

info@stratebi.com

91.788.34.10

www.stratebi.com

www.todobi.com (Agosto 2010)



Comparativa de Algoritmos de Herramientas de Data mining

Contenido

| | |
|--------------------------------------|----|
| Reglas de Asociación..... | 3 |
| Clasificadores Bayesianos | 4 |
| Funciones de Clasificación | 5 |
| Algoritmos Perezosos (Lazy) | 7 |
| Meta Clasificadores..... | 8 |
| Clasificadores Multi Instancia | 12 |
| Reglas de clasificación..... | 14 |
| Árboles | 15 |
| Evaluación de subconjuntos | 17 |
| Evaluación de atributos..... | 18 |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|----------------------|---------------------------------|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Reglas de Asociación | Apriori | Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence. The algorithm has an option to mine class association rules. | Si | Si | Si | Si * | No | Si * |
| | Bitvector Generator | Generates bitvectors either from a table containing numerical values, or from a string column containing the bit positions to set, hexadecimal or binary strings. | No | No | No | Si | No | No |
| | eclat | Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal. | No | Si | No | No | No | No |
| | Filtered Associator | Class for running an arbitrary associator on data that has been passed through an arbitrary filter. Like the associator, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure. | Si | No | No | Si * | No | No |
| | Generalized Sequential Patterns | Class implementing a GSP algorithm for discovering sequential patterns in a sequential data set. The attribute identifying the distinct data sequences contained in the set can be determined by the respective option. Furthermore, the set of output results can be restricted by specifying one or more attributes that have to be contained in each element/itemset of a sequence. | Si | No | No | Si * | No | Si * |
| | HotSpot | HotSpot learns a set of rules (displayed in a tree-like structure) that maximize/minimize a target variable/value of interest. With a nominal target, one might want to look for segments of the data where there is a high probability of a minority value occurring (given the constraint of a minimum support). For a numeric target, one might be interested in finding segments where this is higher on average than in the whole data set. For example, in a health insurance scenario, find which health insurance groups are at the highest risk (have the highest claim ratio), or, which groups have the highest average insurance payout. | Si | No | No | No | No | Si * |
| | Predictive Apriori | Class implementing the predictive apriori algorithm to mine association rules. It searches with an increasing support threshold for the best 'n' rules concerning a support-based corrected confidence value. | Si | No | No | Si * | No | Si * |
| | Tertius | Finds rules according to confirmation measure (Tertius-type algorithm). | Si | No | No | Si * | No | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|---------------------------|---|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Clasificadores Bayesianos | AODE | AODE achieves highly accurate classification by averaging over all of a small space of alternative naive-Bayes-like models that have weaker (and hence less detrimental) independence assumptions than naive Bayes. The resulting algorithm is computationally efficient while delivering highly accurate classification on many learning tasks. | Si | No | No | Si * | No | Si * |
| | AODEsr | AODEsr augments AODE with Subsumption Resolution. AODEsr detects specializations between two attribute values at classification time and deletes the generalization attribute value. | Si | No | No | No | No | Si * |
| | Bayesian Logistic Regression | Implements Bayesian Logistic Regression for both Gaussian and Laplace Priors. | Si | Si | No | No | No | Si * |
| | BayesNet | Bayes Network learning using various search algorithms and quality measures. | Si | No | No | Si * | No | Si * |
| | Complement Naive Bayes | Class for building and using a Complement class Naive Bayes classifier. | Si | No | No | Si * | No | Si * |
| | DMNBtext | Class for building and using a Discriminative Multinomial Naive Bayes classifier. For more information see | Si | No | No | No | No | No |
| | Editable Bayes Net | Bayes Network learning using various search algorithms and quality measures. | Si | No | No | No | No | Si * |
| | HNB | Constructs Hidden Naive Bayes classification model with high classification accuracy and AUC. | Si | No | No | Si * | No | Si * |
| | Naive 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 UpdateableClassifier (which in typical usage are initialized with zero training instances) | Si | No | No | Si * | No | Si |
| | Naive Bayes Multinomial | Class for building and using a multinomial Naive Bayes classifier | Si | No | No | Si * | No | Si * |
| | Naive Bayes Multinomial Updateable | Class for building and using a multinomial Naive Bayes classifier | Si | No | No | No | No | Si * |
| | Naive Bayes Simple | Class for building and using a simple Naive Bayes classifier. Numeric attributes are modelled by a normal distribution. | Si | No | Si | Si * | No | No |
| Naive Bayes Updateable | Class for a Naive Bayes classifier using estimator classes. This is the updateable version of NaiveBayes. This classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances. | Si | No | No | Si * | No | No | |
| WAODE | WAODE constructs the model called Weightily Averaged One-Dependence Estimators. | Si | No | No | Si * | No | Si * | |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|----------------------------|---|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Funciones de Clasificación | Gaussian Processes | Implements Gaussian Processes for regression without hyperparameter-tuning | Si | Si | No | Si * | No | Si |
| | Isotonic Regression | Learns an isotonic regression model. Picks the attribute that results in the lowest squared error. Missing values are not allowed. Can only deal with numeric attributes.Considers the monotonically increasing case as well as the monotonicallydecreasing case | Si | Si | No | Si * | No | Si * |
| | Kernel Logistic Regression | A kernel logistic regression learner for binary classification tasks | No | No | No | No | No | Si |
| | Least MedSq | Implements a least median squared linear regression utilising the existing weka LinearRegression class to form predictions. Least squared regression functions are generated from random subsamples of the data. The least squared regression with the lowest median squared error is chosen as the final model. | Si | Si | No | Si * | No | Si * |
| | Linear Regression | Class for using linear regression for prediction. Uses the Akaike criterion for model selection, and is able to deal with weighted instances. | Si | Si | Si | Si * | Si | Si |
| | Logistic | Class for building and using a multinomial logistic regression model with a ridge estimator. | Si | Si | Si | Si * | Si | Si |
| | Multilayer Perceptron | A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units). | Si | Si | Si | Si * | No | Si |
| | Pace Regression | Class for building pace regression linear models and using them for prediction. Under regularity conditions, pace regression is provably optimal when the number of coefficients tends to infinity. It consists of a group of estimators that are either overall optimal or optimal under certain conditions. | Si | No | No | Si * | No | Si * |
| | PNN | Trains a Probabilistic Neural Network (PNN) on labeled data. | No | No | No | Si | No | Si |
| | RBF Network | Class that implements a normalized Gaussian radial basis function network. It uses the k-means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. If the class is nominal it uses the given number of clusters per class.It standardizes all numeric attributes to zero mean and unit variance. | Si | No | Si | Si * | No | Si * |
| | RProp MLP Learner | Builds and learns an Multi Layer Perceptron with resilient backpropagation | No | No | No | Si | No | No |
| Simple Linear Regression | Learns a simple linear regression model. Picks the attribute that results in the lowest squared error. Missing values are not allowed. Can only deal with numeric attributes. | Si | No | No | Si * | No | Si * | |

