A Hybrid Machine Learning Approach to Predict Learning Styles in Adaptive E-Learning System

Ouafae El Aissaoui[1](✉), Yasser El Madani El Alami[2], Lahcen Oughdir[1], and Youssouf El Allioui[3]

[1] LSI, FPT, University of Sidi Mohammed Ben Abdellah, Taza, Morocco
ouafae.elaissaoui@usmba.ac.ma
[2] ENSIAS, Mohammed V University, B.P.: 713, Agdal, Rabat, Morocco
[3] LS3M, FPK, Hassan I University, B.P.: 145, 25000 Khouribga, Morocco

Abstract. The increasing use of E-learning environments by learners makes it indispensable to implement adaptive e-learning systems (AeS). The AeS have to take into account the learners' learning styles to provide convenient contents and enhance the learning process. Learning styles refer to the preferred way in which an individual learns best. The traditional methods detecting learning styles (using questionnaires) present many limits, as: (1) the time-consuming process of filling in the questionnaire and (2) producing inaccurate results because students aren't always aware of their own learning preferences. Thus in this paper we have proposed an approach for detecting learning styles automatically, based on Felder and Silverman learning style model (FSLSM) and using machine learning algorithms. The proposed approach is composed of two parts: The first part aims to extract the learners' sequences from the log file, and then using an unsupervised algorithm (K-means) in order to group them into sixteen clusters according to the FSLSM, and the second part consists in using a supervised algorithm (Naive Bayes) to predict the learning style for a new sequence or a new learner. To perform our approach, we used a real dataset extracted from an e-learning system's log file. In order to evaluate the performance, we used the confusion matrix technique. The obtained results demonstrate that our approach yields excellent results.

Keywords: Adaptive E-Learning systems · Felder-Silverman learning style model · Unsupervised algorithm · Supervised algorithm · K-Means · Naïve bayes

1 Introduction

Adaptive E-learning refers to e-learning systems which provide customized contents for learners based on their profiles, with the final goal of enhancing their learning process. Learner profile is a representation of the learner's behavior while he/she is interacting with the system, the learner's behavior can be captured from the web logs using data mining techniques and then translated to a set of characteristics such as; skills, knowledge level, preferences, and learning style which is considered as an

M. Ezziyyani (Ed.): AI2SD 2018, AISC 915, pp. 772–786, 2019.
https://doi.org/10.1007/978-3-030-11928-7_70

essential factor that directly affects the student's learning process. Learning style is a vital learner's characteristic which must be taken into account in the learning personalization process, since it refers to the preferred way in which a learner perceives, treats and grasps the information. There are many learning style models where each learner is assigned to a learning style class based on the way he/she learns, a lot of LS models have been proposed such as [1–3], etc. The FSLSM is a popular learning style model which defines four dimensions (pre-processing, perception, input and understanding) and eight categories of learners (Active, reflective, sensing, intuitive, visual, verbal, Sequential and Global). In this work we have relied on the FSLSM for many reasons; firstly, because it is the most used in adaptive e-learning systems and the most appropriate to implement them [4, 5], one other reason is that the FSMSM enables the LS to be measured according to an Index of Learning Styles (ILS), The ILS consists of the four FSLSM's dimensions, each with 11 questions. Using the ILS we can link LS to appropriate learning objects.

There are many methods to detect a learning style, the traditional one consists in using questionnaires, applying this method can lead to time consuming and inaccurate results. Thus to acquire an efficient learning style, we have to detect it automatically from the log file which contains the learner's behavior using data mining techniques. In this paper we have proposed an approach which aims to detect the learners' learning styles dynamically based on Felder and Silverman [2] learning style model and using machine learning algorithms. In the first step of our approach we have extracted the learners' sequences from the log file, then we have used an unsupervised algorithm in order to group them into sixteen clusters where each cluster corresponds to a learning style, while in the second step, the clusters obtained from the first step have been considered as a training dataset which was used in order to perform a supervised algorithm on it and then predicting the learning style of a new sequence

This paper is organized as follows. Section 2 describes the learning style model and the algorithms used in our approach. Section 3 introduces a literature review of related work. Section 4 describes the methodology of our approach. The experiments and results are presented in Sect. 5; Finally, Sect. 6 presents our conclusions and future works.

2 Background

2.1 Learning Styles

Many definitions appear in the literature concerning the term learning style. Laschinger and Boss [1] defined learning style as the way in which individuals organize information and experiences, while Garity [6] defined it as the preferred way to learn and process information. Keefe [7] described learning styles as the cognitive, effective, and psychological behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment.

