Chang, M. et al. (Eds.) (2019). Proceedings of the 27th International Conference on Computers in Education. Taiwan: Asia-Pacific Society for Computers in Education

Knowledge-based recommendation system for teaching computational thinking in primary level students

Julio Vera-Sanchi, Eduardo De-Rivero[b], Christian Condori-Mamani[c], Vidal Soncco-Merma [d], Gustavo Suero-Soto [e], Klinge Villalba-Condori [f]
[abcde f]Universidad Catolica de Santa Maria, Peru
[a]jveras@ucsm.edu.pe
[b]ederivero@ucsm.edu.pe
[c]ccondorim@ucsm.edu.pe
[d]vsoncco@ucsm.edu.pe
[e]gsuero@ucsm.edu.pe
[f]kvillalba@ucsm.edu.pe

Abstract: The proposal is a video game developed in Unity, which interacts with the students according to their degree of studies and, according to the STEM curriculum, it is the subject that the student has to learn and in what degree of difficulty, so Once the registration is complete, the video game will provide levels according to your academic degree so that the student interacts with the video game and use the Vector Machine Support (SVM) algorithm that enters the student data and video game data, such as its level, difficulty, time and number of movements and its result send it to the recommendation system to determine what the student should learn and provide related links or documentation, or see if he learned and at what level, verify the SVM because it is a discriminatory classifier formally defined by a hyperplane of separation, which in our case only interests us to know if the student learned or not.
Keywords: Recommendation system, machine learning, computational thinking.

1. Introduction

At present we have a large number of students, who receive basic training and who does not agree with the vanguard of the educational system such as computational thinking. Part of our reality is the still existing digital divide that exists in our country Peru, since only 48.7% of the population aged 6 and over have access to the Internet. But if we separate the urban area from the rural area, the difference is marked, in the urban area 58.2% has a connection, while the rural area only 15.4%, this digital divide means that resources and technologies cannot be accessed to train teachers and the development in computational thinking students. The gamification in the educational context and development of computational thinking helps to understand about the concepts its application in daily problems, from a playful point of view, improving the learning process of students, that is why there is a set of applications that help to develop certain skills, but many of these systems do not have an intelligent approach to all the data that can be collected through children's interaction with these applications.[22] As part of a change in this approach to the development of computational thinking is that we make use of Machine Learning techniques, to make the processing of all the information collected by students, through algorithms that collect the high dimensionality of the data set of the interactions to generate learning predictions or not, and that serves us as part of a process for the generation of a knowledge-based recommendation system that helps students, recommending educational resources found in a specialized database according to the STEM methodology (Science, Technology, Engineering and Mathematics), this proposal is based on creating an adaptive learning environment according to the students' current knowledge.[21]

2. Conceptual Framework

2.1 Recommendation Systems

Recommendation systems (RS) are software tools and techniques that offer suggestions on the elements that can be used by the user. In this introductory chapter we briefly discuss basic ideas and concepts of RS. Our main objective is to delineate, in a coherent and structured manner, the chapters included in this manual and help the reader navigate the extremely rich and detailed content offered in the manual. [12] To provide a first overview of the different types of SR, it has been possible to find about six different kinds of recommendation approaches, but as the need grows and new problems arise, they can increase but the most important are:

- Content based: The system learns to recommend items similar to what the user liked in the past. The similarity of the elements is calculated based on the characteristics associated with the elements compared. For example, if a user has positively rated a movie that belongs to the genre of comedy, then the system can learn to recommend other movies of this genre. [12]
- Collaborative filtering: the simplest and most original implementation of this approach [16] recommends the active user the articles that other users with similar tastes liked in the past.
- Knowledge-based: Knowledge-based systems recommend elements based on specific domain knowledge about how certain features of the item meet the needs and preferences of users and, ultimately, how the item is useful for the user. Notable knowledge-based recommendation systems are based on cases [2]. In these systems, a similarity function estimates how much the user's needs (description of the problem) coincide with the recommendations (solutions of the problem). Here the similarity score can be interpreted directly as the utility of the recommendation for the user. Restriction-based systems are another type of knowledge-based RS. In terms of knowledge used, both systems are similar: user requirements are collected; repairs for inconsistent requirements are automatically proposed in situations where solutions cannot be found; and the results of the recommendations are explained. The main difference lies in the way the solutions are calculated. Case based recommendations determine recommendations based on similarity metrics, while restriction-based recommendations predominantly exploit predefined knowledge bases that contain explicit rules on how to relate customer requirements to article characteristics. Knowledge-based systems tend to work better than others at the beginning of their implementation, but if they are not equipped with learning components, they can be overcome by other shallow methods that can exploit human / computer interaction records (such as of Collaborative Filtering). [9]
- Hybrid recommendation systems: These RSs are based on the combination of the techniques mentioned above. A hybrid system that combines techniques A and B tries to use the advantages of A to correct the disadvantages of B. For example, Collaborative Filtering methods suffer from problems of new items, that is, they cannot recommend items that do not have qualifications. [9]

