# PREDICTING STUDENTS'PERFORMANCE IN DISTANCE LEARNING USING MACHINE LEARNING TECHNIQUES

**Article** *in* Applied Artificial Intelligence · January 2004

Source: DBLP

CITATIONS
23

READS
1,130

**3 authors:**

Sotiris Kotsiantis
University of Patras
**191** PUBLICATIONS **5,683** CITATIONS

SEE PROFILE

Christos Pierrakeas
Hellenic Open University
**48** PUBLICATIONS **775** CITATIONS

SEE PROFILE

P. E. Pintelas
University of Patras
**158** PUBLICATIONS **2,924** CITATIONS

SEE PROFILE

Applied Artificial Intelligence, 18:411–426, 2004

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510490442058

Taylor & Francis
Taylor & Francis Group

# PREDICTING STUDENTS' PERFORMANCE IN DISTANCE LEARNING USING MACHINE LEARNING TECHNIQUES

S. KOTSIANTIS
Department of Mathematics, University of Patras, Patras, Greece

C. PIERRAKEAS
School of Science & Technology, Hellenic Open University, Patras, Greece

P. PINTELAS
Department of Mathematics, University of Patras, Patras, Greece

*The ability to predict a student's performance could be useful in a great number of different ways associated with university-level distance learning. Students' key demographic characteristics and their marks on a few written assignments can constitute the training set for a supervised machine learning algorithm. The learning algorithm could then be able to predict the performance of new students, thus becoming a useful tool for identifying predicted poor performers. The scope of this work is to compare some of the state of the art learning algorithms. Two experiments have been conducted with six algorithms, which were trained using data sets provided by the Hellenic Open University. Among other significant conclusions, it was found that the Naïve Bayes algorithm is the most appropriate to be used for the construction of a software support tool, has more than satisfactory accuracy, its overall sensitivity is extremely satisfactory, and is the easiest algorithm to implement.*

Computers do not learn as well as people do, but many machine-learning algorithms have been found that are effective for some types of learning tasks. They are especially useful in poorly understood domains where humans might not have the knowledge needed to develop effective knowledge-engineering algorithms. Generally, machine learning (ML) explores

Address correspondence to S. B. Kotsiantis, Educational Software Development Laboratory, Department of Mathematics, University of Patras, P.A. Box 1399, Patras 26500, Greece. E-mail: sotos@math.upatras.gr

algorithms that reason from externally supplied instances (input set) to produce general hypotheses, which will make predictions about future instances. The externally supplied instances are usually referred to as training set. To induce a hypothesis from a given training set, a learning system needs to make assumptions (*biases*) about the hypothesis to be learned. A learning system without any assumption cannot generate a useful hypothesis since the number of hypotheses that are consistent with the training set is usually huge. Since every inductive learning algorithm uses some biases, it behaves well in some domains where its biases are appropriate, while it performs poorly in other domains (Schaffer 1994).

This paper uses existing ML techniques in order to predict students' performance in a distance learning system. It compares some of the state of the art learning algorithms to find out which algorithm is more appropriate not only to predict student's performance accurately, but also to be used as an educational supporting tool for tutors. To the best of our knowledge, there is no similar publication in the literature. Whittington (1995) studied only the factors that impact on the success of distance education students of the University of the West Indies.

For the purpose of our study, the *informatics* course of the Hellenic Open University (HOU) provided the training set for the ML algorithms. The basic educational unit at HOU is the module and a student may register for up to three modules per year. The informatics course is composed of 12 modules and leads to a bachelor's degree. The total number of registered students with the course of informatics in the academic year 2000-1 was 510. Of those students, 498 (97.7%) selected the module *Introduction to Informatics* (INF10). This fact enabled the authors to focus on INF10 and collect data only from the tutors involved in this module.

The tutor in a distance-learning course has a specific role. Despite the distance between him/her and his/her students, he/she has to teach, evaluate, and continuously support them. The communication between them by post, telephone, e-mail, through the written assignments, or at optional consulting meetings helps the tutor to respond to this complex role (Baath 1994; Narasimharao 1999). In all circumstances, the tutor should promptly solve students' educational problems, discuss in a friendly way the issues that distract them, instruct their study, and, most of all, encourage them to continue their studies, understanding their difficulties, and effectively supporting them. Furthermore, tutors have to give them marks, comments, and advice on the written assignments and they have to organize and carry out the face-to-face consulting meetings.

For all the above mentioned reasons, it is important for the tutors to be able to recognize and locate students with a high probability of poor performance (students at risk) in order to take precautions and be better prepared to face such cases.

Regarding the INF10 module of HOU, during an academic year, students have to hand in four written assignments, participate in four optional face-to-face meetings with their tutor, and sit for final examinations after an 11-month period. The students' marking system in Hellenic Universities is the 10-grade system. A student with a mark >= 5 "passes" a lesson or a module, while a student with a mark < 5 "fails" to complete a lesson or a module.