| | | | | | | | |
|------------------------|--|----|----|----|------|----|------|
| Simple Logistic | Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross-validated, which leads to automatic attribute selection | Si | No | No | Si * | No | Si * |
| SGD | Implements stochastic gradient descent for learning various linear models (binary class SVM, binary class logistic regression and linear regression). globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes, so the coefficients in the output are based on the normalized data. This implementation can be trained incrementally on (potentially) infinite data streams | Si | No | No | No | No | No |
| SMO | Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data – this is important for interpreting the classifier.) | Si | No | No | Si * | No | Si * |
| SMOreg | Implements Alex Smola and Bernhard Scholkopf's sequential minimal optimization algorithm for training a support vector regression model. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (Note that the coefficients in the output are based on the normalized/standardized data, not the original data.) | Si | No | No | Si * | No | Si * |
| SPegasos | Implements the stochastic variant of the Pegasos (Primal Estimated sub-GrAdient SOLver for SVM) method of Shalev-Shwartz et al. (2007). This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes, so the coefficients in the output are based on the normalized data. Can either minimize the hinge loss (SVM) or log loss (logistic regression). This implementation can be trained incrementally on (potentially) infinite data streams. | Si | No | No | No | No | No |
| SVMreg | Implements Alex Smola and Bernhard Scholkopf's sequential minimal optimization algorithm for training a support vector regression model. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (Note that the coefficients in the output are based on the normalized/standardized data, not the original data.) | Si | No | No | Si * | No | Si * |
| Voted Perceptron | Implementation of the voted perceptron algorithm by Freund and Schapire. Globally replaces all missing values, and transforms nominal attributes into binary ones. | Si | No | No | Si * | No | Si * |
| Weighted least squares | If the spread of residuals is not constant, the estimated standard errors will not be valid. Use Weighted Least Square to estimate the model instead (for example, when predicting stock values, stocks with higher shares values fluctuate more than low value shares.) | No | No | No | No | Si | No |
| Winnow | Implements Winnow and Balanced Winnow algorithms by Littlestone. | Si | No | No | Si * | No | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-----------------------------|----------------------|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Algoritmos Perezosos (Lazy) | Attribute Based Vote | AttributeBasedVotingLearner is very lazy. Actually it does not learn at all but creates an AttributeBasedVotingModel. This model simply calculates the average of the attributes as prediction (for regression) or the mode of all attribute values (for classification). | No | No | No | No | No | Si |
| | IB1 | Nearest-neighbour classifier. Uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. | Si | No | No | Si * | No | Si |
| | IBk | K-nearest neighbours classifier. Can select appropriate value of K based on cross-validation. Can also do distance weighting. | Si | Si | No | Si * | No | Si * |
| | KStar | K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. | Si | No | No | Si * | No | Si * |
| | LBR | Lazy Bayesian Rules Classifier. The naive Bayesian classifier provides a simple and effective approach to classifier learning, but its attribute independence assumption is often violated in the real world. Lazy Bayesian Rules selectively relaxes the independence assumption, achieving lower error rates over a range of learning tasks. LBR defers processing to classification time, making it a highly efficient and accurate classification algorithm when small numbers of objects are to be classified. | Si | No | No | Si * | No | Si * |
| | LWL | Locally weighted learning. Uses an instance-based algorithm to assign instance weights which are then used by a specified WeightedInstancesHandler. Can do classification (e.g. using naive Bayes) or regression (e.g. using linear regression). | Si | No | No | Si * | No | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|---------------------------|---|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Meta Clasificadores | AdaBoost M1 | Class for boosting a nominal class classifier using the Adaboost M1 method. Only nominal class problems can be tackled. Often dramatically improves performance, but sometimes overfits. | Si | Si | No | Si * | No | Si * |
| | Additive Regression | Meta classifier that enhances the performance of a regression base classifier. Each iteration fits a model to the residuals left by the classifier on the previous iteration. Prediction is accomplished by adding the predictions of each classifier. Reducing the shrinkage (learning rate) parameter helps prevent overfitting and has a smoothing effect but increases the learning time. | Si | Si | No | Si * | No | Si * |
| | Arcing [Arc-x4] | Run several learning process, with reweighted examples. | No | No | Si | No | No | No |
| | Attribute Selected Classifier | Dimensionality of training and test data is reduced by attribute selection before being passed on to a classifier | Si | No | No | Si * | No | No |
| | Bagging | Class for bagging a classifier to reduce variance. Can do classification and regression depending on the base learner. | Si | Si | Si | Si * | No | Si * |
| | Bayesian Boosting | Boosting operator based on Bayes' theorem. | No | No | No | No | No | Si |
| | Best Rule Induction | This operator returns the best rule regarding WRAcc using exhaustive search. Features like the incorporation of other metrics and the search for more than a single rule are prepared. The search strategy is BFS, with save pruning whenever applicable. This operator can easily be extended to support other search strategies. | No | No | No | No | No | Si |
| | Binary 2 Multi Class Learner | A metaclassifier for handling multi-class datasets with 2-class classifiers. This class supports several strategies for multiclass classification including procedures which are | No | No | No | No | No | Si |
| | Classification Via Clustering | A simple meta-classifier that uses a clusterer for classification. For cluster algorithms that use a fixed number of clusterers, like SimpleKMeans, the user has to make sure that the number of clusters to generate are the same as the number of class labels in the dataset in order to obtain a useful model. | Si | No | No | No | No | Si * |
| | Classification Via Regression | Class for doing classification using regression methods. Class is binarized and one regression model is built for each class value. | Si | No | No | Si * | No | No |
| Cost Sensitive Classifier | A metaclassifier that makes its base classifier cost-sensitive. Two methods can be used to introduce cost-sensitivity: reweighting training instances according to the total cost assigned to each class; or predicting the class with minimum expected misclassification cost (rather than the most likely class). Performance can often be improved by using a Bagged classifier to improve the probability estimates of the base classifier. | Si | No | No | Si * | No | No | |