Learning style refers to the preferred way in which a learner perceives, reacts, interacts with, and responds to the learning environment. Each learner has his own preferred ways of learning; there are some students who prefer to study in a group, other prefer to study alone; some prefer to learn by reading written explanations, other by seeing visual representations, pictures, diagrams, and charts. In order to ensure an efficient learning process for learners; the e-learning system should consider the differences in learners' learning styles.

2.2 The Felder and Silverman Learning Style Model

A learning style model classifies the learners into a specific number of predefined dimensions, where each dimension pertains to the way they receive and process information. Many learning style models have been proposed in the literature, in this work we have based on the Felder and Silverman model.

According to the previous researches [4, 5], the FSLSM is the most used in adaptive e-learning systems and the most appropriate to implement them. The FSLSM presents four dimensions with two categories for each one, where each learner has a dominant preference for one category in each dimension: processing (active/reflective), perception (sensing/intuitive), input (visual/verbal), understanding (sequential/global).

Active (A) learners prefer to process information by interacting directly with the learning material, while **reflective (R)** learners prefer to think about the learning material. Active learners also tend to study in group, while the reflective learners prefer to work individually.

Sensing (Sen) learners tend to use materials that contains concrete facts and real world applications, they are realistic and like to use demonstrated procedure and physical experiments. While the **intuitive** learners prefer to use materials that contains abstract and theoretical information, they tend to understand the overall pattern from a global picture and then discovering possibilities.

The **visual (Vi)** learners prefer to see what they learn by using visual representations such as pictures, diagrams, and charts. While the **verbal (Ve)** learners like information that are explained with words; both written and spoken.

Sequential (Seq) learners prefer to focus on the details by going through the course step by step in a linear way. In the opposite, the **global (G)** learners prefer to understand the big picture by organizing information holistically.

2.3 K-Means Clustering Algorithm

The K-means is an unsupervised algorithm which aims to group similar objects into k clusters [8–10].

In machine learning, unsupervised refers to the problem of finding hidden structure within unlabeled data. Given a collection of objects each with n measurable attributes, k-means is an analytical technique that for a chosen value of k, identifies k clusters of objects based on the objects' proximity to the center of the k groups. The center is determined as the arithmetic average (mean) of each cluster's n-dimensional vector of attributes.

The following steps illustrate how to find k clusters from a collection of M objects with n attributes, where each object i is described by n attributes or property values $(P_{i1}, P_{i2}, \ldots, P_{in})$ for $i = 1, 2, \ldots, M$:

1. The first step consists in choosing the value of k and the k initial guesses for the centroids.
2. In the second step we compute the distance from each data point P_i, at $(P_{i1}, P_{i2}, \ldots, P_{in})$ to each centroid q located at $q_1, q_2, \ldots q_n$ using the following equation:

$$d(p_i - q) = \sqrt{\sum_{j=1}^{n} \left(p_{ij} - q_j\right)^2} \tag{1}$$

3. Then each point is assigned to the closest centroid. This association defines the first k clusters.
4. After determining the first k clusters, we compute the centroid, the center of mass, of each newly defined cluster from Step 3 using the following equation.

$$(\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_n) = \left(\frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} p_{i1}}{m}, \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} p_{i2}}{m}, \ldots, \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} p_{\mathrm{in}}}{m}\right) \tag{2}$$

5. Finally, we Repeat Steps 2, 3 and 4 until the algorithm reaches the final answer.

2.4 Naïve Bayes

The Naive Bayes is a probabilistic classification method based on Bayes' theorem with a few tweaks [11, 12]. According to the Bayes' Theorem, the conditional probability of event C occurring, given that event A has already occurred, is denoted as P(C/A), which can be found using the following formula:

$$P(c \backslash A) = \frac{P(A \backslash C)\ *\ P(C)}{P(A)} \tag{3}$$

A more general form of Bayes' theorem assigns a classified label $(\mathrm{C} \in c_1, c_2, \ldots, c_n)$ to an object A with multiple attributes $(a_1, a_2, \ldots, a_m)$ such that the label corresponds to the largest values of $p(c_i/A)$ for $i = 1, 2, \ldots, n$. Mathematically, this is shown in the following equation:

$$P(C_i \backslash A) = \frac{P(a_1, a_2, a_3, \ldots a_m \backslash C_i)\ *\ P(C_i)}{P(a_1, a_2, a_3, \ldots a_m)}, \quad i = 1, 2, \ldots, n \tag{4}$$

With two simplifications, Bayes' theorem can be extended to become a naive Bayes classifier.