Key demographic characteristics of students (such as age, sex, residence, etc.) and their marks in written assignments constituted the initial training set for a supervised learning algorithm in order to predict if a certain student will eventually pass or not pass a specific module. A total of 354 instances (students' records) have been collected out of the 498 who had registered for INF10 (Xenos et al. 2002).

Two separate experiments were conducted. The first experiment used the entire set of 354 instances for all algorithms, while the second experiment used only a small set of 28 instances, corresponding to the number of students in a tutor's class.

The application of machine learning techniques in predicting students' performance proved to be useful for identifying poor performers and it can enable tutors to take remedial measures at an earlier stage, even from the beginning of an academic year using only students' demographic data, in order to provide additional help to the groups at risk. The probability of more accurate diagnosis of students' performance is increased as new curriculum data has entered during the academic year, offering the tutors more effective results.

## MACHINE LEARNING ISSUES

Inductive machine learning is the process of learning from examples (instances), a set of rules, or more generally speaking a concept or a classifier that can be used to generalize to new examples. Inductive learning can be loosely defined for a two-class problem as the following. Let c be any Boolean target concept that is being searched for. Given a classifier L and a set of instances X for which c is defined over, train L on X to estimate c. The instances X, which L is trained on, are known as training examples and are made up of ordered pairs $< x, c(x) >$, where x is a vector of attributes (which have values), and c(x) is the associated classification of the vector x. L's approximation of c is its hypothesis h. In an ideal situation, after training L on X, h equals c, but in reality a classifier can only guarantee a hypothesis h, such that it fits the training data. Without any other information, we assume that the hypothesis, which fits the target concept on the training data, will also fit the target concept on unseen examples (Mitchell 1997).

In Table 1, a confusion matrix is presented, which shows the type of classification errors a classifier can make for the two-class case. Thus, the

**TABLE 1** A Confusion Matrix

| Hypothesis (prediction) | | |
|---|---|---|
| + | − | Actual Class |
| *a* | *b* | + |
| *c* | *d* | − |

breakdown of a confusion matrix is as follows: *a* is the number of positive instances correctly classified, *b* is the number of positive instances misclassified as negative, *c* is the number of negative instances misclassified as positive, and *d* is the number of negative instances correctly classified.

Actually, the most well-known classifier criterion is its prediction accuracy. The prediction accuracy (denoted as *acc*) is commonly defined over all the classification errors that are made and it is calculated as:

$$acc = (a + d)/(a + b + c + d).$$

In machine learning techniques, classification speed is also in many cases a crucial property that is demanded by the classifier. This efficiency criterion is less often considered, but arises from the requirement that a classifier should use only reasonable amounts of time and memory for training and application (Gaga 1996).

In order to predict student's performance, the application of six of the most common machine learning techniques, namely Decision Trees (Murthy 1998), Neural Networks (Mitchell 1997), Naïve Bayes algorithm (Domingos and Pazzani (1997), Instance-Based Learning algorithms (Aha 1997), Logistic Regression (Long 1997), and Support Vector Machines (Burges 1998) are used. In the next sub-section we will briefly describe these supervised machine learning techniques. A detailed description can be found in Kotsiantis et al. (2002b).

## Brief Description of the Used Machine Learning Techniques

Murthy (1998) provides a recent overview of existing work in decision trees. Decision trees are trees that classify instances by sorting them based on attribute values. Each node in a decision tree represents an attribute in an instance to be classified, and each branch represents a value the node can take. Instances are classified starting at the root mode and sorting them based on their attribute values. The main advantage of decision trees in particular and hierarchical methods in general, is that they divide the classification problem into a sequence of sub problems which are, in principle, simpler to solve than the original problem. The attribute that best divides the training data would be the root node of the tree. The algorithm is then repeated on each partition of the divided data, creating subtrees until the training data are divided into subsets of the same class.

Artificial neural networks (ANNs) are another method of inductive learning based on computational models of biological neurons and networks of neurons as found in the central nervous system of humans (Mitchell 1997). A multi-layer neural network consists of large number of units (neurons) joined together in a pattern of connections. Units in a net are usually segregated into three classes: input units, which receive information to be processed, output units, where the results of the processing are found, and units in between called hidden units. Classification with a neural network takes place in two distinct phases. First, the network is trained on a set of paired data to determine the input-output mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data.

Naïve Bayes classifier is the simplest form of Bayesian network (Domingos and Pazzani 1997). This algorithm captures the assumption that every attribute is independent from the rest of the attributes, given the state of the class attribute. Naïve Bayes classifiers operate on data sets where each example $x$ consists of attribute values $\langle a_1, a_2 \ldots a_i \rangle$ and the target function f(x) can take on any value from a predefined finite set $V = (v_1, v_2 \ldots v_j)$. The formula used by the Naïve Bayes classifier is:

$$v_{\max} = \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

where $v$ is the target output of the classifier and $P(a_i|v_j)$ and $P(v_i)$ can be calculated based on their frequency in the training data.