| | | | | | | | |
|------------------------|---|----|----|----|------|----|------|
| CV Parameter Selection | Class for performing parameter selection by cross-validation for any classifier. | Si | No | No | Si * | No | No |
| Dagging | This meta classifier creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier. Predictions are made via majority vote, since all the generated base classifiers are put into the Vote meta classifier. Useful for base classifiers that are quadratic or worse in time behavior, regarding number of instances in the training data. | Si | No | No | Si * | No | Si * |
| Decorate | DECORATE is a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training examples. Comprehensive experiments have demonstrated that this technique is consistently more accurate than the base classifier, Bagging and Random Forests. Decorate also obtains higher accuracy than Boosting on small training sets, and achieves comparable performance on larger training sets. | Si | No | No | Si * | No | Si * |
| END | A meta classifier for handling multi-class datasets with 2-class classifiers by building an ensemble of nested dichotomies. | Si | No | No | Si * | No | Si * |
| Ensemble Selection | Combines several classifiers using the ensemble selection method | Si | No | No | No | No | Si * |
| Filtered Classifier | Class for running an arbitrary classifier on data that has been passed through an arbitrary filter. Like the classifier, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure. | Si | No | No | Si * | No | No |
| Grading | Implements Grading. The base classifiers are "graded". | Si | No | No | Si * | No | Si * |
| Grid Search | Performs a grid search of parameter pairs for the a classifier (Y-axis, default is LinearRegression with the "Ridge" parameter) and the PLSFilter (X-axis, "# of Components") and chooses the best pair found for the actual predicting. | Si | No | No | Si * | No | Si * |
| Logit Boost | Class for performing additive logistic regression. This class performs classification using a regression scheme as the base learner, and can handle multi-class problems. | Si | Si | No | Si * | No | Si * |
| Meta Cost | This classifier should produce similar results to one created by passing the base learner to Bagging, which is in turn passed to a CostSensitiveClassifier operating on minimum expected cost. The difference is that MetaCost produces a single cost-sensitive classifier of the base learner, giving the benefits of fast classification and interpretable output (if the base learner itself is interpretable). This implementation uses all bagging iterations when reclassifying training data (the MetaCost paper reports a marginal improvement when only those iterations containing each training instance are used in reclassifying that instance). | Si | No | No | Si * | No | Si |

| | | | | | | | |
|-------------------------------|---|----|----|----|------|----|------|
| Multi Boost AB | Class for boosting a classifier using the MultiBoosting method. MultiBoosting is an extension to the highly successful AdaBoost technique for forming decision committees. MultiBoosting can be viewed as combining AdaBoost with wagging. It is able to harness both AdaBoost's high bias and variance reduction with wagging's superior variance reduction. Using C4.5 as the base learning algorithm, MultiBoosting is demonstrated to produce decision committees with lower error than either AdaBoost or wagging significantly more often than the reverse over a large representative cross-section of UCI data sets. It offers the further advantage over AdaBoost of suiting parallel execution. | Si | No | No | Si * | No | Si * |
| Multi Class Classifier | A metaclassifier for handling multi-class datasets with 2-class classifiers. This classifier is also capable of applying error correcting output codes for increased accuracy | Si | No | No | Si * | No | Si * |
| Multi Scheme | Class for selecting a classifier from among several using cross validation on the training data or the performance on the training data. Performance is measured based on percent correct (classification) or mean-squared error (regression). | Si | No | No | Si * | No | Si * |
| One Class Classifier | Performs one-class classification on a dataset. Classifier reduces the class being classified to just a single class, and learns the data without using any information from other classes. The testing stage will classify as 'target' or 'outlier' - so in order to calculate the outlier pass rate the dataset must contain information from more than one class. | Si | No | No | No | No | No |
| Ordinal Class Classifier | Meta classifier that allows standard classification algorithms to be applied to ordinal class problems. | Si | No | No | Si * | No | Si * |
| Raced Incremental Logit Boost | Classifier for incremental learning of large datasets by way of racing logit-boosted committees. | Si | No | No | Si * | No | Si * |
| Random Committee | Class for building an ensemble of randomizable base classifiers. Each base classifiers is built using a different random number seed (but based one the same data). The final prediction is a straight average of the predictions generated by the individual base classifiers. | Si | No | No | Si * | No | Si * |
| Random SubSpace | This method constructs a decision tree based classifier that maintains highest accuracy on training data and improves on generalization accuracy as it grows in complexity. The classifier consists of multiple trees constructed systematically by pseudorandomly selecting subsets of components of the feature vector, that is, trees constructed in randomly chosen subspaces. | Si | No | No | Si * | No | Si * |
| Real AdaBoost | Class for boosting a 2-class classifier using the Real Adaboost method. | Si | No | No | No | No | No |
| Regression By Discretization | A regression scheme that employs any classifier on a copy of the data that has the class attribute (equal-width) discretized. The predicted value is the expected value of the mean class value for each discretized interval (based on the predicted probabilities for each interval). | Si | No | No | Si * | No | No |
| Rotation Forest | Class for construction a Rotation Forest. Can do classification and regression depending on the base learner. | Si | No | No | No | No | Si * |
| Stacking | Combines several classifiers using the stacking method. Can do classification or regression | Si | No | No | Si * | No | Si * |
| StackingC | Implements StackingC (more efficient version of stacking). | Si | No | No | Si * | No | Si * |