The first simplification is to use the conditional independence assumption. That is, each attribute is conditionally independent of every other attribute given a class label C_i. See the following equation:

$$P(a_1, a_2, \ldots a_m \backslash C_i) = P(a_1 \backslash C_i) P(a_2 \backslash C_i) \ldots P(a_m \backslash C_i) = \prod_{j=1}^{m} P\left(a_j \backslash C_i\right) \tag{5}$$

The second simplification is to ignore the denominator $P(a_1, a_2, a_3, \ldots a_m)$. Because $P(a_1, a_2, a_3, \ldots a_m)$ appears in the denominator of $\boldsymbol{P}(\boldsymbol{C_i} \backslash \boldsymbol{A})$ for all values of i, removing the denominator will have no impact on the relative probability scores and will simplify calculations.

After applying the two simplifications mentioned earlier, the Eq. (4) can be extended to become a naive Bayes classifier as follow:

$$P(C_i \backslash A) \sim P(C_i) \cdot \prod_{j=1}^{m} P\left(a_j \backslash C_i\right), \quad i = 1, 2, \ldots, n \tag{6}$$

As a result, for an object A, the naïve Bayes classifier assigns the class label C_i that maximizes the equation:

$$P(C_i) \cdot \prod_{j=1}^{m} P\left(a_j \backslash C_i\right), \quad i = 1, 2, \ldots, n \tag{7}$$

3 Related Works

Many approaches have been proposed to automatically detect students' learning styles based on machine learning techniques. Those literature works have been relied on various classifiers. Feldman et al. [13] found that the Bayesian network classifier is one of the most widely adopted classifiers to infer the leaning style.

Garcia et al. [14] used Bayesian Networks to detect the learning style of a student in a Web-based education system. To evaluate the precision of their proposed approach, they compared the learning style detected by their approach against the learning style obtained with the index of learning style questionnaire.

Bunt and Conati [15] addressed this problem by building a BN able to detect the difficulty that learners face during the exploration process; and then providing specific assessment to guide and improve the learner's exploration of the available material.

Decision tree is a classification algorithm that is also frequently used in the field of automatic detection of learning styles. Kalhoro [16] proposed an approach to automatically detect the learners' learning styles from web logs of the students using the Data Mining technique, and the Decision Trees classifier. Kolb's learning style theory was incorporated to understand e-learners' learning styles on web. Pantho and Tiantong [17] proposed an approach to classify VARK (Visual, Aural, Read/Write, Kinesthetic) learning styles of learners by using Decision Tree C4.5 algorithm. Data concerning learning styles of learners were collected via a questionnaire responded by 1205 students. The collected data were then classified using Decision Tree C4.5 algorithm.

Neural networks are also commonly used in the automatic detection of learning styles, Hmedna et al. [18] proposed an approach that uses neural networks to identify and track learners learning styles in order to ensure efficient recommendation of resources. Their work was based on Felder-Silverman' dimensions. Hmedna et al. [18] introduced an automatic student modeling approach for identifying learning style in learning management systems according to FSLSM. They proposed the use of fuzzy cognitive maps FCMs as a tool for identifying learner's learning style. FCMs are a soft computing tool which is a combination of fuzzy logic and neural network.

Similarly to the previous algorithms; KNN is frequently used to detect the learning styles automatically, Chang et al. [19] proposed a learning style classification mechanism to classify and then identify students' learning styles. The proposed mechanism improves k-nearest neighbor (k-NN) classification and combines it with genetic algorithms (GA).

As can be noticed, all the previous works relied on a learning style model in their approaches. Most of the proposed approaches used the FSLSM's dimensions and considered that there are 8 learning styles where each one corresponds to a dimension's category. In reality, there are sixteen learning style combinations obtained by combining one category from each dimension. Similarly, to the previous work; we have also relied on the FSLSM, but by considering sixteen LSC.

4 Methodology

An adaptive E-learning system takes into account the learner's learning style and provides contents to the learners based on their preferred learning styles which are identified using the learners' sequences. To identify a learner's learning style we have to rely on a standard learning style model such as FSLSM, where the captured sequences can be labeled with a specific learning style combination using an unsupervised algorithm. In order to implement the unsupervised algorithm, the learner's sequences which have been extracted from the log file, must be transformed to the input of that algorithm.