Instance-based learning algorithms belong in the category of lazy-learning algorithms (Mitchell 1997), as they defer in the induction or generalization process until classification is performed. One of the most straightforward instance-based learning algorithms is the nearest neighbor algorithm (Aha 1997). K-nearest neighbor (kNN) is based on the principal that the instances within a data set will generally exist in close proximity with other instances that have similar properties. If the instances are tagged with a classification label, then the value of the label of an unclassified instance can be determined by observing the class of its nearest neighbors.

Logistic regression analysis (Long 1997) extends the techniques of multiple regression analysis to research situations in which the outcome variable (class) is categorical. The relationship between the classifier and attributes is not a linear function; instead, the logistic regression function is used, which is the logit transformation of $p_i$.

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} = \ln\left(\frac{\text{Prob}(y_i = 1)}{\text{Prob}(y_i = 0)}\right).$$

The dependent variable (class) in logistic regression is binary, that is, the dependent variable can take the value 1 with a probability of success $p_i$, or the value 0 with probability of failure $1 - p_i$. Comparing these two probabilities, the larger probability indicates the class label value that is more likely to be the actual label.

The SVM technique revolves around the notion of a *margin*, either side of a hyperplane that separates two data classes. Maximizing the margin, and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it, is proven to reduce an upper bound on the expected generalization error (Burges 1998). Nevertheless, most real-world problems involve non-separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. One solution to the inseparability problem is to map the data into a higher-dimensional space and define a separating hyperplane there. This higher-dimensional space is called the feature space, as opposed to the input space occupied by the training instances. Generally, with an appropriately chosen feature space of sufficient dimensionality, any consistent training set can be made separable.

## DATA DESCRIPTION AND RESEARCH DESIGN

Data were collected from two distinct sources, the students' registry of the HOU and the records of the tutors. This enabled the authors to collect data concerning almost all students. With regard to the data collected from the student registry of HOU, a common feature of most faculties of science and technologies was confirmed: the low percentage of female students, which is a phenomenon that characterizes this course of informatics as well. The male-female ratio in the total of 510 students for this course, for the academic year 2000-1, was 72%/28%. As anticipated, the majority of students selected only one module (INF10 module), fewer selected two, and even less selected all three offered modules.

### Data Description and Attribute Selection

According to the data collected in the framework of this research, the students' age follows a normal distribution with an average value of 31.1 years ($\pm 5.1$). It must be noted that no students under the age of 24 years can be accepted, according to the regulation of HOU, since it is considered that such students could easily attend conventional Hellenic universities.

A student must submit at least three assignments (out of four). The tutors evaluate these assignments and a mark greater or equal to 20 should be obtained in total in order for each student to successfully complete the INF10 module. Students who meet the above criteria also have to sit for the final examination test.

The variables (attributes) used are presented in Table 2, along with the values of every attribute.

The set of the attributes was divided in three groups. The "Registry Class," the "Tutor Class," and the "Classroom Class." The "Registry Class" represents attributes which were collected from the student's registry of the HOU concerning students' sex, age, marital status, number of children, and occupation where overtime is the group of students working in more than one job.

In addition to the above attributes, the previous—post high school—education in the field of informatics and the association between students' jobs and computers were also taken into consideration. A student having attended at least a seminar (of 100 hours or more) on informatics after high school would qualify as "yes" in computer literacy. Furthermore, a student who uses software packages (such as word processor) at his job without having any deep knowledge in informatics was considered as "junior-user." A student who works as a programmer or in data processing departments was considered a "senior user." The remaining students' jobs were listed as "no" concerning association with computers.

"Tutor Class" represents attributes, which were collected from tutors' records concerning students' marks on the written assignments and their presence or absence in face-to-face meetings. Marks in the written assignments were categorized in five groups where "no" means no submission of the specific assignment, "fail" means a mark less that 5, "good" means a mark between 5 and 6.5, "very good" means a mark between 6.5 and 8.5, and "excellent" means a mark higher than 8.5.

**TABLE 2** The Attributes Used and Their Values

| | |
|---|---|
| Student's Registry (demographic) attributes | |
| Sex | male, female |
| Age | 24–46 |
| Marital status | single, married, divorced, widowed |
| Number of children | none, one, two, three, four of more |
| Occupation | no, part-time, fulltime, over-time |
| Computer literacy | no, yes |
| Job associated with computers | no, junior-user, senior-user |
| Attributes from tutors' records | |
| 1st face to face meeting | absent, present |
| 1st written assignment | no, fail, good, very good, excellent |
| 2nd face to face meeting | absent, present |
| 2nd written assignment | no, fail, good, very good, excellent |
| 3rd face to face meeting | absent, present |
| 3rd written assignment | no, fail, good, very good, excellent |
| 4th face to face meeting | absent, present |
| 4th written assignment | no, fail, good, very good, excellent |
| Class | |
| Final examination test | fail, pass |

Finally, the "class attribute" (dependent variable) represents the result on the final examination test with two values. "Fail" represents students with poor performance. "Poor performance" indicated students who suspended their studies during the academic year (due to personal or professional reason or inability to hand in two of the written assignments), students who did not participate in the final examination, or students who did sit for the final examination but got a mark less than 5. "Pass" represents students who completed the INF10 module getting a mark of 5 or more in the final test.

