Machine Learning: What, Why, and How? 16

Salma Jamal, Sukriti Goyal, Abhinav Grover, and Asheesh Shanker

16.1 Introduction

Machine learning involves a set of algorithms which deal with the automatic recognition of hidden patterns in data and making predictions about the future unseen data (Kohavi and Provost 1998). It has been defined by Arthur Samuel (1959) as "Field of study that gives computers the ability to learn without being explicitly programmed" (Simon 2013). As quoted from Tom M. Mitchell's definition of machine learning which is "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell 1997). Due to its importance, machine learning has become the integral part of analysis pipeline in this era of ever-increasing amounts of data.

The fundamental part of machine learning is to learn from the known properties of the data and from past experiences and then give accurate predictions on new cases based on learning from the trained sets. A specific set of methods/algorithms including decision tree-based learning, support vector machines, Bayesian networks, instance-based learning, and artificial neural networks has been used for training the model system. Various parameters are needed to be tuned to optimize the performance of the learned model systems (Bishop 2006). Machine learning helps in

S. Jamal (✉) · S. Goyal
Department of Bioscience and Biotechnology, Banasthali Vidyapith, Rajasthan, India

A. Grover
School of Biotechnology, Jawaharlal Nehru University, New Delhi, India

A. Shanker
Department of Bioscience and Biotechnology, Banasthali Vidyapith, Rajasthan, India

Department of Bioinformatics, Central University of South Bihar, Gaya, Bihar, India

A. Shanker (ed.), *Bioinformatics: Sequences, Structures, Phylogeny*,
https://doi.org/10.1007/978-981-13-1562-6_16

finding solutions to a wide range of problems, and its applications include search engines, information retrieval, bioinformatics, cheminformatics, disease diagnosis, speech and handwriting recognition, image processing, and many more to mention.

16.1.1 Types of Machine Learning

Machine learning is usually classified into two types based on the availability of the data and the input given to the learning system; these include supervised learning and unsupervised learning approach (Stuart and Peter 2003). A third type of learning approach, rather less frequently used, is the reinforcement learning approach.

16.1.1.1 Supervised Learning

In the supervised learning approach which is also known as the predictive approach, the task is to predict for the unknown from a labeled set of training data (Mohri et al. 2012). The training data comprises a set of input objects say i, which are basically represented by vectors describing the properties of the objects and the corresponding output classes, forming input-output pairs. Consider the input space as X and output space as Y in the training data and an example which lies in the form, $\{(x_i, y_i)_{i=1 \text{ to } N}\}n$; then x_i is the feature vector of the i-th object of the training data and y_i where y is the output label of the i-th object. These properties, which may be anything, say the age and height of a person contain information about the input objects and are known as features or attributes. It is advised that the number of features must not be too large as it would result in many dimensions which may confuse the learning algorithm. A supervised learning approach examines the training data and results in a function using which it further attempts to determine the output class for unobserved cases (Murphy 2012). Various supervised learning algorithms use a subset of training data, known as a validation set, to determine the accuracy of the learning algorithm by means of cross-validation.

16.1.1.2 Unsupervised Learning

In unsupervised learning, also known as descriptive learning, the data is unlabeled and the task is to find some hidden interesting patterns in the data (Murphy 2012). This differentiates the unsupervised learning approach from the supervised learning and the reinforcement learning. The data consists of only input space which lies in the form $\{(x_i)_{i=1 \text{ to } N}\}$, and there are no input-output pairs. This algorithm does not use any explicit output labels; therefore, it uses various other approaches such as clustering of the input data based on the similarities in the features and further placing the unseen instances into one or the other cluster (Daumé 2012). Other approaches used by unsupervised learning include discovering the most contributing attributes employing dimensionality reduction techniques like principal component analysis (PCA) and also by determining the correlated variables using graph theory.

16.1.1.3 Reinforcement Learning

Reinforcement learning is a less commonly used area of machine learning in which learning is associated with a reward or a punishment and the behavior of the learning algorithm or agent is based on a set of environment states. The task of the agent is to determine the ideal behavior and maximize the rewards (Sutton and Barto 1998).

16.1.2 Applications of Machine Learning

Machine learning is used widely in the plethora of tasks, mostly for classification purposes; some examples include email filtering, web page ranking, disease diagnosis, face detection, and many more (Alex and Vishwanathan 2008).

Through machine learning, a system can be generated which can sort the emails by redirecting received emails containing significant information into the inbox, sent mails in the outbox, and other emails about discounted products and offers into the spam section. The learning algorithm is trained to classify the mails based on the information they carry and process them automatically as spam or not spam (Tretyakov 2004).

One of very interesting application of machine learning is information retrieval where when a user enters a query in a search engine, it displays a list of web pages sorted according to the significance of the information they contain matching the user's query term. The training data consists of the content of various web pages, the link structure, etc., and the learned system classifies the information as relevant or irrelevant (Yong et al. 2008).