The learning sequence of a learner is defined by the various learning objects accessed by that specific learner during a session. Each sequence contains the sequence id, session id, learner id, and the set of learning objects accessed by the learner in a session.

After detecting the learners' sequences; our first aim is to classify them according to FSLSM by assigning a specific learning style combination to each sequence, and our second aim is to use those labeled sequences as training set in order to predict the learning style of a new sequence.

In the first step we have used a clustering algorithm, while in the second step we have used a classification algorithm. In our proposed approach we have used the two following algorithms:

- K-means algorithm: in order to assign a label to each sequence based on FSLSM.
- Naïve Bayes classifier: in order to classify a new learner, or a new sequence of an existing learner according to the FSLSM.

The following schema resumes our proposed approach (Fig. 1).

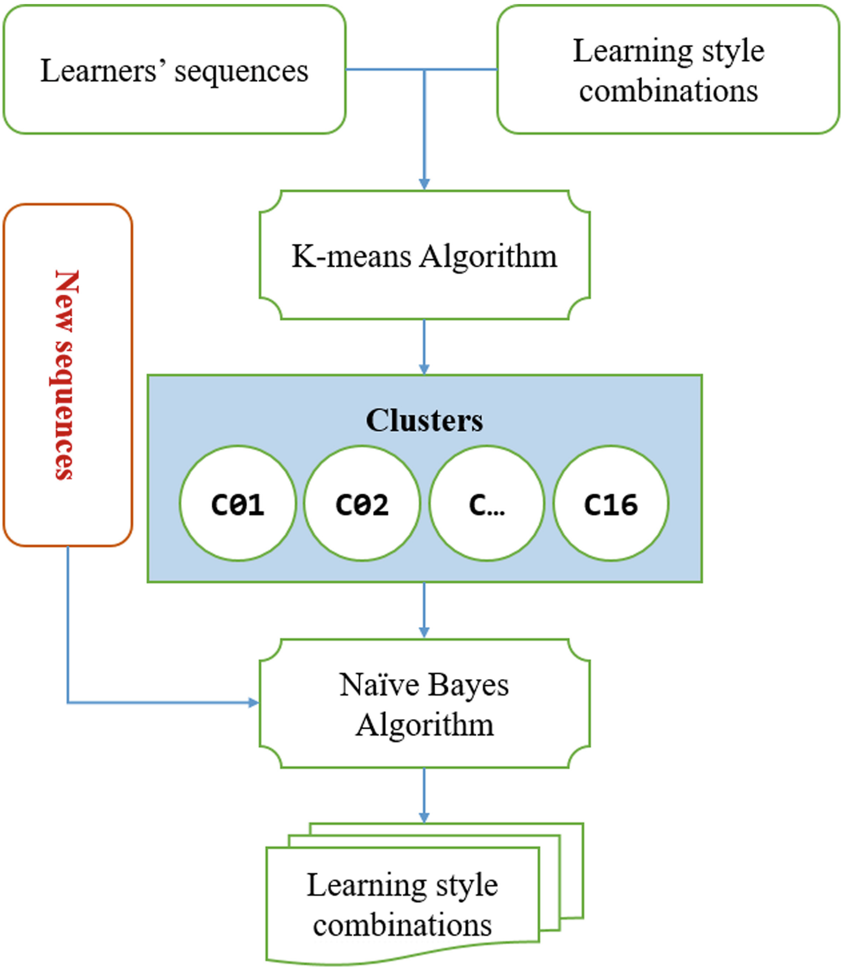

Fig. 1. Our approach

The next subsections are organized as follows: first of all, we will represent how to match each learning object with its appropriate learning style combination, then we will describe how the clustering and classification algorithms were used in our approach.

4.1 Matching LSC to Learning Objects

In order to match a LSC to a learning object, we should have relied on a LSM, in our work we have based on the FSLSM for many reasons, one important reason, is that the FSLSM considers that the student's learning style can be changed unexpectedly and in a non-deterministic way [4], therefore, our approach aims to update the student's learning style dynamically after each interaction with the adaptive e-learning systems. According to the previous researches [5], the FSLSM is the most used in adaptive e-learning systems and the most appropriate to implement them. According to FSLSM there are four dimensions, where each dimension contains two opposite categories, and each learner prefers a specific category in each dimension. Thus, to identify the learner' learning style; we have to determine a combination composed by one category from each dimension. As a result we will obtain sixteen combinations:

`Learning Styles Combinations (LSCs) = {(A,Sen,Vi,G), (A,Sen,Vi,Seq), (R,Sen,Vi,G), (A,Sen,Ve,Seq), (A,Sen,Ve,G), (R,Sen,Ve,Seq), (R,Sen,Ve,G), (A,I,Ve,G), (A,I,Vi,Seq), (A, I,Vi,G), (R,I,Vi,Seq), (R,I,Vi,G), (R,S,Vi,Seq), (A,I,Ve, Seq), (R,I,Ve,Seq), (R,I,Ve,G)}.`

Basically, we suppose that each LSC reflects the preferred learning objects (Los) that are accessed by a learner during the learning process, so in order to identify the LSC for each learner, we first have to match the Los with its appropriate LSC. Based on the matching table presented in our previous work [20, 21]; we have obtained the following table, where the mapped learning objects are considered as feature values of the K-means clustering algorithm.

4.2 The K-Means Algorithm

The first step in our approach consists in using the K-means algorithm in order to classify the learners' sequences according to the sixteen learning style combinations where the sequences of learners are given as an input to the algorithm, and the sixteen LSCs are given as the labels to assign to the resulted clusters.

After extracting the learners' sequences from the log file using data mining techniques, we can use them as an input to the K-means by turning them into a matrix with M rows corresponding to M sequences and sixteen columns to store the attribute values where the attributes of each sequence correspond to the sixteen Los presented in the previous mapping table (Table 1). Therefore each sequence S_i is described by sixteen attributes values $(a_{i1}, a_{i2}, \ldots, a_{i16})$ for $i = 1, 2, \ldots, M$ where a_{i1} corresponds to the number of occurrences of the Lo video in a sequence S_i, a_{i2} Corresponds to the number of occurrences of the Lo PTT, and so on.

In order to perform the k-means algorithm we used the R software framework for statistical analysis and graphics, and in order to write and execute the R code easily, we used the RStudio as a graphical user interface. The dataset employed in our approach was extracted from the E-learning platform's log file[1] of Sup'Management Group.[2] This dataset records 1235 learners' sequences. The following steps describe how we used the K-means with the R language.

- Firstly, we installed the following necessary packages: plyr, ggplot2, cluster, lattice, graphics, grid, gridExtra.
- Secondly, we have imported the data file that contains 17 columns, where the first column holds a sequence identification (ID) number, and the other columns store the number of occurrences of the sixteen Los in each sequence. Because the

[1] http://www.supmanagement.ma/fc/login/index.php.

[2] http://www.supmanagement.ma/fc/.

Table 1. Learning objects as per FSLSM

Cluster ID	Combination	Videos	PPTs	Demo	Exercise	Assignments	PDFs	Announcements	References	Examples	Practical material	Forum	Topic list	Images	Charts	Email	Sequential
C01	(R, I, Ve, G)	3	2	0	1	1	3	2	3	0	0	1	2	0	0	1	0
C02	(A, I, Ve, G)	3	2	1	2	2	2	1	2	0	0	1	2	0	0	1	0
C03	(R, Sen, Ve, G)	3	1	0	1	1	3	2	2	1	1	0	1	0	0	1	0
C04	(A, Sen, Ve, G)	3	1	1	2	2	2	1	1	1	1	0	1	0	0	1	0
C05	(R, I, Vis, G)	3	2	0	1	1	2	1	4	0	0	1	2	1	1	0	0
C06	(A, I, Vi, G)	3	2	1	2	2	1	0	3	0	0	1	2	1	1	0	0
C07	(R, S, Vi, G)	3	1	0	1	1	2	1	3	1	1	0	1	1	1	0	0
C08	(A, S, Vi, G)	3	1	1	2	2	1	0	2	1	1	0	1	1	1	0	0
C09	(R, I, Ve, Seq)	3	2	0	1	1	3	2	3	0	0	1	1	0	0	1	1
C10	(A, I, Ve, Seq)	3	2	1	2	2	2	1	2	0	0	1	1	0	0	1	1
C11	(R, Sen, Ve, Seq)	3	1	0	1	1	3	2	2	1	1	0	0	0	0	1	1
C12	(A, Sen, Ve, Seq)	3	1	1	2	2	2	1	1	1	1	0	0	0	0	1	1
C13	(R, I, Vi, Seq)	3	2	0	1	1	2	1	4	0	0	1	1	1	1	0	1
C14	(A, I, Vi, Seq)	3	2	1	2	2	1	0	3	0	0	1	1	1	1	0	1
C15	(R, Sen, Vi, Seq)	3	1	0	1	1	2	1	3	1	1	0	0	1	1	0	1
C16	(A, Sen, Vi, Seq)	3	1	1	2	2	1	0	2	1	1	0	0	1	1	0	1

sequence id was not used in the clustering analysis, it was excluded from the K-means input matrix.

