CSPC 54 : Assignment 10

WEKA tool Abstract

Name : Rajneesh Pandey,

Roll no. 106119100,

Class : CSE-B

---

1. Features of 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 our own Java code. It contains its own Dataset also.

Weka features include machine learning, data mining, pre-processing, classification, regression, clustering, association rules, attribute selection, experiments, workflow and visualization. Weka is written in Java, developed at the University of Waikato, New Zealand.

All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining.

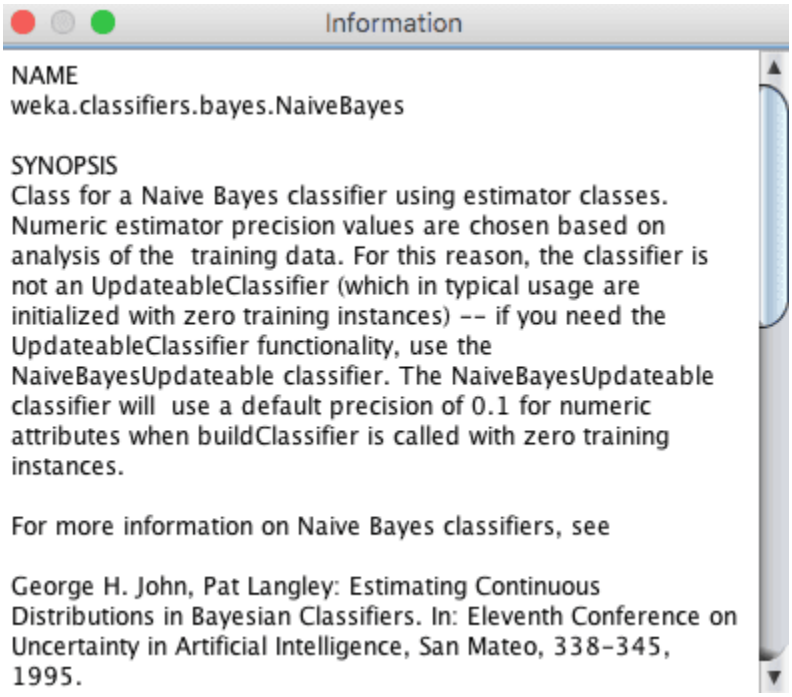
Some more features,

- **Open Source**: It is released as open-source software under the GNU GPL. It is dual licensed, and Pentaho Corporation owns the exclusive license to use the platform for business intelligence in their own product.

- **Graphical Interface**: It has a Graphical User Interface (GUI). This allows to complete machine learning projects without programming.

- **Command Line Interface**: All features of the software can used from the command line. This can be very useful for scripting large jobs.

- **Java API**: It is written in Java and provides a API that is well documented and promotes integration into our own applications. Note that the GNU GPL means that in turn software would also have to be released as GPL.
- **Documentation**: There books, manuals, wikis, and MOOC courses that can train how to use the platform effectively.
- **More Info** : we can get more information about the algorithm in WEKA



2. List of machine learning algorithms supported by WEKA :

- **Bayes**: Algorithms that use Bayes Theorem in some core way, like Naive Bayes.
- **Function**: Algorithms that estimate a function, like Linear Regression.
- **Lazy**: Algorithms that use lazy learning, like k-Nearest Neighbors.
- **Meta**: Algorithms that use or combine multiple algorithms, like Ensembles.
- **Misc**: Implementations that do not neatly fit into the other groups, like running a saved model.
- **Rules**: Algorithms that use rules, like One Rule.
- **Rees**: Algorithms that use decision trees, like Random Forest.

More details about algorithms

Linear Machine Learning Algorithms

Linear algorithms assume that the predicted attribute is a linear combination of the input attributes.

- Linear Regression: function.LinearRegression
- Logistic Regression: function.Logistic

Nonlinear Machine Learning Algorithms

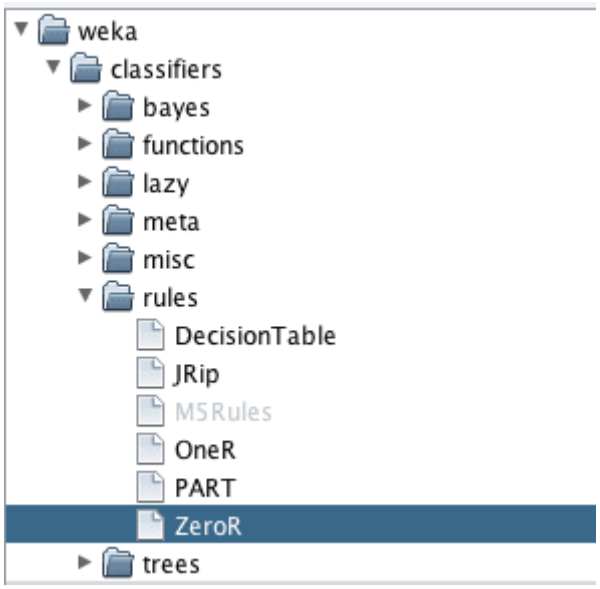
Nonlinear algorithms do not make strong assumptions about the relationship between the input attributes and the output attribute being predicted.

