Supervised Classification Algorithms in Machine Learning: A Survey and Review

Pratap Chandra Sen, Mahimarnab Hajra and Mitadru Ghosh

Abstract Machine learning is currently one of the hottest topics that enable machines to learn from data and build predictions without being explicitly programmed for that task, automatically without human involvement. Supervised learning is one of two broad branches of machine learning that makes the model enable to predict future outcomes after they are trained based on past data where we use input/output pairs or the labeled data to train the model with the goal to produce a function that is approximated enough to be able to predict outputs for new inputs when introduced to them. Supervised learning problems can be grouped into regression problems and classification problems. A regression problem is when outputs are continuous whereas a classification problem is when outputs are categorical. This paper tries to compare different types of classification algorithms precisely widely used ones on the basis of some basic conceptions though it is obvious that a complete and comprehensive review and survey of all the supervised learning classification algorithms possibly cannot be accomplished by a single paper, but the references cited in this paper hopefully cover the significant theoretical issues and our survey has been kept limited to the widely used algorithms because the field is highly growing and not possible to cover all the algorithms in a single paper. One more point to be mentioned here that any study of complex procedure like neural networks has not been included as it has been tried to keep the content as much simple as possible.

Keywords Machine learning · Classification algorithm · Supervised learning · Accuracy

P. C. Sen (✉) · M. Hajra · M. Ghosh
Department of Computer Science and Engineering, Institute of Engineering and Management, Kolkata, India
e-mail: pchsen97@gmail.com

M. Hajra
e-mail: mahimarnab2014@gmail.com

M. Ghosh
e-mail: mitadrughosh100@gmail.com

J. K. Mandal and D. Bhattacharya (eds.), *Emerging Technology in Modelling and Graphics*, Advances in Intelligent Systems and Computing 937,
https://doi.org/10.1007/978-981-13-7403-6_11

1 Introduction

This paper will focus on summarizing the key advantages of different, widely renowned, and most frequently used machine learning algorithms used for classification task and to do a comparative study on them to find out which algorithm works best based on different parameters so that while doing any classification task, one can know when to use which algorithm to get best result. Classification can be performed on both the unstructured or structured dataset. Members of a dataset are classified according to some given label or category and for new input instances, the class or label that will be assigned to it is predicted by this technique. Some terms frequently used in this field are also introduced here with explanation. A classifier algorithm is an algorithm that learns from the training set and then assigns new data point to a particular class. A classification model concludes some valid mapping function from training dataset and predicts the class label with the help of the mapping function for the new data entry. An attribute or feature is a parameter found in the given problem set that can sufficiently help to build an accurate predictive model. There are different types of classification task.

Binary classification is a classification with two possible outcomes. For example, weather forecast (it will rain or not), spam or fraud detection (predict whether an email is spam or not).

Multi-label classification is a classification task with more than two possible outcomes. For example, classify academic performance of students as excellent or good or average or poor.

In classification, a sample can even be mapped to more than one tag labels. For example, a sample of news article can be labeled as sport article, an article about some player, and an article about a certain venue at the same time.

A classification model can be built by following steps:

1. Collect and clean the dataset or data preprocessing.
2. Make the classifier model initialized.
3. Split the dataset using cross-validation and feed the classifier model with training data. Python-based scikit-learn package has inbuilt methods named fit-transform (X, Y)/fit(X, Y) that map the input data member set X and corresponding label set Y to prepare the classifier model.
4. Predict the label for a new observation data. There is also a method predict (X) that returns the mapped label Y for the input instance X.
5. Evaluate error rate of the classifier model on the test dataset (Fig. 1).

So, the very first step is to collect the dataset. The most relevant attributes/fields/features are to be figured out next, which is called feature extraction. "Brute-force" method is the simplest one which isolates the most relevant/informative attributes by measuring everything available. Besides, there is a process named feature subset selection that identifies and eliminates as many redundant, irrelevant, and unnecessary attributes as possible.[1] Secondly, it is required to do some data prepro-

[1] See Ref. [28].