Another application of machine learning is face detection in security systems where the computer system or the software identifies a person or tells if the face is unknown. The system takes into account various features like the complexions, person wearing glasses or not, hairstyle, expressions, shape and size of eyes, nose, etc. and learns to recognize a person based on these features (Brunelli and Poggio 1993).

Machine learning can also be used in the disease diagnosis. The learned model predicts if a person suffers from a particular disease or not. The system uses all the data related to the disease including the associated symptoms, the histological data, the time period, and regional information as attributes and deduces if a person is a sufferer or non-sufferer (Sajda 2006).

Translation between two documents is a tedious task as one needs to fully understand a text prior to its translation plus it involves huge chances of loss of accuracy of information and grammatical errors. Machine learning has proved to be quite successful in automatic translation of the documents making it fast and accurate.

Optical character recognition (OCR) involves electronic translation of images of the handwritten documents, say scanned documents, into machine-readable language like ASCII code. The technique is used for entering data into the system for a wide range of documents that include passports, bank statements, printed receipts, and other different documents. The role of machine learning in OCR is to classify a

character into a character region; the features of the character regions are already stored in the classifier. Whenever a new character comes across, the classifier tries to match the properties of the character to the character region and assigns a region best matching the properties (Dervisevic 2006).

There are other areas where machine learning is applied efficiently like in bioinformatics, computational and systems biology, and cheminformatics which include bioactivity data analysis, gene expression data classification and gene prediction, protein structure prediction, identification of biomarkers, and many more. Various learning algorithms have been increasingly used for analyzing gene expression data from microarrays for more accurate phenotypic classification of diseases and diagnosis in addition to the prediction of novel disease-associated genes (Jamal et al. 2017; Moore et al. 2007). Protein structure prediction, which is one of the most complex problems in structural biology and bioinformatics, has also been addressed by machine learning methods (Cheng et al. 2008). Machine learning algorithms have been widely used for generating predictive classification models which could identify probable active compounds from large unscreened chemical compounds libraries (Singh et al. 2016).

16.2 Steps to Build a Machine Learning Model

16.2.1 Inputs in the Form of Instances and Features

Machine learning is basically training a model using some objects and then performing predictions on some other objects. An instance can be any example, object, case, or item to be classified by the learned model system. The instance is an object used by the learning algorithm for training a model and on which the model carries out the predictions. These objects or instances are represented by feature vectors (Christopher 2006). Features, also known as descriptors or attributes, are the set of predetermined quantifiable properties of an object; say in flower classification, an object is the flower, so the features might be color of the flower, number of sepals, number of petals, sepal length, petal length, etc., the objects are encoded as features, and then these features are used to decide the class for the object (Murphy 2012). If the instance is a molecule, the chemical information encoded within the molecule is transformed into a mathematical representation of that molecule which is known as the molecular descriptors or features. A number of commercial and free molecular descriptor generation software are available which include ADAPT (Valla et al. 1993), ADMET Predictor [Simulations Plus Inc., Lancaster, CA], Dragon [Talete, Milano, Italy], JOELib (JOELib/JOELib2 cheminformatics library), Marvin Beans [ChemAxon], Molecular Operating Environment (MOE; Chemical Computing Group Inc. 2015), PaDEL (Yap 2011), PowerMV (Liu et al. 2005), and many more.

Choosing a subset of features which contain relevant information toward the classification to overcome the dimensionality curse and simplify the classification process is a primary step in machine learning.

16.2.2 Feature Selection

Feature selection is a method that involves discovering an optimal subset of features from the original set of features. The accuracy and robustness of some machine learning algorithms depend on the number of features chosen to represent the objects/instances (Mitchell 2014). Feature selection techniques, one of the most significant steps in machine learning, are used to simplify and fasten the learned system generation process and increase the accuracy of the classification by reducing the dimensionality and noise from the data. The irrelevant features, those which do not give any information in making predictions by the classifier, add noise and increase the complexity of the data. In feature selection, descriptors which contribute most toward the prediction task have been searched. A subset of relevant features, though may be a few in number, prove to be extremely important for the prediction task (Daumé 2012). The remaining irrelevant features are not considered during the training process.

Another issue to be taken care of is the redundancy in the descriptors. If two features have very similar values for the objects, then they are highly correlated and thus can be discarded without much information loss (Ethem 2009).

The basic principle behind feature selection techniques is testing each subset of features and finding the subset which decreases the error the most. Two methods are employed in the subset selection process, backward selection and forward selection. The backward selection method starts with the complete set of features and removes the features by deleting one feature at a time. The process continues until the removal of a feature increases the error. In forward selection algorithm, the process starts with an empty set of features, and then the features are added one by one till the error is decreased (Guyon and Elisseeff 2003).