| | | | | | | | |
|---------------------------|--|----|----|----|------|----|------|
| Support Vector Clustering | Clustering with support vectors | No | No | No | No | No | Si |
| Threshold Selector | A metaclassifier that selecting a mid-point threshold on the probability output by a Classifier. The midpoint threshold is set so that a given performance measure is optimized. Currently this is the F-measure. Performance is measured either on the training data, a hold-out set or using cross-validation. In addition, the probabilities returned by the base learner can have their range expanded so that the output probabilities will reside between 0 and 1 (this is useful if the scheme normally produces probabilities in a very narrow range). | Si | No | No | Si * | No | Si |
| Vote | Class for combining classifiers. Different combinations of probability estimates for classification are available. | Si | No | No | Si * | No | Si |
| Class Balanced ND | A meta classifier for handling multi-class datasets with 2-class classifiers by building a random class-balanced tree structure. | Si | No | No | Si * | No | Si * |
| Data Near Balanced ND | A meta classifier for handling multi-class datasets with 2-class classifiers by building a random data-balanced tree structure. | Si | No | No | Si * | No | Si * |
| ND | A meta classifier for handling multi-class datasets with 2-class classifiers by building a random tree structure | Si | No | No | Si * | No | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|--------------------------------|--|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Clasificadores Multi Instancia | Citation KNN | Modified version of the Citation kNN multi instance classifier | Si | No | No | No | No | Si * |
| | MDD | Modified Diverse Density algorithm, with collective assumption. | Si | No | No | No | No | Si * |
| | MI Boost | MI AdaBoost method, considers the geometric mean of posterior of instances inside a bag (arithmetic mean of log-posterior) and the expectation for a bag is taken inside the loss function. | Si | No | No | No | No | Si * |
| | MIDD | Re-implement the Diverse Density algorithm, changes the testing procedure. | Si | No | No | No | No | Si * |
| | MIEMDD | EMDD model builds heavily upon Dietterich's Diverse Density (DD) algorithm. It is a general framework for MI learning of converting the MI problem to a single-instance setting using EM. In this implementation, we use most-likely cause DD model and only use 3 random selected positive bags as initial starting points of EM. | Si | No | No | No | No | Si * |
| | MILR | Uses either standard or collective multi-instance assumption, but within linear regression. For the collective assumption, it offers arithmetic or geometric mean for the posteriors. | Si | No | No | No | No | Si * |
| | MINND | Multiple-Instance Nearest Neighbour with Distribution learner. It uses gradient descent to find the weight for each dimension of each exemplar from the starting point of 1.0. In order to avoid overfitting, it uses mean-square function (i.e. the Euclidean distance) to search for the weights. It then uses the weights to cleanse the training data. After that it searches for the weights again from the starting points of the weights searched before. Finally it uses the most updated weights to cleanse the test exemplar and then finds the nearest neighbour of the test exemplar using partly-weighted Kullback distance. But the variances in the Kullback distance are the ones before cleansing. | Si | No | No | No | No | Si * |
| MI Optimal Ball | This classifier tries to find a suitable ball in the multiple-instance space, with a certain data point in the instance space as a ball center. The possible ball center is a certain instance in a positive bag. The possible radiuses are those which can achieve the highest classification accuracy. The model selects the maximum radius as the radius of the optimal ball. | Si | No | No | No | No | Si * | |