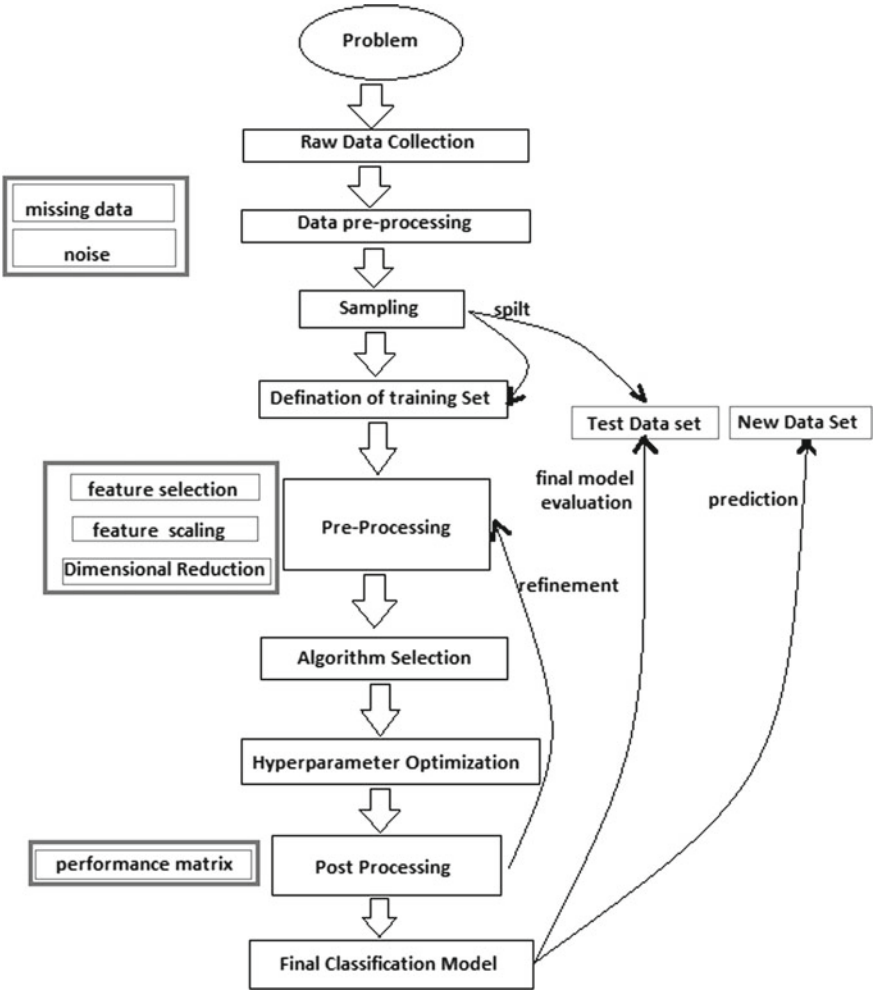

Fig. 1 Working flowchart of supervised classification model

cessing.[2] The dataset often contains noise (outliers) and missing feature values and some categories that needs to be converted into dummy variable. Various methods are available to deal with missing data[3] which should be paid attention. Thus, it requires a well data preprocessing.

[2] See Ref. [5].

[3] See Ref. [27].

2 General Issues of Supervised Classification

There are contemporary techniques by Batista and Monard [4] to detect outliers. All the data analysis is done on significant samples taken from survey dataset and there available a few number of approaches for sampling purpose. It leads the dataset toward significant dimensionality reduction and makes the different data mining algorithms capable of working more efficiently and also faster. Due to inter-dependencies of many attributes, often results inordinate influences on the accuracy level of these classification models heavily. Feature transformation is an efficient technique that constructs new features or dummy features from the originally given feature set to stand against these unduly effects on model by feature inter-dependencies. This method performs the classification task with the help of already classified/categorized training data. The input point under consideration and the corresponding desired output is already known for the training dataset. When the supervised learning algorithms for classification purpose are fed with data that is when it is trained with the already known training dataset, it become capable to generalize the new unseen data and predict corresponding class. Here, the dataset is about the *salary of employees of a company, attributes: 7 and instances: 48,842.* From the census bureau database,[5] this data sample is extracted.

The next section describes the basic definition and working method of most widely used supervised classification machine learning algorithms with a brief review so that the survey explanation can be well understood. Those algorithms learn through different ways and based on that we can classify them.