16.2.3 Methods to Search Features

16.2.3.1 Best First

The best-first search approach employs greedy hill climbing algorithm and derives a subset of features. Once this subset is obtained, its features are examined for the information gain. A new feature is defined on the basis of the information available from the features of this subset, and then previously chosen features are removed. The procedure is repeated until all the features have been taken into account (Dang and Croft 2010).

16.2.3.2 Exhaustive Search

Exhaustive search is a simple approach which starts from a random point, selects an empty set of features, and then performs a comprehensive search over all probable subsets of features (Karuppasamy et al. 2008).

16.2.3.3 Genetic Search

The genetic search approach uses the genetic algorithm and finds an optimal feature subset. The data is in the binary form, i.e., a feature is either present or absent from the subset. Further the fitness function values, the larger the better, for these features are calculated. This process is continual till better solutions are obtained (Tiwari and Singh 2010).

16.2.3.4 Greedy Stepwise

The algorithm performs a greedy forward and backward search throughout the feature space. The algorithm either starts with no attributes or does a random selection of attributes considering the most descriptive attributes and discards the remaining ones. The process stops when addition or deletion of attributes effect the accuracy of prediction (Farahat et al. 2011).

16.2.3.5 Scatter Search

Unlike other feature selection algorithms, the scatter search is a directed search which includes a predefined reference subset of diverse attributes. This subset acts as a reference point and an attempt is made to increase its diversity. Further, the search is applied and the reference set is updated, and the procedure terminates when a predecided threshold is achieved or the search no longer produces improved results (López et al. 2006).

16.3 Machine Learning Algorithms

16.3.1 Naïve Bayes

The Naïve Bayes (NB) algorithm is a simple classifier that employs Bayes formula and estimates the probability of an object belonging to a particular class. The classifier assumes that the occurrence of one feature does not relate to the presence or absence of any other feature and considers all attributes as statistically independent of each other. For example, an animal is an elephant if it has large ears and has trunk and tusks; all these features are dependent on each other, but the Naïve Bayes classifier considers all these features as independently contributing toward the probability of the animal of being an elephant. The algorithm computes the posterior probability of each class, and the object is placed in the class which is the most probable (Friedman et al. 1997). The Bayesian classifier provides a flexible approach to machine learning where the probability for each hypothesis can be increased or decreased and the test instances are assigned the class based on the observed data, i.e., it calculates the prior probability and then the posterior probabilities. The Bayesian learning-based NB classifier finds its application in a wide range of classification problems (Mitchell 2014).

16.3.2 Random Forest

Random Forest (RF) classifier is an ensemble classifier developed by Leo Breiman. The algorithm uses decision trees which are generated by randomly selecting the features from the training data. The nodes of the tree are the features, the branches are the values, and the edges correspond to the classes. Each node in the tree links to an attribute, and each branch from this node represents a value of that attribute. The classifiers consist of a forest of trees which are then used to categorize a new instance. Initially the tree, at each node, uses the subset of features chosen randomly, and the best subset is used to split the node. The attribute which has the maximum information gain provides the best prediction and thus is selected as the decision-making attribute (Ali et al. 2012). The classifier does not involve pruning of the trees, and each tree is grown as long as possible, and the process is terminated when each attribute has been incorporated at least once or if all the training instances associated with that attribute have the same value. When a test instance is encountered, each tree is examined for the features at the nodes, and the instance is assigned the class which is the output of the larger number of trees (Mitchell 2014).

16.3.3 Support Vector Machines

Support vector machines (SVM) are non-probabilistic classifiers that use a kernel function and attempts to find a hyperplane in a high-dimensional space. The algorithm tries to find a linearly separating hyperplane amid the two classes, and then the margins of the hyperplane are maximized. The support vectors lie on either side of the margins of the hyperplane. In case of high-dimensional data, the algorithm makes use of kernel functions which convert the original input space into nonlinear input space. For SVM to perform multiclass classification task, the algorithm will reduce it to several binary classification problems. The various kernel functions include linear, radial basis function (RBF), polynomial, and the sigmoid kernel. The efficiency of the SVM classifier depends on the choice of the kernel function and kernel parameters and one more parameter, which is the trade-off between training error and the margin (Platt 1998). The use of the type of the kernel function depends on the type of the classification problem; however, RBF is the kernel of choice in most cases. The SVM training generates learned model system which classifies the test instances into any of the two categories which are on either side of the separating hyperplane (Hsu et al. 2003).

16.3.4 Artificial Neural Network

Artificial neural network (ANN) is a widely used algorithm inspired by the central nervous system and works on the same principle as the human brain works. ANNs are generally complex interconnected neurons which transfer messages to and from each other. The algorithm consists of three layers, an input layer where the input is

given, a hidden layer where the processing takes place, and an output layer which records the output. A number of features are fed into the input unit which is then forwarded to the hidden unit, and the hidden unit further feeds these features to the single output layer. The edges that connect these layers are the weighted neurons, and during the training phase, the algorithm tries to fluctuate these weights for the system to learn to connect between the input and output layers (Mitchell 1997). Initially, the weights are varied in the hidden layer based on the features in the input layer following which the output units are computed based on the hidden layer features and weights.