| | | | | | | | |
|------------|--|----|----|----|----|----|------|
| MISMO | <p>Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.</p> <p>This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data – this is important for interpreting the classifier.)</p> <p>Multi-class problems are solved using pairwise classification.</p> <p>To obtain proper probability estimates, use the option that fits logistic regression models to the outputs of the support vector machine. In the multi-class case the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method.</p> | Si | No | No | No | No | Si * |
| MISVM | <p>Implements Stuart Andrews' mi_SVM (Maximum pattern Margin Formulation of MIL). Applying weka.classifiers.functions.SMO to solve multiple instances problem. The algorithm first assign the bag label to each instance in the bag as its initial class label. After that applying SMO to compute SVM solution for all instances in positive bags And then reassign the class label of each instance in the positive bag according to the SVM result Keep on iteration until labels do not change anymore.</p> | Si | No | No | No | No | Si |
| MI Wrapper | A simple Wrapper method for applying standard propositional learners to multi-instance data. | Si | No | No | No | No | Si * |
| Simple MI | Reduces MI data into mono-instance data. | Si | No | No | No | No | Si * |
| TLD | Two-Level Distribution approach, changes the starting value of the searching algorithm, supplement the cut-off modification and check missing values. | Si | No | No | No | No | Si * |
| TLD Simple | A simpler version of TLD, mu random but sigma^2 fixed and estimated via data | Si | No | No | No | No | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-------------------------|--|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Reglas de clasificación | Conjunctive Rule | This class implements a single conjunctive rule learner that can predict for numeric and nominal class labels. | Si | No | No | Si * | No | Si * |
| | Decision Table | Class for building and using a simple decision table majority classifier. | Si | No | No | Si * | No | Si * |
| | DTNB | Class for building and using a decision table/naive bayes hybrid classifier. At each point in the search, the algorithm evaluates the merit of dividing the attributes into two disjoint subsets: one for the decision table, the other for naive Bayes. A forward selection search is used, where at each step, selected attributes are modeled by naive Bayes and the remainder by the decision table, and all attributes are modelled by the decision table initially. At each step, the algorithm also considers dropping an attribute entirely from the model. | Si | No | No | No | No | Si * |
| | FURIA | Fuzzy Unordered Rule Induction Algorithm | Si | No | No | Si | No | No |
| | JRip | This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP. | Si | No | No | Si * | No | Si * |
| | M5 Rules | Generates a decision list for regression problems using separate-and-conquer. In each iteration it builds a model tree using M5 and makes the "best" leaf into a rule. | Si | No | No | Si * | No | Si * |
| | NNge | Nearest-neighbor-like algorithm using non-nested generalized exemplars (which are hyperrectangles that can be viewed as if-then rules). | Si | No | No | Si * | No | Si * |
| | OneR | Class for building and using a 1R classifier; in other words, uses the minimum-error attribute for prediction, discretizing numeric attributes. | Si | No | No | Si * | No | Si |
| | PART | Class for generating a PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule. | Si | No | No | Si * | No | Si * |
| | Prism | Class for building and using a PRISM rule set for classification. Can only deal with nominal attributes. Can't deal with missing values. Doesn't do any pruning. | Si | No | No | Si * | No | Si |
| Ridor | The implementation of a Ripple-Down Rule learner. It generates a default rule first and then the exceptions for the default rule with the least (weighted) error rate. Then it generates the "best" exceptions for each exception and iterates until pure. Thus it performs a tree-like expansion of exceptions. The exceptions are a set of rules that predict classes other than the default. IREP is used to generate the exceptions. | Si | No | No | Si * | No | Si * | |
| ZeroR | Class for building and using a 0-R classifier. Predicts the mean (for a numeric class) or the mode (for a nominal class). | Si | No | No | Si * | No | Si * | |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-------------------|---|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Árboles | AD Tree | Class for generating an alternating decision tree. This version currently only supports two-class problems. The number of boosting iterations needs to be manually tuned to suit the dataset and the desired complexity/accuracy tradeoff. Induction of the trees has been optimized, and heuristic search methods have been introduced to speed learning. | Si | No | No | Si * | No | Si * |
| | BF Tree | Class for building a best-first decision tree classifier. This class uses binary split for both nominal and numeric attributes. For missing values, the method of 'fractional' instances is used. | Si | No | No | Si * | No | Si * |
| | Decision Stump | Class for building and using a decision stump. Usually used in conjunction with a boosting algorithm. Does regression (based on mean-squared error) or classification (based on entropy). Missing is treated as a separate value. | Si | Si | No | Si * | No | Si * |
| | CHAID | Learns a pruned decision tree based on a chi squared attribute relevance test. | No | No | No | No | Si | Si |
| | FT | Classifier for building 'Functional trees', which are classification trees that could have logistic regression functions at the inner nodes and/or leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values. | Si | No | No | No | No | Si * |
| | Id3 | Class for constructing an unpruned decision tree based on the ID3 algorithm. Can only deal with nominal attributes. No missing values allowed. Empty leaves may result in unclassified instances. | Si | No | Si | Si * | Si | Si |
| | ID3 Numerical | This operator learns decision trees without pruning using both nominal and numerical attributes. Decision trees are powerful classification methods which often can also easily be understood. This decision tree learner works similar to Quinlan's ID3. | No | No | No | No | No | Si |
| | J48 | Class for generating a pruned or unpruned C4.5 decision tree. | Si | No | No | Si * | No | Si * |
| | J48 graft | Class for generating a grafted (pruned or unpruned) C4.5 decision tree. | Si | No | No | No | No | Si * |
| | LAD Tree | Class for generating a multi-class alternating decision tree using the LogitBoost strategy. | Si | No | No | No | No | No |
| | LMT | Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values. | Si | No | No | Si * | No | Si * |
| | Multi Criterion Decision Stump | A DecisionStump clone that allows to specify different utility functions. It is quick for nominal attributes, but does not yet apply pruning for continuous attributes. Currently it can only handle boolean class labels. | No | No | No | No | No | Si |
| | M5P | M5Base. Implements base routines for generating M5 Model trees and rules | Si | No | No | Si * | No | Si * |
| NBTree | Class for generating a decision tree with naive Bayes classifiers at the leaves | Si | No | No | Si * | No | Si * | |