2.1 Logically Learning Algorithm

Logic-based algorithms deal with the problem with step by step data streaming with functioning a logic in each step. Here **decision tree** has been overviewed as a classical example of logic-based algorithm. It is statistical technique used in regression and classification. Given a data of features along their correct labels, decision tree is an example of logically learning supervised algorithm mainly used for classification that generates a set of decision sequences that if followed will lead to predicting the label of an unlabeled data.

A. *Definition* A decision supported tool that is represented as a logical tree having a range of conditions and conclusions as nodes and branches that connects the conditions with conclusions. Decision tree takes some decision in every stage of proceeding and considers some alternative choices of action, and among them it selects the most relevant alternatives. A decision tree is a mathematical model used to make decisions. It uses estimates and probabilities to calculate

[4] See Ref. [13].

[5] https://www.census.gov/DES/www/welcome.html.

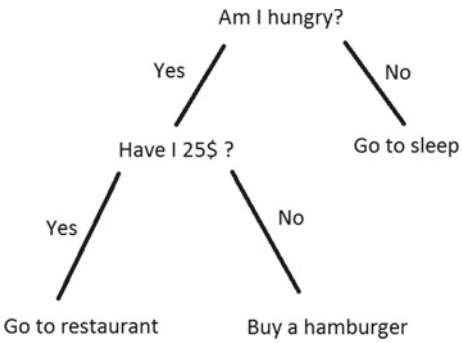

Fig. 2 A decision tree to decide how to have food

likely outcomes and helps to decide whether the net gain from a decision is worthwhile.

B. *Method* An object defined by a set of features is an input here, output is a decision for the corresponding input. In the tree structure, each internal node tests an attribute and each of the nodes is assigned to a classification. Following is an example of how a decision tree looks like (Fig. 2).

C. *Details* Like SVM, decision tree also works fine for both categorical and continuous dependent instances. This method actually constructs a sequential decision stream-based model on actual values of features in the dataset. Decisions are split into tree-like structure. A decision is made at every node of this tree unless a prediction is made for a certain input data item. Decision trees are trained on data for classification problems. This algorithm works very fast often with satisfactory accuracy. It is a big favorite and widely used in machine learning and also works efficiently on comparatively less amount of dataset. This algorithm splits the set of data items into two or more homogeneous sets based on most significant attribute to make as distinct groups as possible. To split up the data into different groups, various techniques like information gain, chi-square, Gini, entropy, etc. are used. Entropy has a well discriminatory power for classification. By the name it is known, it represents some kind of randomness. Entropy defines the amount of randomness in features and measures discriminatory capability of an attribute for the classification problem. Information gain ranks the attributes for the purpose of filtering at given node.

D. *Examples of decision tree algorithms* Widely used algorithms are *ID3, C4.5, CART.*

E. *Advantage* Decision tree is a simple method, easy to understand and visualize, fast, requires less data preprocessing, and can deal with both categorical and numerical data.

F. *Disadvantage* Sometimes this algorithm may lead to a complex tree structure not generalized enough, besides it is quite unstable model.

G. Application It has applications in various fields like *determining galaxy counts, control system, financial analysis,* etc.

In the next session, the latest method of class prediction has been introduced named as support vector machine.

2.2 Support Vector Machine

Support vector machine or in short SVM is an advanced supervised algorithm that can deal with both regression and classification tasks though it is more favorable for the classification. It can handle multiple continuous and categorical instances.