16.3.5 k-Nearest Neighbors

The k-nearest neighbor algorithm (kNN), also known as the lazy learning algorithm, is one of the simplest nonparametric machine learning algorithms which is based on instance-based learning. The algorithm takes as input the training instances and assigns a test instance the class voted by the majority of its closest neighbors, i.e., where k is a positive integer. Mostly, the value of k is kept small if $k = 1$ the algorithm will assign the instance same class as of its nearest neighbor (Mitchell 2014). The training instances lie as position vectors in the feature space and the distance between the training instances and the query is calculated. Euclidean distance matrix is generally used to calculate the distances. To increase the accuracy of the classification, weights can be added to the closest neighbors so that they contribute more toward the classification. The effectiveness of the classifier depends on the value of k; it is preferred to choose an odd value for k in the binary classification problems (Altman 1992).

16.4 Model Validation

16.4.1 Testing Set

A learned model system generated is only effective if it can make the prediction for the previously uncharacterized data which is known as the testing set. The model systems are generated using the training data in which the class to which a particular instance belongs is already known. However, the model systems generated are validated to assess the performance of the classification algorithms using the testing data in which the outcome is not already known to the learned system. The test set is a set of instances that did not have any role during the learning of the model system.

16.4.2 Cross-Validation

To gain insights into the performance of the learned system on previously unknown data and to use the best parameter values for generating the model, cross-validation

technique is used. An internal assessment of the learned system is performed by breaking the training data into subsets which are known as validation sets. The validation sets are used for tuning of the parameters during the formation of the classifiers.

16.4.2.1 N-Fold Cross-Validation

In n-fold cross-validation, the training data is divided into N equally sized folds, and each time during the learning process, N-1 folds are employed for training, and the onefold left is used as test set. This procedure is repeated N times until every fold has been used as the test set at least once following which the average performance over all the folds is taken to produce a single output. Usually, five- or tenfold cross-validation is used depending upon the dimensions of the training set.

16.4.2.2 Leave-One-Out (LOO) Cross-Validation

In LOO cross-validation, if the training data is comprised of K instances, K-1 instances are used to generate the model, and the remaining one instance is used as the testing set. K-learned systems are obtained at the end among which either any of the K model is used as final learned system or a newly learned system can be generated on the whole data using the best parameter values selected by cross-validation. The LOO form of cross-validation makes comprehensive use of the data and thus is computationally very expensive.

16.4.3 Evaluating Classifier Performance

A variety of statistical figures have been suggested to test the predictive ability of the learned model system. A learned system is not considered as an accurate system if it produces an error on training data predictions. In case of binary classification problems, the instances are divided into true positives, TP (positive prediction); true negatives, TN (negative prediction); false positives, FP (negative predicted as positive); and false negatives, FN (positive predicted as negative). There are various metrics used which include true positive rate or sensitivity or recall, (TP/(TP + FN)), which is the proportion of the positive predictions. True negative rate or specificity, (TN/(TN + FP)), is the percentage of negative predictions identified as negative. Another very popularly used metric is precision (TP/(TP + FP)) also referred to as positive prediction value, which is the fraction of positive predictions which are actual positives. Accuracy (TP + TN/(TP + TN + FP + FN)) is the percentage of the correct positive and negative predictions. A good model system is one which gives highly accurate prediction on training data; however, accuracy alone cannot be used as a measure for classification tasks as it will always predict the majority class for all the instances. A balanced measure is required to overcome the accuracy paradox. F-measure or F-balanced score, ($2 \times$ PrecisionxRecall/Precision + Recall), is the harmonic mean between precision and recall which is used to evaluate the accuracy of the model systems.

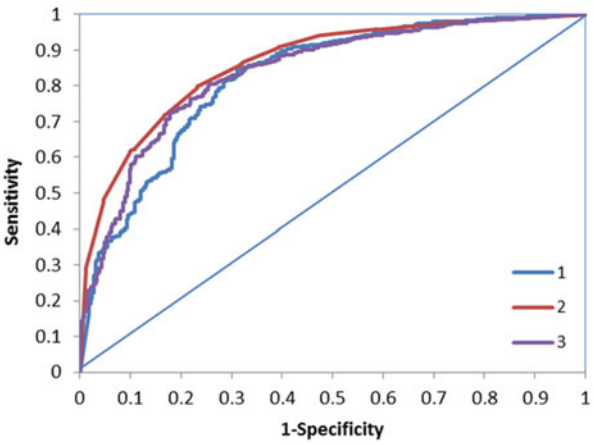

Fig. 16.1 A ROC plot generated using Weka

The performance of the learned systems can also be visualized by computing curve between sensitivity and 1-specificity; the plot is known as receiver operating characteristic (ROC; Fig. 16.1). The area under curve (AUC) value can be computed from the ROC plot which gives information about the performance of the learned systems. The value of AUC lies between 0 and 1 which is the best possible value. A range of other useful evaluation metrics have also been proposed which can be used depending on the requirement of the classification task (Demsar 2006).