***Algorithm Selection***

For the purpose of the present study, a representative algorithm for each machine learning technique described earlier was selected. The most commonly used C4.5 algorithm (Quinlan 1993) was the representative of the decision trees in our study. The most well-known learning algorithm to estimate the values of the weights of a neural network—the Back Propagation (BP) algorithm (Mitchell 1997)—was the representative of the ANNs. The Naïve Bayes algorithm in our case is based on estimating:

$$R = \frac{P(FinalTest = pass|X)}{P(FinalTest = fail|X)} = \frac{P(FinalTest = pass)P(X|FinalTest = pass)}{P(FinalTest = fail)P(X|FinalTest = fail)} \Rightarrow$$

$R = P(FinalTest = pass)\prod P(X_r|FinalTest = pass)/P(FinalTest = fail)\prod P(X_r|FinalTest = fail)$, where $X_r$ are the attributes from Table 2. In our study, we also used the 3-NN algorithm that combines robustness to noise and less time for classification than using a larger k for kNN (Wettschereck et al. 1997). MLE (Maximum Likelihood Estimation) is the statistical method for estimating the coefficients of the logistic model (Long 1997). Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs in our study because it is one of the fastest methods to train SVMs (Platt 1999).

Detailed descriptions of all these algorithms can be found in Kotsiantis et al. (2002a). It must be also mentioned that we used the free available source code for these algorithms by Witten and Frank (2000) for our experiments.

## Research Design

In order to rank the representative algorithms of the machine learning techniques that are used in this study, three basic criteria are used. These would be prediction accuracy, sensitivity, and specificity. The sensitivity of the algorithm measures how good an algorithm is in classifying correctly positive instances and is defined as the ratio:

$$\text{sen} = \text{a}/(\text{a} + \text{b})$$

where, $a$ is the number of positive instances correctly classified and $b$ is the number of positive instances misclassified as negative, specifying the accuracy in predicting students who will finally "pass" the module. The specificity of the algorithm measures how good an algorithm is in classifying correctly negative instances and is defined as the ratio:

$$\text{spe} = \text{d}/(\text{c} + \text{d})$$

where $c$ is the number of negative instances misclassified as positive and $d$ is the number of negative instances correctly classified, specifying the accuracy in predicting students who will finally "fail" the module.

Two separate experiments were conducted based on the attributes described earlier. The first experiment used all 354 instances for the training of every algorithm. The second experiment took place with fewer instances. The need for the second experiment is obvious because it is very difficult for a tutor to collect more than 30 instances per academic year. Therefore, the second experiment took place with only 28 instances as a training set.

During the first phase (training phase), every algorithm was trained using the data collected from the academic year 2000-1. The training phase was divided into nine consecutive steps. The first step included the demographic data and the resulting class (pass or fail); the second step included both the demographic data along with the data from the first face-to-face meeting and the resulting class. The third step included data used for the second step and the data from the first written assignment. The fourth step included data used for the third step and the data from the second face-to-face meeting and so on until the ninth step that included all attributes described in Table 2. The nine-step technique during the training phase described above was used for both experiments.

Subsequently, ten groups of data for the new academic year (2001-2) were collected from ten tutors and the corresponding data from the HOU registry. Each one of these ten groups was used to measure the prediction accuracy within these groups (testing phase).

The testing phase also took place in nine steps. During the first step, the demographic data of the new academic year were used to predict the class (pass or fail) of each student. This step was repeated ten times (for every tutor's data) and the average prediction accuracy is denoted in the row labeled "DEMOGR" in Table 3 for each algorithm. During the second step, these demographic data along with the data from the first face-to-face meeting were used in order to predict the class of each student. This step was also repeated ten times and the average prediction accuracy is denoted in the row labeled "FTOF-1" in Table 3 for each algorithm. During the third step, the data of the second step along with the data from the first written assignment were used in order to predict the class and the average prediction accuracy is denoted in the row labeled "WRI-1" in Table 3.