A. *Definition* It represents the dataset items or records each having "*n*" number of features plotted as points in a n-dimensional space segregated into classes by a clear margin widest possible known as hyperplane. Into that same n-dimensional space, data items are then mapped to get the prediction of the category they belong to, based on the side of hyperplane they fall.
B. *Method* The coordinates of these data items are actually termed as "support vectors." If there is a linear type hyperplane present between these classes, the problem becomes very simple. But, another question arises is the manual addition of these attributes to have a hyperplane is actually necessary or not. The answer is "Not Really" as this algorithm uses a technique known as "kerneltrick." Kernels are nothing but a group of functions which transforms the low-dimensional input space into a higher-dimensional space. In simple words, what it does is, by performing some typical data transformations that are actually complex in nature, does some conversions and non-separable problem becomes separable problem. After that, it figures out the procedure to segregate the data according to the defined labels.
C. *Details* It is the newest supervised classification technique.[6] An example is depicted in the illustration below. Here, two kinds of objects are considered either belonging to class GREEN or to class BLUE. The segregating boundary has all GREEN objects on the right side of it and on the left, all the BLUE objects lie. A new observation object when plotted in the n-dimensional vector space is labeled as GREEN if it lies in the right side of the separating boundary or hyperplane, otherwise it is tagged with the label BLUE (Fig. 3).
   The shown figure depicts a linear classifier. Most of the classification tasks are not really simple like this one and often leads to complex operations that should be executed in order to do an optimal segregation in case of some complex structures. SVM can be classified in two categories: C-SVM classification and nu-SVM classification.
D. *Advantages* It shows a noticeable hike in performance where the "*n*" of the n-dimensional space is greater than the total size of sample set. Thus, it is a good choice to pick this algorithm while dealing high-dimensional data. If the hyperplane is well built, it shows high performance. This is also memory efficient

[6] See Ref. [12].

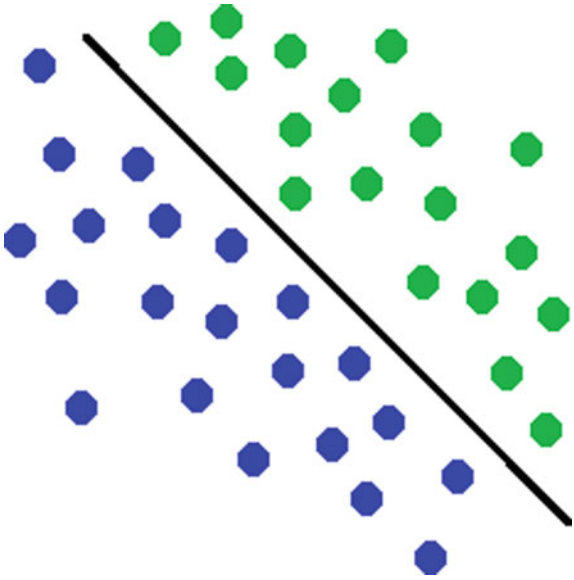

Fig. 3 A depiction of SVM classification with hyperplane

because of the use of subset training points in the decision function (priorly known as support vectors).

E. *Disadvantages* One flaw present here is that training time is quite high compared to others so if the dataset is very large then the prediction task turns out to be slow noticeably. When target classes are overlapping that is the dataset has more noise, its performance also decreases. In case of probability estimation calculation, a *n*-fold cross-validation is used which is again quite expensive for computation time.

F. *Applications* This is used in stock marketing for various predictions besides it has various applications in *bioinformatics, face detection, classification of images, handwriting recognition, classification of images,* etc.

Unlike the last one discussed, there is a branch of algorithm that learns predicting task through broad statistical description and this one has been explained next.

2.3 Statistics-Based Algorithm

Statistics-based algorithms generalize problems with help of distributive statistics and look into the distribution structure to continue the predicting task. Here, ***Naïve Bayes*** has been explained as a popular example of statistics-based algorithm Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling. This is a collection of classification techniques based on Bayesian theorem of probability.

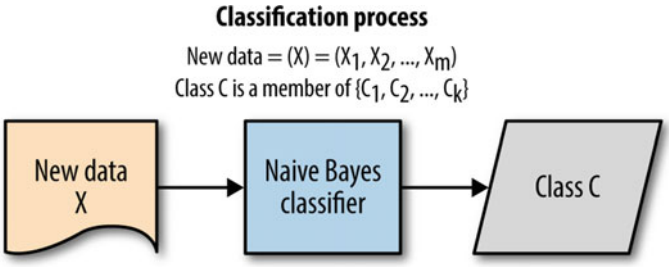

Fig. 4 Classification model of Naïve Bayes

A. *Definition* Naive Bayes classifier produces the probabilities for every case. Then it predicts the highest probability outcome. A naive assumption is made that the features are independent, classification of every pair of features is independent of each other.
B. *Details* These classifiers are capable of handling an arbitrary number of independent continuous and categorical variables efficiently. Let us consider a set of variables, $X = \{x_1, x_2, x_3,\ldots, x_t\}$; it is required to find out the posterior probability for the event C_j from the sample space set $C = \{c_1, c_2, c_3,\ldots, c_t\}$. Simply, the predictor is X and C is the set of categorical levels present in the dependent variable. Applying Bayes' rule:

$$P\left(C_j|x_1, x_2, x_3, \ldots, x_t\right) \cdot P\left(x_1, x_2, x_3, \ldots, x_t\middle|C_j\right)P\left(C_j\right)$$

where $P(C_j\ |x_1, x_2, x_3,\ldots, x_t)$ is the posterior probability that is the probability of the event X belonging to C_j is indicated. In Naive Bayes, there is an assumption that the conditional probabilities of the independent variables have statistical independence. Using Bayes' rule, a new case X is labeled with a class level C_j that accomplishes the highest posterior probability. Although this naive assumption that the predictor variables are independent of each other is not always accurate actually. This assumption makes the classification process simpler, as it allows the class conditional densities $P(x_d\ |C_j)$ to be calculated for each variable separately and thus a multidimensional task is reduced to some one-dimensional tasks. More precisely, it converts a high-dimensional density estimation task to a one-dimensional kernel density estimation. Classification task remains unaffected as this assumption does not greatly affect the posterior probabilities, mainly in regions located closely around the decision boundaries (Fig. 4).
C. *Different ways of Naive Bayes Modeling* Naïve Bayes can be modeled in several different ways including normal, gamma, and Poisson density functions.
D. *Advantages* Naive Bayes algorithm does not require huge dataset, for estimation of required parameters, a small-sized training data is good enough here. It also performs explicit probability calculation for hypothesis. A useful prospect to

understand various learning algorithm is also provided by this method. In the comparison with other subtle methods, this classifier is also intensely fast. It can solve diagnostic problems efficiently.

E. *Disadvantage* This method is comparatively known to be a bad estimator.
F. *Application* It has various applications *like recommendation system, text classification, spam filtering, real-time prediction,* etc.

One more renowned process of supervised machine learning is lazy learning which is also known as instance-based learning which has been described in next section.

2.4 Lazy Learning Algorithm

There is another renowned category that falls under the title of statistical methods. This method in general known as "Instance-based learning" which delays the process of generalization until the classification task is performed resulting the naming of this algorithm with the tag of "lazy" and widely known as lazy learning algorithm. Here. **kNN** or K-nearest neighbor algorithm has been discussed as a fine example of instance-based learning or lazy learning. kNN is one of the easiest to understand and very simple classification algorithms available. Despite of simplicity, it can give highly competitive results. It can deal with both of the classification and regression types predictive problems. However, it is more concisely used to perform and execute classification task.

A. *Definition K*-nearest neighbor stores all available records and predicts the class of a new instances giving attention to similarity measurements from the nearest neighbors in likelihood. This classification technique is known to be lazy learning method because it keeps the data members stored simply in efficient data structures like hash table by virtue of which computation cost becomes less to check and apply the appropriate distance function between the new observation and all k number of different data points stored and then come to any conclusion about the label of the new data point, without constructing a mapping function or internal model like other classification algorithms. Result is obtained from a simple majority support of the k number of nearest neighbors of each new data point.
B. *Details* The "*K*" in k-NN algorithm is "k" number of nearest neighbors whose vote is taken to predict label for a new record around which those neighbors are situated. Let us say $K = 3$. Then a circle with new data item as center will be visualized just as big as to enclose only three nearest neighbor data points on the plane and the virtue of the distance between the record and each of the neighbors will decide the label of the new record. For a given *K*-value, boundaries of each class can be built. These boundaries can efficiently segregate one class from another. It is observed that with increasing value of *K*, the boundary becomes smoother. If *K* increases to a very high vale and finally tends to infinity, it all becomes finally one class or the one with the total majority.