- Naive Bayes: bayes.NaiveBayes
- Decision Tree (specifically the C4.5 variety): trees.J48
- k-Nearest Neighbors (also called KNN: lazy.IBk
- Support Vector Machines (also called SVM): functions.SMO
- Neural Network: functions.MultilayerPerceptron

Ensemble Machine Learning Algorithms

Ensemble methods combine the predictions from multiple models in order to make more robust predictions.

- Random Forest: trees.RandomForest
- Bootstrap Aggregation (also called Bagging): meta.Bagging
- Stacked Generalization (also called Stacking or Blending): meta.Stacking

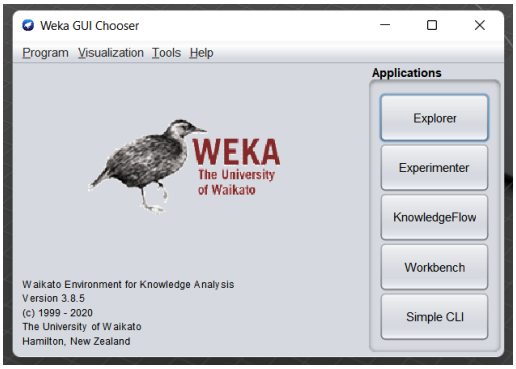


3. Run a sample algorithm and mark your observations

Association Rule Learning :

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases.

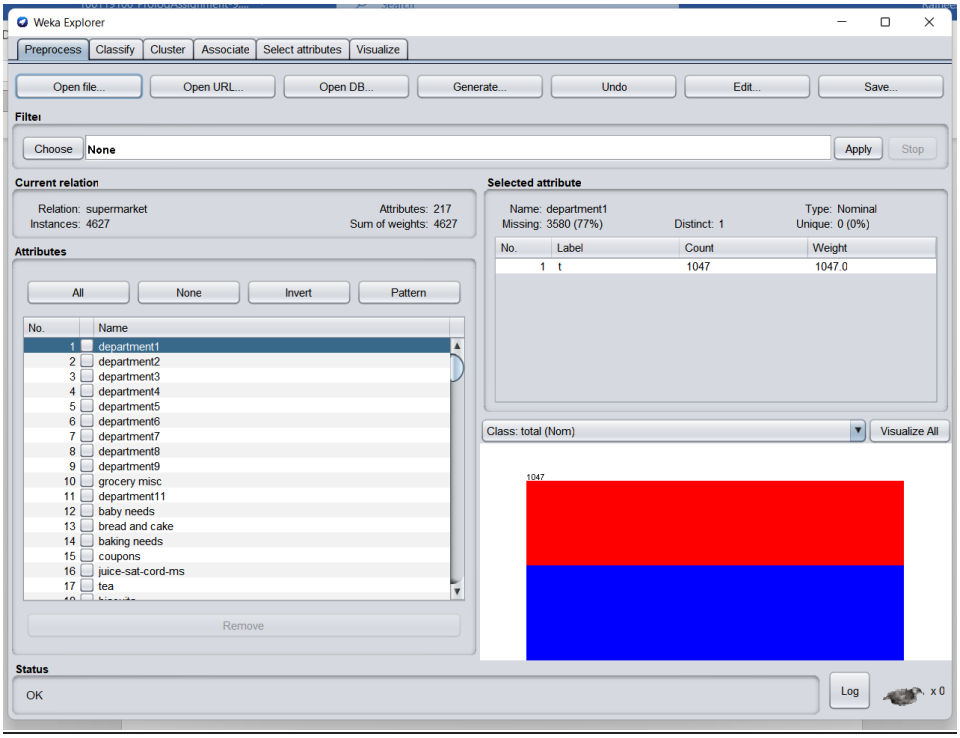
1. Start the Weka Explorer



2. Load the Supermarket Datasets

Load the Supermarket dataset (*data/supermarket.arff*). This is a dataset of point-of-sale information. The data is nominal, and each instance represents a customer transaction at a supermarket, the products purchased and the departments involved.