**TABLE 3** Accuracy of Algorithms in the First Experiment

| | Naïve Bayes | C4.5 | BP | SMO | 3-NN | Logistic |
|---|---|---|---|---|---|---|
| DEMOGR | 62.95% | 61.65% | 61.85% | 64.47% | 58.84% | 61.38% |
| FTOF-1 | 62.72% | 61.56% | 61.14% | 64.47% | 59.12% | 61.56% |
| WRI-1 | 66.23% | 65.35% | 63.62% | 63.11% | 60.21% | 65.32% |
| FTOF-2 | 69.78% | 62.93% | 68.18% | 68.58% | 62.41% | 69.40% |
| WRI-2 | 75.36% | 74.16% | 75.55% | 75.99% | 68.45% | 75.88% |
| FTOF-3 | 75.38% | 72.44% | 76.38% | 76.22% | 68.78% | 76.02% |
| WRI-3 | 78.58% | 79.22% | 80.11% | 77.71% | 72.62% | 79.20% |
| FTOF-4 | 79.20% | 74.84% | 78.02% | 78.37% | 75.14% | 80.14% |
| WRI-4 | 82.14% | 77.80% | 82.14% | 80.68% | 76.77% | 82.01% |

The remaining steps use data of the new academic year in the same way as described above. These steps are also repeated ten times and the average prediction accuracy is denoted in the rows labeled "FTOF-2," "WRI-2," "FTOF-3," "WRI-3," "FTOF-4," and "WRI-4," concurrently in Table 3 for each algorithm.

The nine-step technique during testing phase described above was used for the second experiment too (see Table 4).

## EXPERIMENT RESULTS

In this section, the results of the testing of each algorithm with our data set are presented. A more detailed description can be found in Kotsiantis et al. (2002a). In Table 3, the average prediction accuracy of each algorithm for all the testing steps of the first experiment is presented.

In Table 4, the average prediction accuracy of each algorithm for all the testing steps of the second experiment is presented.

The main statistical tests used to compare algorithms were one-way within-subjects (repeated measures) analysis of variance test (ANOVA) followed, whenever needed, by Tukey-test post-hoc analysis (Siegel and Castellan 1988). The resulting differences between classifiers were assumed

**TABLE 4** Accuracy of Algorithms in the Second Experiment

| | Naïve Bayes | C4.5 | BP | SMO | 3-NN | Logistic |
|---|---|---|---|---|---|---|
| DEMOGR | 56.29% | 57.73% | 53.37% | 54.97% | 55.67% | 55.64% |
| FTOF-1 | 56.47% | 57.42% | 50.71% | 55.28% | 55.95% | 55.21% |
| WRI-1 | 57.58% | 57.58% | 51.56% | 58.87% | 55.77% | 56.32% |
| FTOF-2 | 61.20% | 58.99% | 50.46% | 60.25% | 59.33% | 58.01% |
| WRI-2 | 66.75% | 62.21% | 59.69% | 65.08% | 64.08% | 61.32% |
| FTOF-3 | 69.60% | 62.76% | 62.94% | 67.24% | 67.39% | 61.13% |
| WRI-3 | 74.69% | 68.99% | 71.50% | 73.53% | 71.75% | 66.72% |
| FTOF-4 | 76.29% | 71.20% | 73.87% | 73.65% | 73.31% | 66.90% |
| WRI-4 | 79.51% | 73.99% | 74.88% | 76.96% | 76.60% | 64.85% |

**TABLE 5** The Overall Results for the Criteria Used for All the Algorithms (1st Experiment)

| Algorithm | Acc | Sen | Spe |
|---|---|---|---|
| C4.5 | 69.99% | 73.89% | 66.44% |
| BP | 72.26% | 76.32% | 68.31% |
| Naïve Bayes | 72.48% | 78.00% | 67.37% |
| 3-NN | 66.93% | 71.49% | 62.00% |
| Logistic regression | 72.32% | 76.06% | 68.52% |
| SMO | 72.17% | 76.05% | 69.06% |
| ANOVA test results | $F = 204.23$, $p < 0.001$ | $F = 204.83$, $p < 0.001$ | $F = 148.64$, $p < 0.001$ |

statistically significant when $p < 0.05$. Otherwise, they were assumed not statistically significant (NS) (Siegel and Castellan 1988). In Table 5, the overall average values for the six algorithms of the criteria used (overall accuracy [Acc], overall sensitivity [Sen], and overall specificity [Spe] in the first experiment are presented.

In order to estimate the most appropriate algorithm, comparisons are made for the criteria used. There were statistically significant differences among the six algorithms for all the criteria used, as presented in Table 5 ($p < 0.001$, according to ANOVA test for all the criteria).

The results of the post-hoc analysis for the overall accuracy (Acc) shows that the best algorithm is the Naïve Bayes (72.48%), followed by the Logistic Regression (72.32%), the BP (72.26%), and the SMO (72.17%). There were no statistically significant differences among the above four. On the contrary, the C4.5 (69.99%) algorithm that follows was of statistically significant lower accuracy than all the above-mentioned algorithms ($p < 0.001$). Finally, the lowest accuracy was calculated for the 3-NN (66.93%) algorithm, which was statistically significant different from all the others ($p < 0.001$) (see Table 6).