16.5 Machine Learning Software

The following software suites are the implementations of various machine learning algorithms:

16.5.1 Open Source Software

- dlib, a C++ based machine learning library
- OpenNN, a C++ implementation of neural networks
- Torch, LuaJIT-based computing framework for machine learning algorithms
- ELKI (for Environment for Developing KDD-Applications Supported by Index-Structures), a Java-based platform for knowledge discovery in databases
- Orange, C++ and Python-based machine learning suite
- Scikit-learn, largely Python-based machine learning library
- R, a programming language that implements a range of techniques, one among which is m machine learning
- Weka (Waikato Environment for Knowledge Analysis), a very popular Java-based suite of machine learning techniques

There is a wide range of other open source software suites including Apache's Spark, Intel's OpenCV (Open Source Computer Vision), Encog, and Shogun.

16.5.2 Commercial Software

- Amazon machine learning, machine learning platform offered by Amazon.
- KXEN modeler.
- Neural designer developed by Intelnics.
- Mathematica, written in Wolfram language.
- STATISTICA Data Miner developed by StatSoft.
- MATLAB (matrix laboratory) is a programming language developed by MathWorks that allows implementation of machine learning algorithms.

Other commercial software for machine learning includes Microsoft Azure, RCASE, SAS Enterprise Miner, IBM SPSS Modeler, and NeuroSolutions.

16.6 A Case Study Using Weka Machine Learning Platform

Weka is one of the most popularly used free accessible machine learning suite developed by University of Waikato, New Zealand (Fig. 16.2). The suite consists of tools for preprocessing of data, classification, clustering, regression, and visualization.

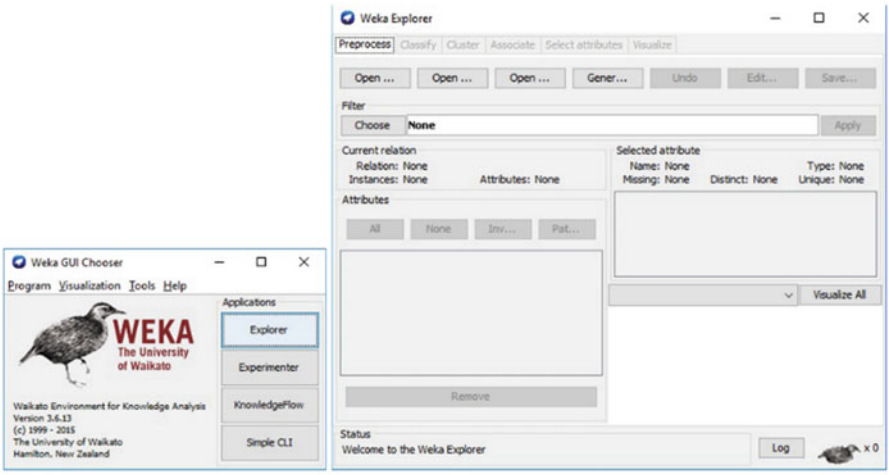

Fig. 16.2 Weka: (**a**) GUI chooser indicating the explorer interface, (**b**) explorer interface

16.6.1 Predicting Activity Outcome for Chemical Compounds

The goal of the present study was to generate machine learning-based predictive model which can find bioactive compounds from a high-throughput bioassay dataset consisting of the active and inactive compounds.

16.6.1.1 Dataset Description

The high-throughput bioassay dataset was downloaded from the PubChem database maintained by National Center of Biotechnology Information (NCBI). The bioassay was conducted to identify inhibitors and substrates of cytochrome P450 2D6. The dataset consisted of 1623 active and 6338 inactive compounds.

16.6.1.2 Data Preparation

The attributes or features for the compounds were generated using the descriptor generation software, PowerMV. A total of 179 attributes were generated, and the problematic attributes were removed. The dimensionality of the dataset was reduced by removing the attributes having identical values throughout the dataset, using the RemoveUseless filter of Weka. Figure 16.3 shows reading in the compounds data and choosing RemoveUseless filter.