- Finally, the K-means algorithm was executed giving k = 16, the results are displayed in the two following captured pictures (Fig. 2).

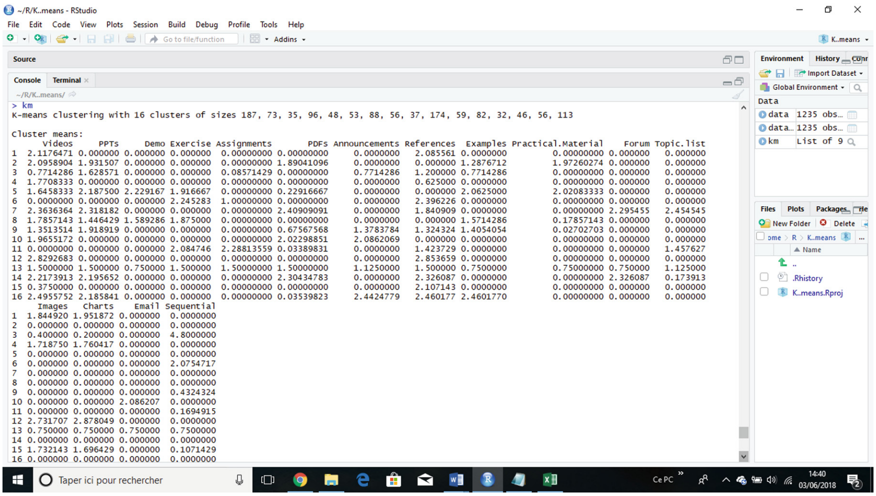

Fig. 2. The location of the cluster means

The picture above shows the number of sequences in each cluster and the coordinates of the clusters' centroid. The obtained clusters were labeled with the LSCs based on the minimum distance between the clusters' centroid and the LSCs' vectors that were presented in Table 1. The result of the clustering is shown in Table 2.

4.3 Naïve Bayes Algorithm

After applying the K-means algorithm and labeling the sequences with the LSCs, the labeled sequences were used as a training dataset to train the classification algorithm, and then use it to predict the LSC for a new sequence.

In our work we have applied the Naive Bayes classifier for many reasons. First of all, because it's one of the most efficient machine leaning algorithms, it learns fast and predicts equally so, and it doesn't require lots of storage. A very important characteristic of Naïve Bayes for our work is that it is a probabilistic classification method, therefore, in our approach we consider that the learner's LSC is not deterministic and not stationary since it can be changed during the learning process in an unexpected way, thus we can measure the LSC for a given learner after each iteration using a probabilistic method.

Given a sequence with a set of attributes A = $\{a_j/j = 1, 2\ldots, 16\}$ where j corresponds to one of the sixteen learning objects existed in the matching table (Table 1),

Table 2. Result of k-means clustering algorithms

Cluster ID	Combination	Number of sequences in each cluster	Total
C01	(R, I, Ve, G)	58	1235
C02	(A, I, Ve, G)	73	
C03	(R, Sen, Ve, G)	174	
C04	(A, Sen, Ve, G)	48	
C05	(R, I, Vis, G)	82	
C06	(A, I, Vi, G)	56	
C07	(R, S, Vi, G)	187	
C08	(A, S, Vi, G)	59	
C09	(R, I, Ve, Seq)	35	
C10	(A, I, Ve, Seq)	32	
C11	(R, Sen, Ve, Seq)	113	
C12	(A, Sen, Ve, Seq)	56	
C13	(R, I, Vi, Seq)	46	
C14	(A, I, Vi, Seq)	96	
C15	(R, Sen, Vi, Seq)	37	
C16	(A, Sen, Vi, Seq)	53	

and a_{ij} takes one value: (yes or No), yes if the jth learning object exists in the sequence, No if the jth learning object doesn't exist in the sequence,

Given a set of classified labels C = $\{C_1, C_2, \ldots, C_{16}\}$ where C_{i} corresponds to one of the sixteen LSCs.

