Weka Workbench for Analysis of Machine Learning Classification Algorithms

Sheeba P
CSE Department, NHCE
Email: sheebaravindra87@gmail.com

Abstract- Weka is a workbench that consists of machine learning algorithms that can be easily applied to any dataset. The dataset can be pre-processed, provide as input to a learning model and the resulting classification and its performance can be analysed without even writing a single line of source code. Clustering, association rule mining, regression, classification, and attribute selection are the main functionalities included in the workbench. The input to all the algorithms in the workbench is from a single relational table or through a query from a database. In this paper, we are discussing how Weka workbench can be used for analysing various classification algorithms.

Index Terms- Weka, classification, machine learning, Bayesian, trees.

1. INTRODUCTION

Weka has a graphical user interface called the explorer. All the functionalities of weka are provided using menu selection and form filling. The dataset can be imported from a file in any of the following formats ARFF, CSV, C4.5, binary, or can be read from a URL or from an SQL database and using the tool a decision tree can be built. The incremental algorithms in Weka can be used to process very large datasets. One main disadvantage of weka explorer is that it copies everything to the main memory. The other graphical user interface called the Knowledge Flow allows design configurations for data processing of streamed dataset [1].The Workbench, is a graphical user interface that combines the explorer, Knowledge Flow and the Experimenter into one single application. The various tabs are:
1. Pre-process: Choosing the dataset and modifying it in different ways. 2. Classification: Training the learning models that perform classification or regression and accordingly evaluate them. 3. Cluster: Learning the clusters from the dataset. 4. Associate: Learn association rules for the data and evaluate them. 5. Select the attributes: Select the most relevant aspects of the dataset. 6. Visualize: Viewing different two-dimensional plots of the data and interact with them. In this paper, we are discussing about the following classifiers: Bayesian classifiers, Trees, Rules, Neural Networks and Lazy classifiers. The dataset can be loaded using the open file option in the pre-process tab. The most commonly loaded data file is the ARFF file. Once the data is loaded, weka lists the attributes in the attribute window.

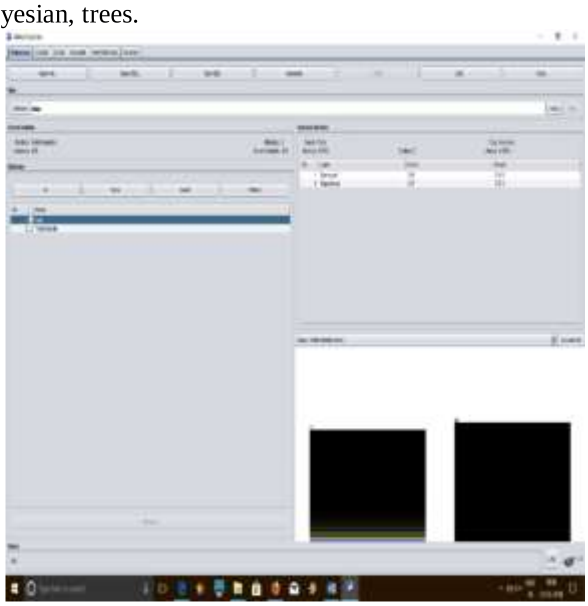

Fig. 1 Weka Graphical User Interface

During the scanning of the data some basic statistics for each attribute is calculated by Weka [2]. In the 'Selected attribute' box on the right panel of 'Pre-process' the following statistics are displayed:
Name: attribute name
Type: the most commonly used Nominal or Numeric
Missing: the total number of instances in the dataset for which the attribute value is not specified,
Distinct: the number of different values that the data contains for this attribute, and
Unique: the total number of instances in the data having a value for this attribute that none of the other instances have.

2. BAYESIAN CLASSIFIER

The probabilistic Naive Bayes classifier is implemented using the Bayesian Classifier. The normal distribution is used to model the numeric attributes in Naive Bayes classifier. On the Classify panel, using the Choose button, when you select a learning algorithm the command-line version of the classifier is seen in the line beside the button, including the parameters specified with minus signs.[3][4] To change the parameters click the corresponding line to get an appropriate object editor. BayesNet option learns Bayesian networks. Four different algorithms are used for calculating the conditional probability tables of the network. The K2 or the TAN algorithms are used to do the searches. The network structure can be observed by right clicking the history item and also selecting the option Visualize graph. Select the 'Choose' button in the 'Classifier' box and select the NaïveBayes classifier.