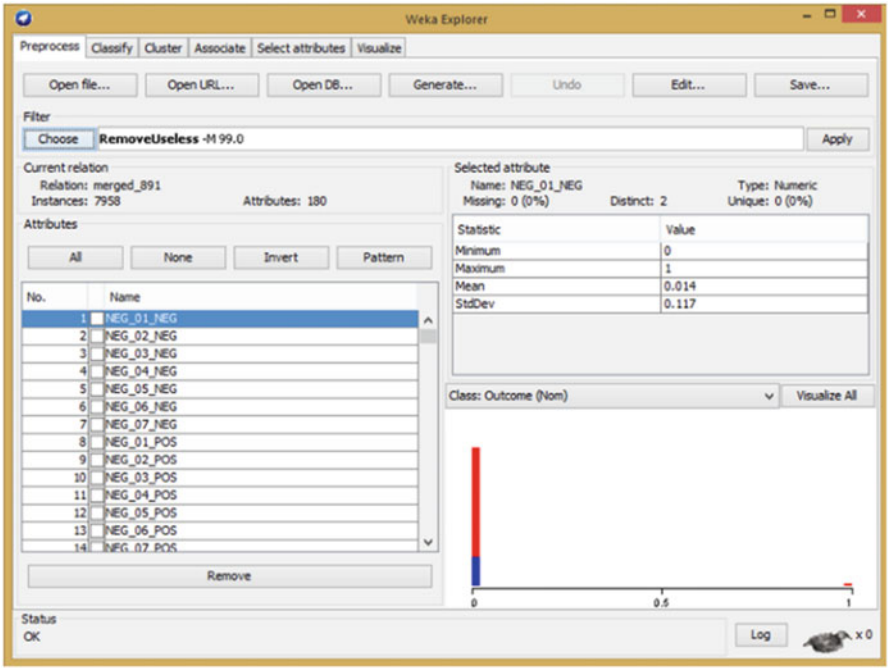

Fig. 16.3 Reading in the compounds data and choosing RemoveUseless filter

16.6.1.3 Training and Testing Set

The resultant significant attributes were saved in CSV (comma space value) format, and the data was divided into 80% training set which was to train the model and 20% test set which was used to assess the performance of the generated model.

16.6.1.4 Model Generation Using Different Learning Techniques

The train and test files were converted to ARFF (Attribute-Relation File Format) using Weka, and the different machine learning algorithms were used to generate the predictive models using the training set. The machine learning algorithms can be used by going to "Classify" tab in Weka, and the different algorithms can be chosen under the Classifier category (Fig. 16.4). The number of folds for cross-validation can also be specified in the box placed under "Cross-validation."

Once the model is generated, its performance can be evaluated using the testing set which can be supplied using "Supplied test set" option available in "Classify" tab of Weka (Fig. 16.5).

The performance of the model can be improved by changing the machine learning algorithm used and altering the parameters of the algorithm.