According to the Naive Bayes Classifier, for each new sequence S_{i}, we will assign the classifier label C_{i} that maximizes the following equation:

$$P(C_i) \cdot \prod_{j=1}^{16} P\left(a_j \backslash C_i\right), \quad i = 1, 2, \ldots, 16 \tag{8}$$

In order to predict the LSC for new sequences, we used the Byes naïve in R with the package e1071, and in order to be sure about the efficiency of the algorithm, we did an experiment using the confusion matrix in R with caret. The following section describes the experiment steps and the obtained results.

5 Experiment and Results

5.1 Performance Metrics for Classification Problems

In order to evaluate the performance of the classifier used in our approach, we have used the confusion matrix technique. The confusion matrix technique is a specific table layout that summarizes the number of correct and incorrect predictions in each class, and it is used to compute several validation metrics.

We suppose that we have a confusion matrix with n classes; the following Eqs. (9)–(12) show how to compute the total number of false negative (FN), false positive (FP), true negative (TN), and the true positive (TTP).

FN is the number of instances the classifier predicted as negative but they are positive.

$$FN_i = \sum_{\substack{*20cj = 1 \\ j \neq 1}}^{n} x_{ij} \tag{9}$$

(FP) is the number of instances the classifier predicted as positive but they are negative.

$$FP_i = \sum_{\substack{*20cj = 1 \\ j \neq 1}}^{n} x_{ji} \tag{10}$$

(TN) is the number of negative instances the classifier correctly identified as negative.

$$TN_i = \sum_{\substack{*20cj = 1 \\ j \neq i}}^{n} \sum_{\substack{*20cj = 1 \\ j \neq i}}^{n} x_{jk} \tag{11}$$

(TP) is the number of positive instances the classifier correctly identified as positive.

$$TTP_{all} = \sum_{j=1}^{n} x_{jj} \tag{12}$$

In order to evaluate the performance of a classifier, the following measures (Eqs. 13–17) can be computed for each class i based on the equations described above.

Precision (Positive Predictive value) is the fraction of true positive instances among the predicted positive instances.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{13}$$

Negative Predictive Value (NPV) is the fraction of true negative instances among the predicted negative instances.

$$NPV_i = \frac{TN_i}{TN_i + FN_i} \tag{14}$$

Recall (Sensitivity) is the proportion of positive instances that are correctly classified as positive.

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{15}$$

Specificity (True negative rate) is the proportion of negative instances that are correctly classified as negative.

$$S_i = \frac{TN_i}{TN_i + FP_i} \tag{16}$$

Accuracy, defines the rate at which a model has classified the records correctly.

$$A = \frac{TTP_{all}}{total\ number\ of\ classifications} \tag{17}$$

5.2 Experiment

In order to compute a confusion matrix we need a dataset with expected outcome values, and a test dataset with predicted outcome values. In our experiment, the dataset with expected outcome value corresponds to the 128 sequences that we have taken from the training dataset obtained after performing the k-means algorithm. While the test dataset with predicted outcome values corresponds to the same 128 sequences after removing their LSCs' classes and predicting them again using the naive Bayes classifier. Therefore we have obtained a confusion matrix with sixteen LSCs' classes, where for each class we have computed the number of correct and incorrect predictions using the expected and predicted classes' values.

To compute the confusion matrix and the validation metrics described in the precedent subsection we have used the R with the package caret, the following subsection describes the obtained results.

5.3 Results and Discussion

A well performed model should have for each class a high Precision (PPV), NPV, Recall (Sensitivity) and Specificity that are perfectly 1, and it should also have a high Accuracy.

The captured picture below shows the results obtained after computing the confusion matrix using R with the package caret. As can be noticed, all the validation metrics mentioned above have high values, so we can say that the classifier used in our approach has well performed (Fig. 3).

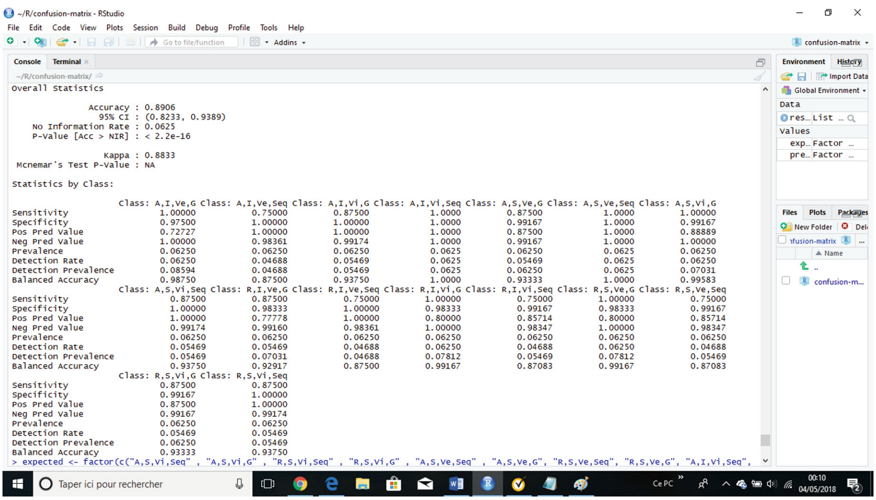

Fig. 3. The confusion matrix obtained from the expected and predicted datasets used in the experiment