| | | | | | | | |
|-----------------|---|----|----|----|------|----|------|
| QUEST | A statistical algorithm that selects variables without bias and builds accurate binary trees quickly and efficiently | No | No | No | No | Si | No |
| Random Forest | Class for constructing a forest of random trees. | Si | Si | No | Si * | No | Si |
| Random Tree | Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning. | Si | No | No | Si * | No | Si |
| REP Tree | Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5). | Si | No | No | Si * | Si | Si * |
| Simple Cart | Class implementing minimal cost-complexity pruning. Note when dealing with missing values, use "fractional instances" method instead of surrogate split method. | Si | Si | No | Si * | No | Si * |
| User Classifier | Interactively classify through visual means. You are Presented with a scatter graph of the data against two user selectable attributes, as well as a view of the decision tree. You can create binary splits by creating polygons around data plotted on the scatter graph, as well as by allowing another classifier to take over at points in the decision tree should you see fit. | Si | No | No | Si * | No | No |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|----------------------------|----------------------------|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Evaluación de subconjuntos | Cfs Subset Eval | Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred | Si | No | No | No | No | No |
| | Classifier Subset Eval | Classifier subset evaluator: Evaluates attribute subsets on training data or a separate hold out testing set. Uses a classifier to estimate the 'merit' of a set of attributes. | Si | No | No | No | No | No |
| | Consistency Subset Eval | Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. Consistency of any subset can never be lower than that of the full set of attributes, hence the usual practice is to use this subset evaluator in conjunction with a Random or Exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes. | Si | No | No | No | No | No |
| | Cost Sensitive Subset Eval | A meta subset evaluator that makes its base subset evaluator cost-sensitive. | Si | No | No | No | No | Si * |
| | Filtered Subset Eval | Class for running an arbitrary subset evaluator on data that has been passed through an arbitrary filter (note: filters that alter the order or number of attributes are not allowed). Like the evaluator, the structure of the filter is based exclusively on the training data. | Si | No | No | No | No | No |
| | Wrapper Subset Eval | Evaluates attribute sets by using a learning scheme. Cross validation is used to estimate the accuracy of the learning scheme for a set of attributes. | Si | No | No | Si * | No | No |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-----------------------------------|--|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Evaluación de atributos | Chi Squared Attribute Eval | Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class. | Si | No | No | No | No | No |
| | Classifier Attribute Eval | Evaluates the worth of an attribute by using a user-specified classifier. | Si | No | No | No | No | No |
| | Cost Sensitive Attribute Eval | A meta subset evaluator that makes its base subset evaluator cost-sensitive. | Si | No | No | No | No | No |
| | Filtered Attribute Eval | Class for running an arbitrary attribute evaluator on data that has been passed through an arbitrary filter (note: filters that alter the order or number of attributes are not allowed). Like the evaluator, the structure of the filter is based exclusively on the training data. | Si | No | No | No | No | No |
| | Gain Ratio Attribute Eval | Evaluates the worth of an attribute by measuring the information gain with respect to the class. $\text{InfoGain}(\text{Class}, \text{Attribute}) = H(\text{Class}) - H(\text{Class} \text{Attribute})$. | Si | No | No | No | No | No |
| | Info Gain Attribute Eval | Evaluates the worth of an attribute by using the OneR classifier. | Si | No | No | No | No | No |
| | OneR Attribute Eval | Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Can operate on both discrete and continuous class data. | Si | No | No | No | No | No |
| | Relief F Attribute Eval | Evaluates the worth of an attribute by using an SVM classifier. Attributes are ranked by the square of the weight assigned by the SVM. Attribute selection for multiclass problems is handled by ranking attributes for each class separately using a one-vs-all method and then "dealing" from the top of each pile to give a final ranking. | Si | No | No | Si * | No | No |
| | SVM Attribute Eval | Evaluates the worth of an attribute by using an SVM classifier. Attributes are ranked by the square of the weight assigned by the SVM. Attribute selection for multiclass problems is handled by ranking attributes for each class separately using a one-vs-all method and then "dealing" from the top of each pile to give a final ranking. | Si | No | No | No | No | No |
| Symmetrical Uncert Attribute Eval | Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class. $\text{SymmU}(\text{Class}, \text{Attribute}) = 2 * (H(\text{Class}) - H(\text{Class} \text{Attribute})) / H(\text{Class}) + H(\text{Attribute})$. | Si | No | No | No | No | No | |

| | | | | | | | |
|---------------------------------------|--|----|----|----|----|----|----|
| Symmetrical Uncert Attribute Set Eval | Evaluates the worth of a set attributes by measuring the symmetrical uncertainty with respect to another set of attributes. $\text{SymmU}(\text{AttributeSet2}, \text{AttributeSet1}) = 2 * (\text{H}(\text{AttributeSet2}) - \text{H}(\text{AttributeSet1} \text{AttributeSet2})) / (\text{H}(\text{AttributeSet2}) + \text{H}(\text{AttributeSet1}))$ | Si | No | No | No | No | No |
|---------------------------------------|--|----|----|----|----|----|----|

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-------------------|--------|-------------|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
|-------------------|--------|-------------|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|

| Algoritmos de Búsqueda | Best First | Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point). | Si | No | No | No | No | No |
|------------------------|--------------------------|--|----|----|----|----|----|----|
| | Exhaustive Search | Performs an exhaustive search through the space of attribute subsets starting from the empty set of attributes. Reports the best subset found. | Si | No | No | No | No | No |
| | FCBF Search | Feature selection method based on correlation measure and redundancy analysis. Use in conjunction with an attribute set evaluator (SymmetricalUncertAttributeEval). | Si | No | No | No | No | No |
| | Genetic Search | Performs a search using the simple genetic algorithm described in Goldberg (1989). | Si | No | No | No | No | No |
| | Greedy Stepwise | Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected. | Si | No | No | No | No | No |
| | Linear Forward Selection | Extension of BestFirst. Takes a restricted number of k attributes into account. Fixed-set selects a fixed number k of attributes, whereas k is increased in each step when fixed-width is selected. The search uses either the initial ordering to select the top k attributes, or performs a ranking (with the same evaluator the search uses later on). The search direction can be forward, or floating forward selection (with optional backward search steps). | Si | No | No | No | No | No |
| | Race Search | Races the cross validation error of competing attribute subsets. Use in conjunction with a ClassifierSubsetEval. RaceSearch has four modes | Si | No | No | No | No | No |
| | Random Search | Performs a Random search in the space of attribute subsets. If no start set is supplied, Random search starts from a random point and reports the best subset found. If a start set is supplied, Random searches randomly for subsets that are as good or better than the start point with the same or fewer attributes. Using RandomSearch in conjunction with a start set containing all attributes equates to the LVF algorithm of Liu and Setiono (ICML-96). | Si | No | No | No | No | No |
| | Ranker | Ranks attributes by their individual evaluations. Use in conjunction with attribute evaluators (ReliefF, GainRatio, Entropy etc). | Si | No | No | No | No | No |