The results of the post-hoc analysis for the overall sensitivity (Sen) shows that again the best algorithm is the Naïve Bayes (78.00%), with statistically significant higher sensitivity than all the others ($p < 0.001$). The BP (76.32%), the Logistic Regression (76.06%), and the SMO (76.05%) algorithms follow with no statistically significant differences among them.

**TABLE 6** The Post-Hoc Results for the Overall Accuracy (1st Experiment)

| Algorithm | | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|---|
| C4.5 | (69.99%) | | | | | |
| BP | (72.26%) | $p < 0.001$ | | | | |
| Naïve Bayes | (72.48%) | $p < 0.001$ | NS | | | |
| 3-NN | (66.93%) | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | | |
| Logistic regression | (72.32%) | $p < 0.001$ | NS | NS | $p < 0.001$ | |
| SMO | (72.17%) | $p < 0.001$ | NS | NS | $p < 0.001$ | NS |

**TABLE 7** The Post-Hoc Results for the Overall Sensitivity (1st Experiment)

| Algorithm | | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|---|
| C4.5 | (73.89%) | | | | | |
| BP | (76.32%) | $p < 0.001$ | | | | |
| Naïve Bayes | (78.00%) | $p < 0.001$ | $p < 0.001$ | | | |
| 3-NN | (71.49%) | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | | |
| Logistic regression | (76.06%) | $p < 0.001$ | NS | $p < 0.001$ | $p < 0.001$ | |
| SMO | (76.05%) | $p < 0.001$ | NS | $p < 0.001$ | $p < 0.001$ | NS |

The C4.5 (73.89%) algorithm that follows was of statistically significant lower sensitivity than all the above-mentioned algorithms ($p < 0.001$) and the statistically significant ($p < 0.001$) lowest sensitivity was calculated for the 3-NN (71.49%) algorithm (see Table 7).

Slightly unlike the previous two criteria, the results of the post-hoc analysis for the overall specificity (Spe) show that the best algorithm is the SMO (69.06%), followed by the Logistic Regression (68.52%) and the BP (68.31%). There were no statistically significant differences among the above three. The Naïve Bayes algorithm follows (67.37%), with statistically significant lower specificity than the SMO ($p < 0.001$) and the Logistic Regression ($p < 0.01$). There was no statistically significant difference between the Naïve Bayes and the BP algorithms. The C4.5 (66.44%) algorithm was of statistically significant lower specificity than the above-mentioned four algorithms ($p < 0.001$), with the exception of the Naïve Bayes algorithm. Finally, the statistically significant lowest specificity calculated ($p < 0.001$) was that of 3-NN alogorithm (62.00%) (see Table 8).

Table 9 presents the overall average values for the six algorithms of the criteria used (overall accuracy [Acc], overall sensitivity [Sen], and overall specificity [Spe]) in the second experiment.

In order to estimate the most appropriate algorithm, comparisons are also made for the criteria used. There were statistically significant differences among the six algorithms for all the criteria used, as presented in Table 9 ($p < 0.001$, according to ANOVA test for all the criteria) in the second experiment as well.

**TABLE 8** The Post-Hoc Results for the Overall Specificity (1st Experiment)

| Algorithm | | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|---|
| C4.5 | (66.44%) | | | | | |
| BP | (68.31%) | $p < 0.001$ | | | | |
| Naïve Bayes | (67.37%) | NS | NS | | | |
| 3-NN | (62.00%) | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | | |
| Logistic regression | (68.52%) | $p < 0.001$ | NS | $p < 0.01$ | $p < 0.001$ | |
| SMO | (69.06%) | $p < 0.001$ | NS | $p < 0.001$ | $p < 0.001$ | NS |

**TABLE 9** The Overall Results for the Criteria Used for All the Algorithms (2nd Experiment)

| Algorithm | Acc | Sen | Spe |
|---|---|---|---|
| C4.5 | 63.43% | 69.79% | 57.67% |
| BP | 53.32% | 65.63% | 47.77% |
| Naïve Bayes | 66.49% | 72.55% | 60.58% |
| 3-NN | 64.43% | 70.77% | 58.54% |
| Logistic regression | 60.68% | 67.34% | 54.53% |
| SMO | 65.09% | 72.03% | 59.14% |
| ANOVA test results | $F = 71.86$<br>$p < 0.001$ | $F = 13.55$<br>$p < 0.001$ | $F = 68.15$<br>$p < 0.001$ |

The results of the post-hoc analysis for the overall accuracy (Acc) shows that the best algorithm is the Naïve Bayes (66.49%), followed by the SMO (65.09%) and the 3-NN (64.43%). There were no statistically significant differences among the above three. The C4.5 (63.43%) algorithm follows, with statistically significant lower accuracy than the Naïve Bayes algorithm ($p < 0.01$). There was no statistically significant difference between the C4.5 and the SMO algorithms or between the C4.5 and the 3-NN algorithms. The lowest accuracy was calculated for both the Logistic Regression (60.68%) and the BP (53.32%), which had statistically significant differences with all the others ($p < 0.001$) and between them ($p < 0.001$) (see Table 10).