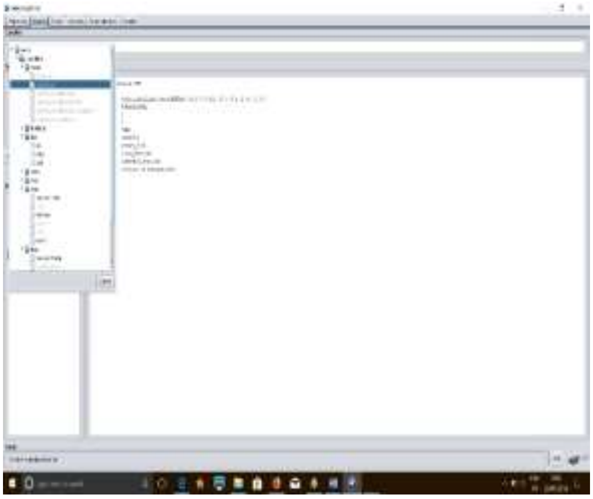

Fig 2. Choosing NaïveBayes classifier

The test options can be set using 'Test options' box. The test options are:
1. The training set: Evaluates the classifier on how well it predicts the class of the instances it was trained on.
2. The test set: Evaluates the classifier on how well it predicts the class of a set of instances loaded from a file.
Clicking on the 'Set...' button brings up a dialog allowing you to choose the file to test on.
3. Cross-validation: Evaluates the classifier by cross-validation, using the number of folds that are entered in the 'Folds' text field.
4. Percentage split: Evaluates the classifier on how well it predicts a certain percentage of the data, which is held out for testing. The amount of data held out depends on the value entered in the '%' field.[5][6]

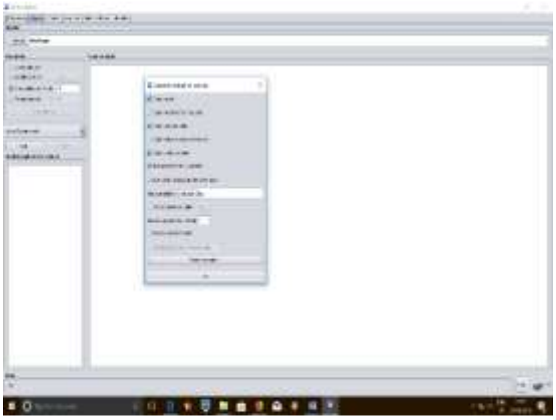

Fig 3. Classifier evaluation options

Using the option 'Classifier evaluation options' what needs to be seen in the output can be selected [7]. Some options are:

1. Output model: The classification model on the full training set can be viewed and visualized.
2. Output per-class stats: The precision/recall and true/false statistics for each class output.
3. Output confusion matrix: The confusion matrix of the predictions of the classifier is included in the output. The classification algorithm can be run once the options have been specified. Selecting the 'Start' button to starts the learning process. The learning process can be stopped at any time by clicking on 'Stop' button [8].