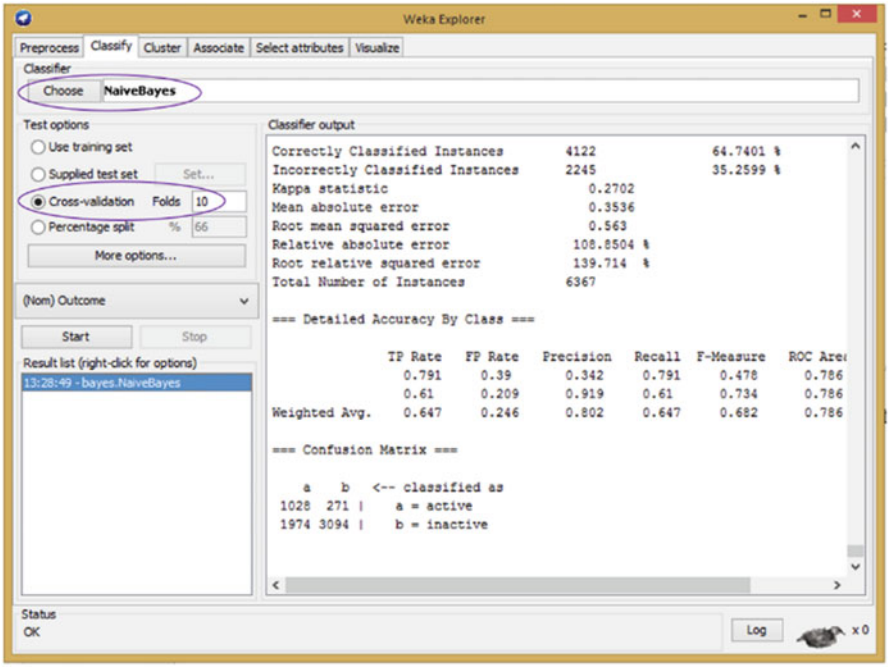

Fig. 16.4 Weka classify tab, model generation using Naïve Bayes classifier

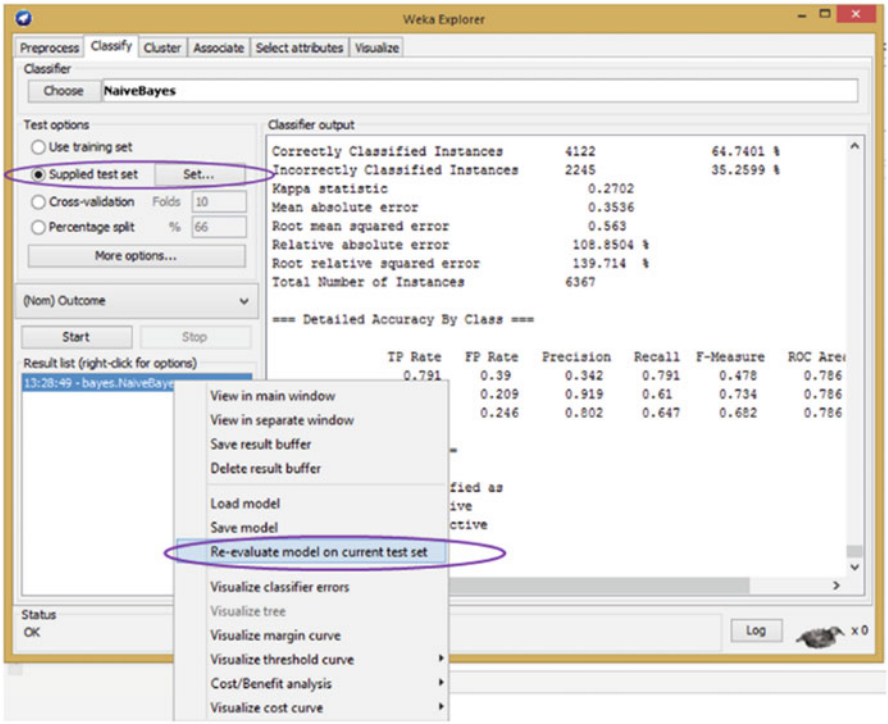

Fig. 16.5 Supply test set to Weka for evaluation of the generated learning model

16.6.1.5 Statistical Assessment of the Generated Models

The suitable metrics can be computed for the data, and the values can be recorded for the components of the confusion matrix which takes account of TP, FP, TN, and FN (Fig. 16.6). The various statistical figures of merit which can be employed have already been discussed in Sect. 16.4.3.

The increasing amount of data generated in recent years and the growing curiosity in using this data to discover new facts and make better and improved decisions for the future has led to the development of various robust and effective machine learning algorithms discussed in this chapter. The types of learning method to be used depend on the nature of the data and can be employed to various applications of machine learning to generate learned model systems for prediction.

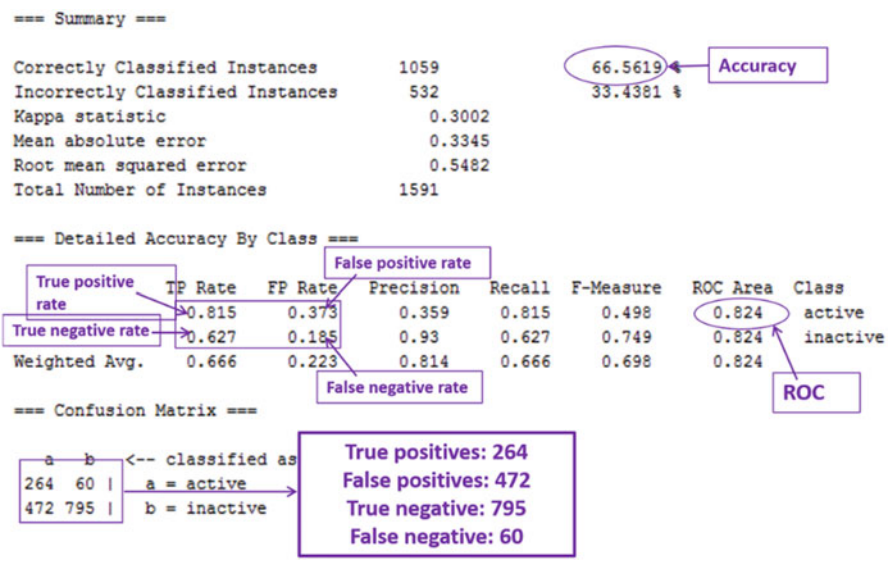

Fig. 16.6 Output obtained from the generated Naïve Bayes classifier

Acknowledgments Salma Jamal acknowledges a Senior Research Fellowship from the Indian Council of Medical Research (ICMR).