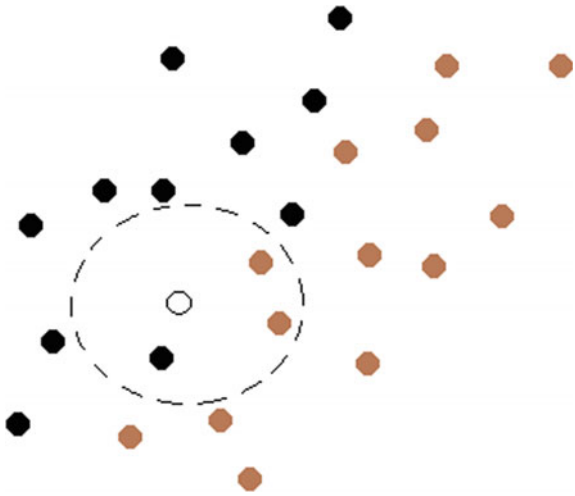

Fig. 5 An example of K-nearest neighbor algorithm approach

There are two parameters: the validation error rate and training error rate that need to be accessed and tested on different *K*-value (Fig. 5).
Here, a new member has been shown by a uncolored circle and its class has to be predicted based on the distance from its three nearest neighbors as we are considering three nearest neighbors so the value of k is 3 here and so a circle having center at the new member has been drawn covering only three previous members. Now, by judging the shortest distance, it is resulted that the new member will be black. Thus, we can predict the label of an unclassified record considering its distance from nearest members, nearer the member more the effect of its label on the new record.

C. *Advantages* Powerful against noisy training data, works effectively with large training data, besides implementation is simple.
D. *Disadvantage* One big flaw of this algorithm is for every new instance, all the distance from *K* neighbors needs to be calculated again and again which leads to high computational time consumption. The value of *K* needs to be determined correctly for lower error rate.
E. *Applications* It is often seen that kNN is used in search applications where "similar" items are searched by the user; that is, when the task is some kind of the form of "find items similar to this one[7]". It is highly recommended *in e-discovery packages* and for *recommended system* also.

In the next section, the comparative study of above-mentioned algorithms has been done with relative to some important parameters and in the end, the overall accuracy of these algorithms has been given and this test has been done on the earlier mentioned dataset.

[7] https://www.quora.com/Industry-applications-of-the-K-nearest-neighbor-algorithm.

Table 1 A comparative study on widely used four supervised classification algorithm

Comparison parameters	Decision tree	Naive Bayes	k-NN	SVM
Learning speed	Average	Best	Best	Worst
Classification speed	Best	Best	Worst	Best
Performance with presence of missing value	Average	Best	Worst	Good
Performance with non-relevant features	Average	Good	Good	Best
Noise tolerance	Good	Average	Average	Good
Performance on discrete/binary attributes	Good	Average	Average	Worst
Tolerance with parity problems	Good	Worst	Worst	Average
Clarity on Classification prediction	Best	Best	Average	Worst
Handling of model parameter	Average	Best	Average	Worst
Overall accuracy	Good (84.13%)	Worst (80.14%)	Good (83.65%)	Best (84.94%)

3 Comparison Table

So, here is the relative comparison between widely used and most popular supervised classification algorithms. Accuracy is measured using confusion matrix. Here, accuracy is defined as ratio of correct predictions to all observations. It is the most instinctive performance measure. The comparison of accuracy is done by applying the algorithms on earlier mentioned dataset. This paper describes different schemes of supervised learning and their parameters which influence to achieve accuracy rather than just detailed study over them (Table 1).

References

1. J.H. Friedman, Regularized discriminant analysis. J. Am. Stat. Assoc. **84**(405), 165–175 (1989)
2. N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers. Mach. Learn. **29**, 131–163 (1997)
3. N. Friedman, D. Koller, Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. Mach. Learn. **50**(1), 95–125 (2003)