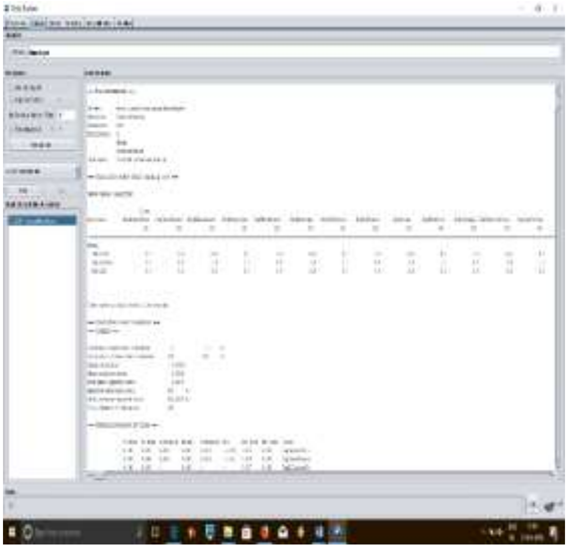

Fig 4. Output model

3. TREES

Using Weka Tree Classifiers, a binary tree can be built instead of the multiway branches [10]. The pruning confidence threshold value, whose default value is 0.25, and the minimum number of instances permissible at a leaf whose default is 2 can also be set. Either C4.5 pruning or reduced-error pruning can be chosen. The numFolds parameter whose default value is 3, determines the size of the pruning set. RandomTree builds trees considering a given number of random features at each node, without any pruning. RandomForest builds random forests by aggregating the ensembles of random trees. Using information gain/variance reduction, REPTree builds a decision or a regression tree and pruning is done using reduced error pruning. In this method, speed is optimized, values are sorted once for numeric attributes. It also considers attributes with missing values by splitting the instances into chunks. LMT is used to build logistic model trees. It can be used for the following attributes, binary and multiclass target variables, numeric and nominal attributes, and missing values. LogitBoost algorithm can be used for fitting the logistic regression functions at the level of a node. Cross-validation is used to calculate the no of iterations. This method improves the run time considerably. Cross validation generally minimizes the misclassification error. The minimal cost-complexity is reduced by LMT pruning mechanism to produce generate a compact tree structure [11][12].

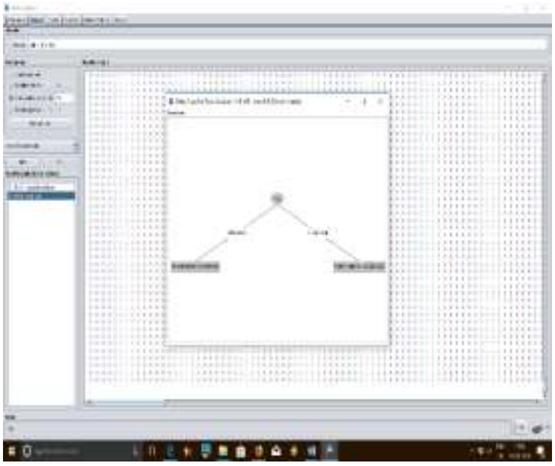

Fig 5. Visualization of Trees in Weka

4. RULES

Decision Table is used to build a decision table classifier. The feature subsets are evaluated using best-first search and cross-validation. The nearest-neighbor method can be used to determine the class for each instance that is not covered by a decision table entry, instead of the table's global majority, based on the same set of features.[12] OneR is the 1R classifier with a single parameter is the minimum bucket size for discretization. The Classifier model part shows that wage-increase-first-year has been identified as the basis of the rule produced, with a split at the value 2.9 dividing bad outcomes from good [13]. Beneath the rules the fraction of training instances correctly classified by the rules is given in parentheses. Three rules are found, and are intended to be processed in order, the prediction generated for any test instance being the outcome of the first rule that fires. Rules for partial decision trees are acquired from the Part option. M5Rules retrieves regression rules from model trees built using M5.