References

Alex S, Vishwanathan SVN (2008) Introduction to machine learning. Cambridge University Press, Cambridge

Ali J, Khan R, Ahmad N, Maqsood I (2012) Random forests and decision trees. Int J Comput Sci Issues 9(5). JOELib/JOELib2 cheminformatics library

Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185

Bishop CM (2006) Pattern recognition and machine learning. In: Information science and statistics. Springer, New York

Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans Pattern Anal Mach Intell 15(10):1042–1052

Chemical Computing Group Inc (2015) Molecular operating environment (MOE). 2013.08 edn., Sherbooke St. West, Suite #910, Montreal, QC, Canada

Cheng J, Tegge AN, Baldi P (2008) Machine learning methods for protein structure prediction. IEEE Rev Biomed Eng 1:41–49. https://doi.org/10.1109/RBME.2008.2008239

Christopher B (2006) Pattern recognition and machine learning. In: Information science and statistics. Springer, New York

Dang V, Croft WB (2010) Feature selection for document ranking using best first search and coordinate ascent. In: Proceedings of SIGIR workshop on feature generation and selection for information retrieval

Daumé H (2012) A course in machine learning. ciml.info

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Dervisevic I (2006) Machine learning methods for optical character recognition. pp 1–25
Ethem A (2009) Introduction to machine learning. The MIT Press, Cambridge
Farahat AK, Ghodsi A, Kamel MS (2011) An efficient greedy method for unsupervised feature selection. In: 11th IEEE international conference on data mining
Friedman N, Geiger D, GoldSzmidt M (1997) Bayesian network classifiers. Mach Learn 29:131–163
Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
Hsu C-W, Chang C-C, Lin C-J (2003) A practical guide to support vector classification. National Taiwan University
Jamal S, Goyal S, Shanker A, Grover A (2017) Computational screening and exploration of disease-associated genes in Alzheimer's disease. J Cell Biochem 118(6):1471–1479. https://doi.org/10.1002/jcb.25806
Karuppasamy S, Indradevi MR, Rajaram R (2008) Combined feature selection and classification – a novel approach for the categorization of web pages. J Inf Comput Sci 3(2):083–089
Kohavi R, Provost F (1998) Glossary of terms. Mach Learn 30:271–274
Liu K, Feng J, Young SS (2005) PowerMV: a software environment for molecular viewing, descriptor generation, data analysis and hit evaluation. J Chem Inf Model 45(2):515–522. https://doi.org/10.1021/ci049847v
López FG, Torres MG, Batista BM, JAM P, Moreno-Vega JM (2006) Solving feature subset selection problem by a parallel scatter search. Eur J Oper Res 169(2):477–489
Mitchell TM (1997) Machine learning. McGraw-Hill Science/Engineering/Math, Maidenhead
Mitchell JB (2014) Machine learning methods in chemoinformatics. Wiley Interdiscip Rev Comput Mol Sci 4(5):468–481. https://doi.org/10.1002/wcms.1183
Mohri M, Rostamizadeh A, Talwalkar A (2012) Foundations of machine learning. The MIT Press, Cambridge (MA)/London
Moore CL, Smagala JA, Smith CB, Dawson ED, Cox NJ, Kuchta RD Rowlen KL (2007) Evaluation of MChip with historic subtype H1N1 influenza A viruses, including the 1918 "Spanish Flu" strain. J Clin Microbiol 45 (11):3807-3810. JCM.01089-07 [pii]https://doi.org/10.1128/JCM.01089-07
Murphy KP (2012) Machine learning: a probabilistic perspective. MIT Press, Cambridge
Platt JC (1998) Sequential minimal optimization: a fast algorithm for training support vector machines. Microsoft Research
Sajda P (2006) Machine learning for detection and diagnosis of disease. Annu Rev Biomed Eng 8:537–565. https://doi.org/10.1146/annurev.bioeng.8.061505.095802
Simon P (2013) Too big to ignore: the business case for big data. Wiley, Hoboken
Singh H, Kumar R, Singh S, Chaudhary K, Gautam A Raghava GP (2016) Prediction of anticancer molecules using hybrid model developed on molecules screened against NCI-60 cancer cell lines. BMC Cancer 16:77. https://doi.org/10.1186/s12885-016-2082-y10.1186/s12885-016-2082-y [pii]
Stuart R, Peter N (2003) Artificial intelligence: a modern approach, 2nd edn. Prentice Hall, Upper Saddle River
Sutton R, Barto A (1998) Reinforcement learning: an introduction. MIT Press, Cambridge, MA
Tiwari R, Singh MP (2010) Correlation-based attribute selection using genetic algorithm. Int J Comput Appl 4(8):0975–8887
Tretyakov K (2004) Machine learning techniques in spam filtering. Institute of Computer Science, University of Tartu
Valla A, Giraud M, Dore JC (1993) Descriptive modeling of the chemical structure-biological activity relations of a group of malonic polyethylenic acids as shown by different pharmacotoxicologic tests. Pharmazie 48(4):295–301
Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474. https://doi.org/10.1002/jcc.21707
Yong SL, Hagenbuchner M, Tsoi AC (2008) Ranking web pages using machine learning approaches. Web Intelligence and Intelligent Agent Technology, 2008 WI-IAT '08 IEEE/WIC/ACM International Conference 3:677–680