| | | | | | | | |
|-------------------------------|---|----|----|----|----|----|----|
| Rank Search | Uses an attribute/subset evaluator to rank all attributes. If a subset evaluator is specified, then a forward selection search is used to generate a ranked list. From the ranked list of attributes, subsets of increasing size are evaluated, ie. The best attribute, the best attribute plus the next best attribute, etc.... The best attribute set is reported. RankSearch is linear in the number of attributes if a simple attribute evaluator is used such as GainRatioAttributeEval. | Si | No | No | No | No | No |
| Scatter Search V1 | Performs an Scatter Search through the space of attribute subsets. Start with a population of many significant and diverse subset stops when the result is higher than a given treshold or there's not more improvement | Si | No | No | No | No | No |
| Subset Size Forward Selection | Extension of LinearForwardSelection. The search performs an interior cross-validation (seed and number of folds can be specified). A LinearForwardSelection is performed on each fold to determine the optimal subset-size (using the given SubsetSizeEvaluator). Finally, a LinearForwardSelection up to the optimal subset-size is performed on the whole data. | Si | No | No | No | No | No |
| Tabu Search | Performs a search through the space of attribute subsets. Evading local maximums by accepting bad and diverse solutions and make further search in the best solutions. Stops when there's not more improvement in n iterations. | Si | No | No | No | No | No |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-------------------|--------|-------------|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
|-------------------|--------|-------------|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|

| Clusters | Agglomerative Flat Clustering | This operator performs generic agglomerative clustering based on a set of ids and a similarity measure. Clusters are merged as long as their number is lower than a given maximum number of clusters. The algorithm implemented here is currently very simple and not very efficient (cubic). | No | No | No | No | No | Si |
|----------|-------------------------------|--|----|----|----|------|----|------|
| | Bagged clustering | A partitioning cluster algorithm such as kmeans is run repeatedly on bootstrap samples from the original data. The resulting cluster centers are then combined using the hierarchical cluster algorithm hclust. | No | Si | No | No | No | No |
| | CLOPE | a fast and effective clustering algorithm for transactional data | Si | No | No | No | No | Si * |
| | Cluster ensembles | combines multiple partitionings of a set of objects into a single consolidated clustering | No | Si | No | No | No | No |
| | Cobweb | Class implementing the Cobweb and Classit clustering algorithms. Note: the application of node operators (merging, splitting etc.) in terms of ordering and priority differs (and is somewhat ambiguous) between the original Cobweb and Classit papers. This algorithm always compares the best host, adding a new leaf, merging the two best hosts, and splitting the best host when considering where to place a new instance. | Si | No | No | No | No | Si * |
| | Convex clustering | an exemplar-based likelihood function that approximates the exact likelihood. This formulation leads to a convex minimization problem and an efficient algorithm with guaranteed convergence to the globally optimal solution. The resulting clustering can be thought of as a probabilistic mapping of the data points to the set of exemplars that minimizes the average distance and the information-theoretic cost of mapping. | No | Si | No | No | No | No |
| | DBScan | A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise | Si | No | No | Si * | No | Si |
| | EM | Simple EM (expectation maximisation) class. EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or you may specify a priori how many clusters to generate | Si | Si | No | Si * | No | Si * |
| | Farthest First | Cluster data using the FarthestFirst algorithm. | Si | No | No | Si * | No | Si * |
| | Filtered Clusterer | Class for running an arbitrary clusterer on data that has been passed through an arbitrary filter. Like the clusterer, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure. | Si | No | No | Si * | No | Si * |
| | Fuzzy C-means clustering | method of clustering which allows one piece of data to belong to two or more clusters | No | Si | No | Si | No | No |

| | | | | | | | |
|------------------------------|--|----|----|----|------|----|------|
| Hierarchical Clusterer | Hierarchical clustering class. Implements a number of classic agglomerative (i.e. bottom up) hierarchical clustering methods. | Si | Si | Si | Si | Si | Si |
| Kohonen's SOM | Kohonen's Self Organization Map. | No | No | Si | No | No | No |
| Kernel KMeans | Clustering with kernel k-means | No | No | No | No | No | Si |
| Kmedoids | Simple implementation of k-medoids. | No | No | No | No | No | Si |
| LVQ | Kohonen's Learning Vector Quantizers, a "supervised" clustering algorithm. | No | No | Si | No | No | No |
| Make Density Based Clusterer | Class for wrapping a Clusterer to make it return a distribution and density. Fits normal distributions and discrete distributions within each cluster produced by the wrapped clusterer. Supports the NumberOfClustersRequestable interface only if the wrapped Clusterer does. | Si | No | No | No | No | No |
| MPC KMeans | This is an implementation of the "Metric Pairwise Constraints K-Means" algorithm (see "Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In Proceedings of the 21st International Conference on Machine Learning, ICML, pages 8188, Ban , Canada, July 2004.") | No | No | No | No | No | Si |
| OPTICS | Ordering Points To Identify the Clustering Structure | Si | No | No | Si * | No | No |
| sIB | Cluster data using the sequential information bottleneck algorithm. Note: only hard clustering scheme is supported. sIB assign for each instance the cluster that have the minimum cost/distance to the instance. The trade-off beta is set to infinite so 1/beta is zero. | Si | No | No | No | No | Si |
| Similarity Comparator | Operator that compares two similarity measures using diverse metrics | No | No | No | No | No | Si |
| Simple KMeans | Cluster data using the k means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean. | Si | Si | Si | Si * | Si | Si |
| SOTA Learner | Clusters numerical data with SOTA | No | No | No | Si | No | No |
| XMeans | X-Means is K-Means extended by an Improve-Structure part In this part of the algorithm the centers are attempted to be split in its region. The decision between the children of each center and itself is done comparing the BIC-values of the two structures. | Si | No | No | Si * | Si | Si * |