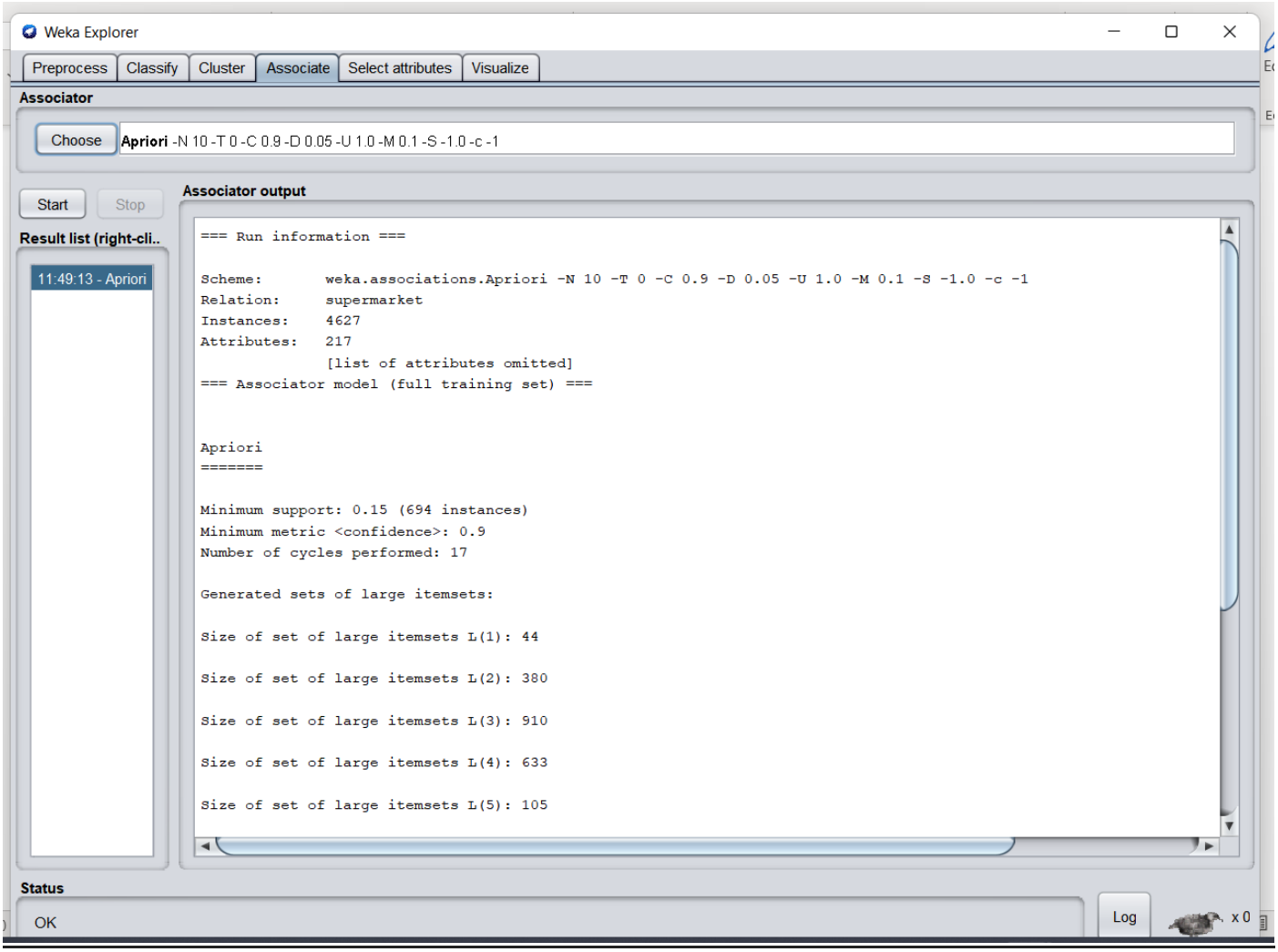
The data contains 4,627 instances and 217 attributes. The data is denormalized. Each attribute is binary and either has a value ("*t*" for true) or no value ("*?*" for missing). There is a nominal class attribute called "total" that indicates whether the transaction was less than $100 (low) or greater than $100 (high).



3. Discover Association Rules and Analyze Results

Clicking on the “Associate” tab in the Weka Explorer. The “**Apriori” algorithm** will already be selected. This is the most well known association rule learning method

It builds up attribute-value (item) sets that maximize the number of instances that can be explained (coverage of the dataset). The search through item space is very much like the problem faced with attribute selection and subset search.



=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: supermarket

Instances: 4627

Attributes: 217

[list of attributes omitted]

=== Associator model (full training set) ===

Apriori

=======

Minimum support: 0.15 (694 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723 <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)

2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696 <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)

3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705 <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)

4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746 <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)

5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779 <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)

6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725 <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)

7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701 <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)

8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)

9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757 <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)

10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)

Observations

As the real work for association rule learning is in the interpretation of results.

From looking at the "*Associator output*" window, we can see that the algorithm presented 10 rules learned from the supermarket dataset

We can see rules are presented in antecedent => consequent format. The number associated with the antecedent is the absolute coverage in the dataset (in this case a number out of a possible total of 4,627). The number next to the consequent is the absolute number of instances that match the antecedent and the consequent. The number in brackets on the end is the support for the rule (number of antecedents divided by the number of matching consequents).we can see that a cut-off of 91% was used in selecting rules, mentioned in the "Associator output" window and indicated in that no rule has a coverage less than 0.91.

Few observations:

- We can see that all presented rules have a consequent of "bread and cake".
- All presented rules indicate a high total transaction amount.
- "biscuits" an "frozen foods" appear in many of the presented rules.