4. R.G. Cowell, Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models, in *Proceedings of 17th International Conference on Uncertainty in Artificial Intelligence*
5. R.L. De Mantaras, E. Armengol, Machine learning from examples: inductive and lazy methods. Data Knowl. Eng. **25**(1–2), 99–123 (1998)
6. D. Heckerman, C. Meek, G. Cooper, A Bayesian approach to causal discovery, in *Computation, Causation, and Discovery*, ed. by C. Glymour, G. Cooper (MIT Press, Cambridge, 1999), pp. 141–165
7. N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study. Intell. Data Anal. **6**(5), 429–449 (2002)
8. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, ed. by D.E. Rumelhart, J.L. McClelland et al. (MIT Press, Cambridge, MA), pp. 318–362.
9. A. Roy, On connectionism, rule extraction, and brain-like learning. IEEE Trans. Fuzzy Syst. **8**(2), 222–227; L. Breiman, Bagging predictors. Mach. Learn. **24**, 123–140
10. I.J. Good, *Probability and the Weighing of Evidence* (London, Charles Grin)
11. N.J. Nilsson, *Learning Machines* (McGraw-Hill, New York)
12. B. Cestnik, I. Kononenko, I. Bratko, Assistant 86: a knowledge elicitation tool for sophisticated users, in *Proceedings of the Second European Working Session on Learning*, pp. 31–45
13. B. Cestnik, Estimating probabilities: a crucial task in machine learning, in *Proceedings of the European Conference on Artificial Intelligence*, pp. 147–149
14. T. Cover, P. Hart, Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **13**(1), 21–27 (1967)
15. W. Cohen, Fast effective rule induction, in *Proceedings of ICML-95*, pp. 115–123
16. J.M. Kalyan Roy, Image similarity measure using color histogram, color coherence vector, and sobel method. Int. J. Sci. Res. (IJSR) **2**(1), 538–543 (2013), India Online ISSN: 23197064
17. A. Smola, S. Vishwanathan, *Introduction to Machine Learning* (United Kingdom at the University Press, Cambridge, 2010)
18. R.G. Cowell, Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models, in *Proceedings of 17th International Conference on Uncertainty in Artificial Intelligence* (2001)
19. W. Gerstner, *Supervised learning for neural networks: a tutorial with JAVA exercises*
20. R. Olshen L. Breiman, J.H. Friedman, "Classification and regression trees." Belmont CA Wadsworth International group, 1984. B. C. U. P.E.tgoff, "Multivariate decision trees: machine learning," no. 19, 1995, pp. 45–47.
21. T. Dietterich, M. Kearns, Y. Mansour, Applying the weak learning framework to understand and improve C4. 5 (Sanfrancisco, Morgan), pp. 96–104
22. Kufmann, in *Proceeding of the 13th International Conference on Machine Learning* (1996)
23. K.M.A. Chai, H.L. Chieu, H.T. Ng, Bayesian online classifiers for text classification and filtering, in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (August 2002), pp. 97–104
24. T. Elomaa, The biases of decision treepruning strategies (Springer, 1999), Lecture Notes in Computer Science, vol. 1642, pp. 63–74
25. A. Kalousis, G. Gama, On data and algorithms: understanding inductive performance. Mach. Learn. **54**, 275–312 (2004)
26. P. Brazdil, C. Soares, J. Da Costa, ranking learning algorithms: using IBL and meta-learning on accuracy and time results. Mach. Learn. **50**, 251–277 (2003)
27. G. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning. Appl. Artif. Intell. **17**, 519–533 (2003)
28. J. Basak, R. Kothari, A classification paradigm for distributed vertically partitioned data. Neural Comput. **16**(7), 1525–1544 (2004)
29. A. Blum, Empirical support for winnow and weighted-majority algorithms: results on a calendar scheduling domain. Mach. Learn. **26**(1), 5–23 (1997)

30. A. Bonarini, *An Introduction to Learning Fuzzy Classifier Systems* (2000), Lecture Notes in Computer Science, vol. 1813, pp. 83–92
31. R. Bouckaert, Choosing between two learning algorithms based on calibrated tests, in *Proceedings of 20th International Conference on Machine Learning* (Morgan Kaufmann, 2003), pp. 51–58
32. R. Bouckaert, *Naive Bayes Classifiers that Perform Well with Continuous Variables* (2004), Lecture Notes in Computer Science, vol. 3339, pp. 1089–1094
33. P. Brazdil, C. Soares, J. Da Costa, ranking learning algorithms: using IBL and meta-learning on accuracy and time results. Mach. Learn. **50**, 251–277 (2003)
34. L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees* (Wadsforth International Group, 1984)
35. L. Breiman, Bagging predictors. Mach. Learn. **24**, 123–140 (1996)
36. L.A. Breslow, D.W. Aha, Simplifying decision trees: a survey. Knowl. Eng. Rev. **12**, 1–40 (1997)
37. H. Brighton, C. Mellish, Advances in instance selection for instance-based learning algorithms. Data Min. Knowl. Disc. **6**, 153–172 (2002)