The results of the post-hoc analysis for the overall sensitivity (Sen) shows that again the best algorithm is the Naïve Bayes (72.55%), followed by the SMO (72.03%), the 3-NN (70.77%), and the C4.5 (69.79%). There were no statistically significant differences among the above four. The Logistic Regression (67.34%) algorithm follows with statistically significant lower sensitivity than all the above-mentioned algorithms (Naïve Bayes [$p < 0.001$], SMO [$p < 0.001$], and 3-NN [$p < 0.05$]), except the C4.5. The lowest sensitivity, which was calculated, was that of the BP (65.63%) algorithm, which had statistically significant differences with all the others ($p < 0.001$), with the exception of the Logistic Regression (see Table 11).

**TABLE 10** The Post-Hoc Results for the Overall Accuracy (2nd Experiment)

| Algorithm | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|
| C4.5 (63.43%) | | | | | |
| BP(53.32%) | $p < 0.001$ | | | | |
| Naïve Bayes (66.49%) | $p < 0.01$ | $P < 0.001$ | | | |
| 3-NN (64.43%) | NS | $p < 0.001$ | NS | | |
| Logistic regression (60.68%) | $p < 0.01$ | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | |
| SMO (65.09%) | NS | $p < 0.001$ | NS | NS | $p < 0.001$ |

**TABLE 11** The Post-Hoc Results for the Overall Sensitivity (2nd Experiment)

| Algorithm | | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|---|
| C4.5 | (69.79%) | | | | | |
| BP | (65.63%) | $p < 0.001$ | | | | |
| Naïve Bayes | (72.55%) | NS | $p < 0.001$ | | | |
| 3-NN | (70.77%) | NS | $p < 0.001$ | NS | | |
| Logistic regression | (67.34%) | NS | NS | $p < 0.001$ | $p < 0.05$ | |
| SMO | (72.03%) | NS | $p < 0.001$ | NS | NS | $p < 0.001$ |

Finally, the results of the post-hoc analysis for the overall specificity (Spe) shows that once again the best algorithm is the Naïve Bayes (60.58%), followed by the SMO (59.14%) and the 3-NN (58.54%). There were no statistically significant differences among the above three. The C4.5 (57.67%) algorithm follows with statistically significant lower specificity than the Naïve Bayes ($p < 0.01$). There was no statistically significant difference between the C4.5 and the SMO algorithms or between the C4.5 and the 3-NN. On the other hand, the Logistic Regression (54.53%) algorithm gave statistically significant lower specificity than all the above-mentioned four algorithm ($p < 0.001$). The lowest specificity calculated was that of the BP (47.77%) algorithm, which had statistically significant differences with all the others ($p < 0.001$) (see Table 12).

To sum up, the accuracy of the 3-NN algorithm was poor. In addition, it requires a reasonable time for training for a lazy algorithm. Therefore, we conclude that it would not be appropriate to use it.

The best algorithm in terms of prediction has proven to be the Naïve Bayes one. This is primarily because the Naïve Bayes algorithm turned out to be much better than all the others in the second experiment for the criteria used. This is of highest importance since tutors can gather only few instances every academic year. In addition, it was proven that the same algorithm manifests the best prediction in overall accuracy as well as in overall sensitivity in the first experiment. It does not do equally well in the overall specificity of the first experiment. Finally, the required time for both training and testing is very little.

**TABLE 12** The Post-Hoc Results for the Overall Specificity (2nd Experiment)

| Algorithm | | C4.5 | BP | Naïve Bayes | 3-NN | Logistic regression |
|---|---|---|---|---|---|---|
| C4.5 | (57.67%) | | | | | |
| BP | (47.77%) | $p < 0.001$ | | | | |
| Naïve Bayes | (60.58%) | $p < 0.01$ | $p < 0.001$ | | | |
| 3-NN | (58.54%) | NS | $p < 0.001$ | NS | | |
| Logistic regression | (54.53%) | $p < 0.01$ | $p < 0.001$ | $p < 0.001$ | $p < 0.001$ | |
| SMO | (59.14%) | NS | $p < 0.001$ | NS | NS | $p < 0.001$ |

## CONCLUSION

This paper aims to fill the gap between empirical prediction of student performance and the existing ML techniques. To this end, six ML algorithms have been trained and found to be useful tools for identifying predicted poor performers in an open and distance learning environment. With the help of machine-learning methods, the tutors are in a position to know which of their students will complete a module or a course with sufficiently accurate precision. This precision reaches 62% in the initial forecasts, which are based on demographic data of the students and exceeds 82% before the final examinations. Our data set is from the module *Introduction in informatics* but most of the conclusions are wide-ranging and present interest for the majority of programs of study of the Hellenic Open University. It would be interesting to compare our results with those from other open and distance learning programs offered by other open universities. So far, however, we have not been able to locate such results.