| Tipo de algoritmo | Nombre | Descripción | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|-------------------|---|---|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Otros | FLR | Fuzzy Lattice Reasoning Classifier (FLR) v5.0 | Si | No | No | Si * | No | Si * |
| | Hyper Pipes | Class implementing a HyperPipe classifier. For each category a HyperPipe is constructed that contains all points of that category (essentially records the attribute bounds observed for each category). Test instances are classified according to the category that "most contains the instance". Does not handle numeric class, or missing values in test cases. Extremely simple algorithm, but has the advantage of being extremely fast, and works quite well when you have "smegloads" of attributes. | Si | No | No | Si * | No | Si * |
| | Image Processing | Image Processing | No | No | No | Si | No | No |
| | Min Max Extension | This class is an implementation of the minimal and maximal extension. All attributes and the class are assumed to be ordinal. The order of the ordinal attributes is determined by the internal codes used by WEKA. | Si | No | No | Si * | No | Si * |
| | Moléculas | Tratamiento de moléculas (traducción a texto, 3d, ...) | No | No | No | Si | No | No |
| | OLM | This class is an implementation of the Ordinal Learning Method | Si | No | No | Si * | No | Si * |
| | OSDL | This class is an implementation of the Ordinal Stochastic Dominance Learner. | Si | No | No | Si * | No | Si * |
| | SerializedClassifier | A wrapper around a serialized classifier model. This classifier loads a serialized models and uses it to make predictions | Si | No | No | No | No | Si * |
| VFI | Classification by voting feature intervals. Intervals are constructed around each class for each attribute (basically discretization). Class counts are recorded for each interval on each attribute. Classification is by voting | Si | No | No | Si * | No | Si * | |

Conclusiones

| | Disponible en WEKA 3.7.2 | Disponible en R > 2.7 | Disponible en Tanagra | Disponible en KNIME | Disponible en PASW (antes Clementine) | Disponible en RapidMiner 4.1 |
|--|--------------------------|-----------------------|-----------------------|---------------------|---------------------------------------|------------------------------|
| Algoritmos implementados de forma nativa | 168 | 24 | 13 | 9 | 10 | 34 |
| Algoritmos Importados desde Weka | 0 | 0 | 0 | 102 | 0 | 101 |
| Total Algoritmos Implementados | 168 | 24 | 13 | 111 | 10 | 135 |
| % algoritmos nativos | 100.0% | 100.0% | 100.0% | 8.1% | 100.0% | 25.2% |
| % algoritmos nativos sobre el total | 84.8% | 12.1% | 6.6% | 4.5% | 5.1% | 17.2% |

Total Algoritmos evaluados 198

Fuentes

| | |
|-----------------|---|
| weka algorithms | http://wiki.pentaho.com/display/DATAMINING/Classifiers |
| R | http://wiki.pentaho.com/download/attachments/3801462/ComparingWekaAndR.pdf |
| Tanagra | http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html |
| K-nime | http://www.knime.org/features |
| SPSS | http://www.spss.com/catalog/ |
| RapidMiner | http://aprendizagem-ua.googlegroups.com/web/rapidminer-4-1-tutorial.pdf |

Sobre Stratebi

Stratebi es una empresa española, radicada en Madrid y oficinas en Barcelona, creada por un grupo de profesionales con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes a través de Stratebi.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi son [profesores y responsables de proyectos](#) del Master en Business Intelligence de la Universidad UOC.

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source.

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.



Stratebi ha sido elegida como [Caso Éxito del Observatorio de Fuentes Abiertas de Cenatic](#).

http://observatorio.cenatic.es/index.php?option=com_content&view=article&id=429:stratebi&catid=2:empresas&Itemid=41

Observatorio Nacional del
Software de Fuentes Abiertas



Asociaciones empresariales de Software Libre empresarial en las que participamos.



TECNOLOGIAS CON LAS QUE TRABAJAMOS



ALGUNAS REFERENCIAS STRATEBI



DEMOS e INFO

- Creadores del principal Portal sobre Business Intelligence en castellano ([TodoBI.com](http://todobi.com))
- Demo [Tablero Futbolero](http://www.tablerofutbolero.es) (<http://www.tablerofutbolero.es>) (Cuadros de Mando) pedir clave en info@stratebi.com
- Demo [BI Open Source Sector Público](http://demo.stratebi.es), (<http://demo.stratebi.es>) pedir clave en info@stratebi.com
- [BI Termometer](http://todobi.blogspot.com/2010/04/checklist-para-hacer-un-proyecto.html). Checklist gratuito (más de 1.500 Kpis), para el éxito de un Proyecto BI. <http://todobi.blogspot.com/2010/04/checklist-para-hacer-un-proyecto.html>
- Video entrevista en [Portal BI-Spain](http://todobi.blogspot.com/2010/04/entrevista-sobre-business-intelligence.html), <http://todobi.blogspot.com/2010/04/entrevista-sobre-business-intelligence.html>
- Zona [YouTube Stratebi](http://www.youtube.com/user/Stratebi). [YouTube](http://www.youtube.com/user/Stratebi), <http://www.youtube.com/user/Stratebi>
- Catálogo de [Soluciones Verticales](http://www.stratebi.com/Inteletter.htm). Encuentra la tuya!!, <http://www.stratebi.com/Inteletter.htm>