6 Conclusion

In this work, we have proposed an approach which aims to predict the learners' learning style automatically, this approach consists of two steps; in the first step the learners' sequences were extracted from the log file then transformed to an input of the K means algorithm. The k means algorithm was used to group students into sixteen clusters based on FSLSM, where each cluster was labeled with a learning style combination. The second step consists in performing an unsupervised algorithm (Naive Bayes) to predict the LS for a new sequence. We evaluated the performance of our approach using the confusion matrix. The produced results show that our approach performs well. In the future work we will have compared the performance of the naive Bayes classifier with other machine learning techniques such as the neural networks and decision tree.

References

1. Laschinger, H.K., Boss, M.W.: Learning styles of nursing students and career choices. J. Adv. Nurs. **9**(4), 375–380 (1984)
2. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Eng. Educ. **78**(7), 674–681 (1988)
3. Biggs, J.: Study process questionnaire manual. Student approaches to learning and studying (1987)
4. Graf, S., Kinshuk.: Advanced adaptivity in learning management systems by considering learning styles. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 235–238 (2009)

5. Kuljis, J., Liu, F.: A Comparison of learning style theories on the suitability for elearning. In: IASTED International Conference on Web Technologies, Applications, and Services, pp. 191–197 (2005)
6. Garity, J.: Learning styles basis for creative teaching and learning. Nurse Educ. **10**(2), 12–16 (1985)
7. Keefe, J.W.: Learning style: an overview. Student Learn. styles Diagnosing Prescr. Programs **1**, 1–17 (1979)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–18 (2009)
9. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2**(3), 283–304 (1998)
10. Kaufman, L., Rousseeuw, P.J.: Partitioning around medoids (Program PAM). In: Finding Groups in Data: An Introduction to Cluster Analysis, pp. 1–67 (2008)
11. EMC.: Data Science & Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Willey, India (2015)
12. Richert, W., Coelho, L.P.: Building Machine Learning Systems with Python, 1st edn. Packt Publishing Ltd., Birmingham (2013)
13. Feldman, J., Monteserin, A., Amandi, A.: Automatic detection of learning styles: state of the art. Artif. Intell. Rev. **44**(2), 157–186 (2015)
14. García, P., Amandi, A., Schiaffino, S., Campo, M.: Using Bayesian networks to detect students' learning styles in a web-based education system. In: Argentine Symposium on Artificial Intelligence (ASAI'2005), vol. 11, pp. 115–126 (2005)
15. Bunt, A., Conati, C.: Probabilistic student modelling to improve exploratory behaviour. User Model. User-Adapted Interact. **13**(3), 269–309 (2003)
16. Kalhoro, A.A., Rajper, S., Mallah, G.A.: Detection of E-learners' learning styles: an automatic approach using decision tree. Int. J. Comput. Sci. Inf. Secur. **14**(8), 420–425 (2016)
17. Pantho, O., Tiantong, M.: Using decision tree C4. 5 algorithm to predict VARK learning styles. Int. J. Comput. Internet Manage. **24**(2), 58–63 (2016)
18. Hmedna, B., El Mezouary, A., Baz, O., Mammass, D.: A machine learning approach to identify and track learning styles in MOOCs. In: 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), pp. 212–216 (2016)
19. Chang, Y.C., Kao, W.Y., Chu, C.P., Chiu, C.H.: A learning style classification mechanism for e-learning. Comput. Educ. **53**(2), 273–285 (2009)
20. El Allioui, Y.: Advanced prediction of learner's profile based on felder silverman learning styles using web usage mining approach and fuzzy C-means algorithm. Int. J. Comput. Aided Eng. Technol. (in press)
21. El Aissaoui, O., El Madani El Alami, Y., Oughdir, L., El Allioui, Y.: Integrating web usage mining for an automatic learner profile detection: a learning styles-based approach. In: 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1–6 (2018)