Two experiments were conducted using data sets of 354 and 28 instances, respectively. The above accuracy was the result of the first experiment with the large data set; however, the overall accuracy of the second experiment for all algorithms was less satisfactory than the accuracy in the first experiment. The 28 instances are probably too few if we want more accurate precision. After a number of experiments with a different number of instances as the training set, it seems that at least 70 instances are needed for a better predictive accuracy (70.51% average prediction accuracy for the Naïve Bayes algorithm).

Besides the overall accuracy of the algorithms, the differences between sensitivity and specificity are quite reasonable since "pass" represents students who completed the INF10 module, getting a mark of 5 or more in the final test, while "fail" represents students who suspended their studies during the academic year (due to personal or professional reasons or due to inability to hand in two of the written assignments), as well as students who did not show up for the final examination. Furthermore, "fail" also represents students who sit for the final examination and get a mark less than 5.

Furthermore, the analysis of the experiments and the comparison of the six algorithms has demonstrated sufficient evidence that the Naïve Bayes algorithm is the most appropriate to be used for the construction of a software support tool. The overall accuracy of the Naïve Bayes algorithm was more than satisfactory (72.48% for the first experiment) and the overall sensitivity was extremely satisfactory (78.00% for the first experiment). Moreover, the Naïve Bayes algorithm is the easiest to implement among the tested algorithms.

In a future work, we intend to study if the use of more sophisticated approaches for discretization of the marks of WRIs, such as the one

suggested by Fayyad and Irani (1993), could increase the classification accuracy. In addition, because FTOFs did not add accuracy and the run time of inductive algorithms grows with the number of attributes, we will examine if the selection of a subset of attributes (Dash and Liu 1997) could be useful. Finally, since with the present work we can only predict if a student passes the module or not, we intend to try to use regression methods (Witten and Frank 2000) in order to predict the student's marks as well.

## REFERENCES


Aha, D. 1997. *Lazy Learning*. Dordrecht: Kluwer Academic Publishers.

Baath, J. 1994. Assignements in distance education – An Overview. *Epistolodidaktika* 1: 13–20.

Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2: 1–47.

Dash, M, and H. Liu. 1997. Feature selection for classification. *Intelligent Data Analysis* 1: 131–156.

Domingos, P., and M. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29: 103–130.

Fayyad, U., and K. Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027, Chambery, France, August 28–September 3.

Gaga, L. 1996. Id+: Enhancing medical knowledge acquisition with machine learning. *Applied Artificial Intelligence* 10: 79–94.

Kotsiantis, S., C. Pierrakeas, and P. Pintelas. 2002a. *Efficiency of Machine Learning Techniques in Predicting Students' Performance in Distance Learning Systems*, TR-02-03, Department of Mathematics, University of Patras, Hellas, Page 42.

Kotsiantis, S., I. Zaharakis, and P. Pintelas. 2002b. *Supervised Machine Learning*, TR-02-02, Department of Mathematics, University of Patras, Hellas, Page 28.

Long, J. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage.

Mitchell, T. 1997. *Machine Learning*. New York: McGraw Hill.

Murthy, S. 1998. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery* 2: 345–389.

Narasimharao, B. 1999. *Issues in Preparing Open University Learners for Open University System*. Available at the Web site http://www.cemca.org/ignou-icde/paper23.html.

Platt, J. 1999. Using sparseness and analytic QP to speed training of support vector machines. In *Advances in Neural Information Processing Systems*, (eds.) M. S. Kearns, S. A. Solla, and D. A. Cohn, 11. Cambridge, MA: The MIT Press.

Quinlan, J. R. 1993. C4.5: *Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann.

Schaffer, C. 1994. A conservation law for generalization performance. In *Proceedings of the Eleventh International Conference on Machine Learning*, Pages 153–178, New Brunswick, USA, July 10–13.

Siegel, S., and N.J. Castellan. 1988. Measures of association and their tests of significance. In *Nonparametric Statistics for the Behavioral Sciences*, 2nd edition. New York: McGraw-Hill.

Wettschereck, D., D. Aha, and T. Mohri. 1997. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence-Review* 10: 1–37.

Whittington, L.A. 1995. *Factors Impacting on the Success of Distance Education Students of the University of the West Indies: A Review of the Literature*, Review, University of West Indies.

Witten, I., and E. Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann.

Xenos, M., C. Pierrakeas, and P. Pintelas. 2002. A survey on student dropout rates and dropout causes concerning the students in the course of informatics of the Hellenic Open University, *Computers & Education* 39: 361–377.