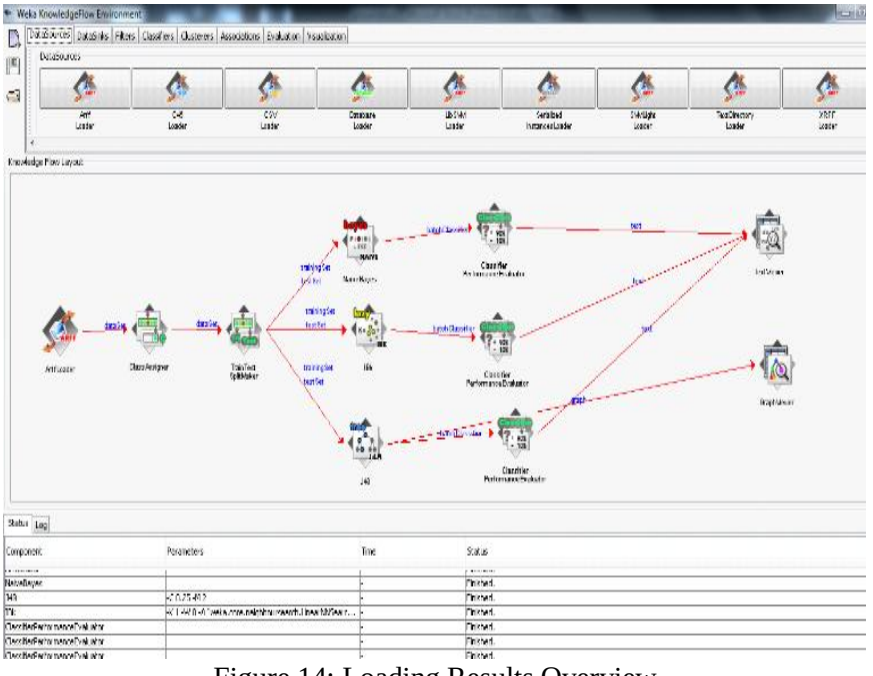

Figure 14: Loading Results Overview

http://www.doi.org/10.62341/NAER1907

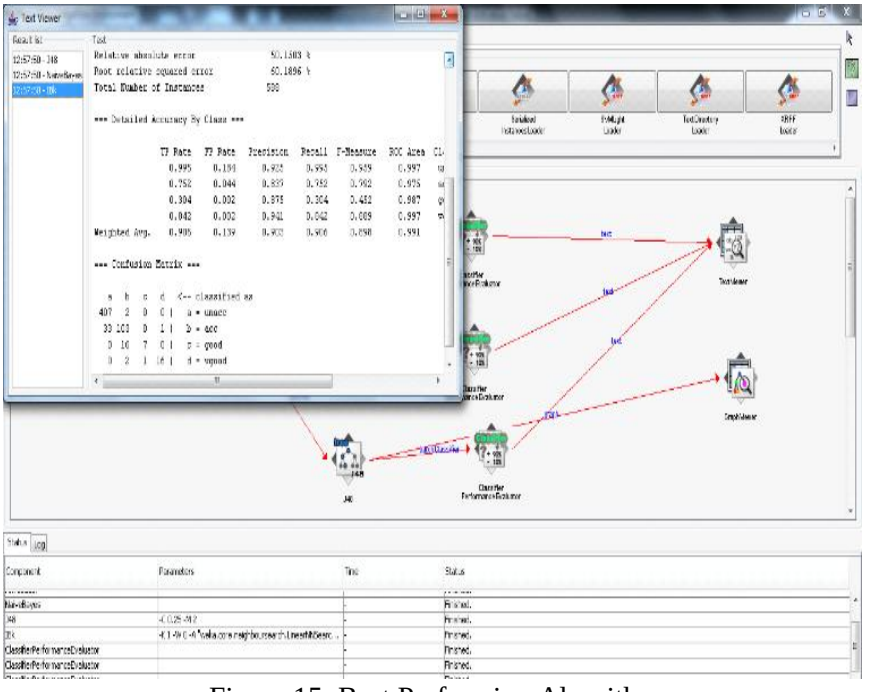

Figure 15: Best Performing Algorithm

From the text results we can explain that the lazy algorithm has very good result then this algorithms is the best in comparison to the others (figure 15).

The figure 16 shows us the graph result of using j48:

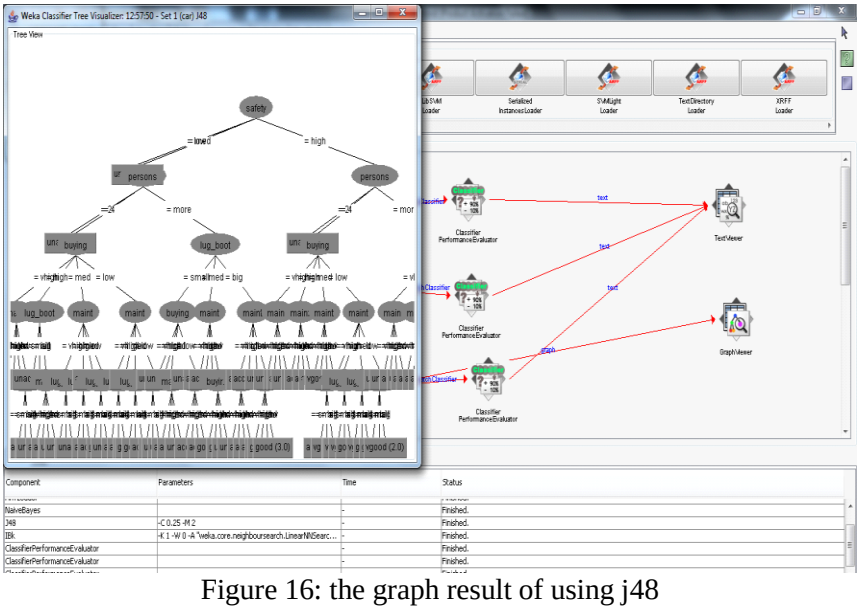

Figure 16: the graph result of using j48

http://www.doi.org/10.62341/NAER1907

Conclusion

This study highlights the importance of applying data mining techniques in the automotive industry, where supervised classification enables the construction of accurate models to evaluate and categorize vehicles based on various features such as engine location, price, number of doors, stroke, and city fuel consumption. Comparisons between the J48 and MONK algorithms using the WEKA tool demonstrated that selecting the appropriate algorithm enhances classification accuracy and reduces errors, thereby improving the reliability of the system. The research confirms that integrating machine learning in the automotive sector can support vehicle quality prediction, optimize operational performance, and enable more effective decision-making in production and development processes. Future studies could focus on expanding the dataset to include a wider range of vehicle types and additional performance and safety features. Incorporating advanced machine learning techniques, such as ensemble methods, deep learning, or hybrid models, may further improve classification accuracy and predictive capabilities. Additionally, integrating real-time sensor data from vehicles could allow for dynamic performance monitoring and adaptive decision-making, enhancing both efficiency and safety in automotive manufacturing and operations.

References

[1] Haritaoglu, I. (2001, September). InfoScope: Link from real world to digital information space. In *International Conference on Ubiquitous Computing* (pp. 247-255). Berlin, Heidelberg: Springer Berlin Heidelberg.

[2] Khalifa, Z., & Rahal, I. (2024). Integration of Blockchain Technology in the Sustainable Supply Chain Management. *International Science and Technology Journal*, *34*(1), 1-23.https://doi.org/10.62341/zkir2928

[3] Nizar, A. H., Dong, Z. Y., Zhao, J. H., & Zhang, P. (2007, June). A data mining based NTL analysis method. In *2007 IEEE power engineering society general meeting* (pp. 1-8). IEEE.

[4] Kumar, U., & Sharma, N. (2025, February). Analysis of data mining tools and techniques for weather forecasting. In *AIP Conference Proceedings* (Vol. 3162, No. 1, p. 020071). AIP Publishing LLC.

http://www.doi.org/10.62341/NAER1907

[5] Ratra, R., Gulia, P., & Gill, N. S. (2021, July). Performance Analysis of Classification Techniques in Data Mining using WEKA. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.

[6] Olanow, C. W., Watts, R. L., & Koller, W. C. (2001). An algorithm (decision tree) for the management of Parkinson's disease (2001): treatment guidelines. *Neurology*, *56*(suppl_5), S1-S88.

[7] Otero, F. E., Freitas, A. A., & Johnson, C. G. (2012). Inducing decision trees with an ant colony optimization algorithm. *Applied Soft Computing*, *12*(11), 3615-3626.

[8] Kotelnikov, E. V., & Milov, V. R. (2018, May). Comparison of rule induction, decision trees and formal concept analysis approaches for classification. In *Journal of Physics: Conference Series* (Vol. 1015, No. 3, p. 032068). IOP Publishing.

[9] Sarang, P. (2023). Naive Bayes: a supervised learning algorithm for classification. In *Thinking data science: A data science practitioner's guide* (pp. 143-152). Cham: Springer International Publishing.

[10] Patil, A. S., & Pawar, B. V. (2012, March). Automated classification of web sites using Naive Bayesian algorithm. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 14-16).

[11] Reddy, E. M. K., Gurrala, A., Hasitha, V. B., & Kumar, K. V. R. (2022). Introduction to Naive Bayes and a review on its subtypes with applications. *Bayesian reasoning and gaussian processes for machine learning applications*, 1-14.

[12] Bai, Y., & Bain, M. (2022). Optimizing weighted lazy learning and Naive Bayes classification using differential evolution algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *13*(6), 3005-3024.

[13] Zheng, Z., & Webb, G. I. (2000). Lazy learning of Bayesian rules. *Machine learning*, *41*(1), 53-84.

[14] Rahal, I., & Elloumi, A. (2024). A Multi-Objective Model for Perishable Products Supply Chain Optimization. *Iranian Economic Review*.

[15] Imen, R., & Abdelkarim, E. (2024). Supply Chain Management for Perishable Products: A Literature Review. *IUP Journal of Supply Chain Management*, *21*(1).