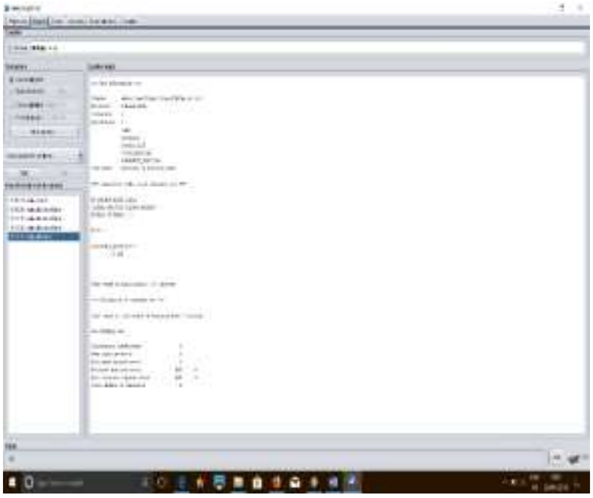

Fig 6. Rules in Weka

5. NEURAL NETWORKS

Backpropagation is used to train a Multilayer Perceptron which is a type of perceptron. In the object editor, invoke MultilayerPerceptron, set GUI to True and execute the network by clicking Start on the Classify panel. The learning rate can also be set. A decay rate can be used to reduce the learning rate. The object editor can be used to set several other parameters for a multilayer perceptron. Hidden layers can be added to the multilayer perceptron using autobuild option. The number of training epochs can be set by the trainingTime parameter [14]. The NominalToBinaryFilter parameter is set by default in the MultilayerPerceptron object editor. Turning the parameter off may improve performance on data in which the nominal attributes are really ordinal. The attributes can be normalized using the normalizeAttributes, and a numeric class can be normalized using normalizeNumericClass [15].

6. CONCLUSION

In this paper, an introduction to the usage of Weka tool for classification algorithms are covered. The various options in classification algorithms like Naïve Bayes, Trees, rules and Artificial Neural network are explained. The Weka tool is a comprehensive workbench for analysing all the machine learning algorithms. Apart from textual representations of the output, we can also use the visualize option to provide graphical representations especially in the case of trees. This paper gives an introduction to weka tool, preprocessing the data, selecting the attributes from the dataset, classification using various algorithms, analysing and visualising the results.

REFERENCES

[1] Ramaswami M, "Validating Predictive Performance of Classifier Models for Multiclass Problem in Educational Data Mining", IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 5, No 2, September 2014.

[2] M. Ramaswami and R. Bhaskaran, "A CHAID based performance prediction model in educational data mining," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1,No.1, pp.10-18,January2010.

[3] International journal of Innovative Technology and Creative Engineering (IJITCE), 1(12), S. K. Yadav, B. K. Bharadwaj & Pal, S. 2011, Data Mining Applications: A comparative study for predicting students' performance.

[4] P. Thangaraju. R. Mehala, Performance Analysis of PSO-KStar Classifier over Liver Diseases, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 7, July 2015

[5] Comparison the various clustering algorithms of weka tools Narendra Sharma, Aman Bajpa , Mr. Ratnesh Litoriya 3 1,2,3 Department of computer science, Jaypee University ofEngg. & Technology

[6] Olivier C.H. Fran,cois and Philippe Leray Learning the Tree Augmented Naive Bayes Classifier from incomplete datasets LITIS Lab., INSA de Rouen, BP 08, avo de I'Unive 76801 Saint-Etienne-Du-Rouvray, France.

[7] Jiangtao Ren* ,Sau Dan Leet , Xianlu Chen* , Ben Kaot , Reynold Cheng'i' and David Cheung'i' * Naive Bayes Classification of Uncertain Data Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China.

[8] Leo Breiman, Jerome Friedman, Richard Olshen, Charles Stone (1984) "Classification and Regression Trees" (Wadsworth).

[9] A. Kusiak, K.H. Kernstine, J.A. Kern, K.A. McLaughlin and T.L. Tseng, Data Mining: Medical and Engineering Case Studies, Proceedings of the Industrial Engineering Research 2000 Conference(2000), Cleveland, Ohio, May 21-23, pp. 1-7

[10] K. Hornik, C. Buchta and A. Zeileis, Open-Source Machine Learning: R Meets Weka, Computational Statistics (2009), p 225-232

[11]R. Bouckaert, E. Frank, M.Hall, R.Kirkby, P. Reutemann, A. Seewald and D. Scuse. WEKA Manual for Version 3-6-0, University of Waikato, Hamilton, New Zealand, 2008

[12] I.H. Witten and E. Frank, Data Mining Practical Machine Learning Tools and Techniques, Second Edition, Elsevier Inc.,2005

[13] R. Frank, M. Ester and A. Knobbe, A Multi-Relational Approach to Spatial Classification, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (2009), Paris, France, pp. 309-318

[14] S.Eyheramendy, D. Lewis, D.Madigan, On the Naïve Bayes Model for Text Categorization. Artificial Intelligence & Statistics, 2003.

[15] V Vapnik, Statistical Learning Theory. New York:Wiley-Interscience Publication. John Wiley and Sons, Inc, 1998.