2.2 Computational Thinking

At present, there is a great current of judgments in the world that support the use of the computer in the teaching-learning process, which allow the development of various types of thinking, skills and competencies of a student of the 21st century. Computational thinking has been positioned as an alternative to develop the aforementioned characteristics. [10]. "Computational Thinking is a problem-solving process that includes, but is not limited to the following characteristics" [20]:

- Formulate problems so that computers and other tools can be used to solve them.
- Organize and analyze data logically.
- Represent data using abstractions, such as models and simulations.
- Automate solutions through algorithmic thinking.
- Identify, analyze and implement possible solutions in order to find the most efficient and effective combination of steps and resources.
- Generalize and transfer this problem-solving process to a great diversity of these.[20]

Existing literature supports the inclusion of TC in the K-12 curriculum (educational systems for primary and secondary schooling. It is used in the United States, Canada, Turkey, Philippines, Australia and Ecuador. It is formed by the initial in English for garden of infants or Kindergarten (between four to six years of age) and the number indicating the last grade (12; between seventeen and nineteen) of free education.), in multiple subjects and from Primary grades onwards. The use of computers as a context for CT skills is often possible, but care must be taken to ensure that CT does not combine with programming or instructional technology in general. [19]

2.3 STEM Methodology

The term STEM is the acronym for the English terms Science, Technology, Engineering and Mathematics. The term was coined by the National Science Foundation (NSF) in the 1990s. [8] the term STEM, only dried, only serves to group the 4 major areas of knowledge in which scientists and engineers work. The concept of "STEM Education" (STEM Education) has been developed as a new way of teaching Science, Mathematics and Technology together (in general, not only computer science) with two distinct characteristics: [11]

- Teaching-learning of Science, Technology, Engineering and Mathematics in an integrated manner rather than as compartmentalized areas of knowledge. Integrated instruction means any program in which there is an explicit assimilation of concepts from two or more disciplines.
- With an engineering approach to the development of theoretical knowledge for its subsequent practical application, always focused on solving technological problems. [14]

In Table 1, we list several of those standards that could be relevant to our teaching concept.[7]

The student may be able to	Level	Grade
Recognize that the software is created to control computer operations.	1	Initial
Understand and use the basic steps in solving algorithmic problems (for example, problem statement and exploration, examination of sample instances, design, implementation and testing)	1st	3rd to 3rd grade
Develop a simple understanding of an algorithm (e.g., search, sequence of events or classification) using exercises without a computer	1st	3rd grade
Use the basic steps in solving algorithmic problems to design solutions (for example, problem statement and exploration, sample sample exam, design, solution implementation, test, evaluation)	2nd	4th grade to 1st grade
Define an algorithm as a sequence of instructions that can be processed by a computer	2nd	4th grade to 1st grade
Act search and classification algorithms.	2nd	4th grade to 1st grade
Describe and analyze a sequence of instructions that are followed (for example, describe the behavior of a character in a video game according to the rules and algorithms)	2nd	4th grade to 1st grade
Represent the data in several ways, including text, sounds, images and numbers	2nd	4th grade to 1st grade
Use abstraction to break down a problem into sub problems	2nd	4th grade to 1st grade
Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts	3rd	2nd to 2nd secondary
Explain how sequence, selection, iteration and recursion are building blocks of algorithms	3rd	2nd to 2nd secondary
Describe how different types of data are stored in a computer system.	3rd	2nd to 2nd secondary
Compare and contrast simple data structures and their uses (for example, matrices and lists)	4	2nd to 4th secondary
Decompose a problem by defining new functions and classes.	5th	2nd to 4th grade

2.4 Support Vector Machine - SVM

A support vector machine (SVM) is a discriminative classifier formally defined by a separation hyperplane. In other words, given the training data labeled (supervised learning), the algorithm generates an optimal hyperplane that categorizes new examples. In two dimensional spaces, this hyperplane is a line that divides a plane into two parts where each class is on each side.

Vector support machines or Support Vector Machine are a set of supervised learning algorithms that are properly related to classification and regression problems. Given a set of data (of samples) we can label the classes and train an SVM to build a model that predicts the class of a new sample. Basically, an SVM is a model that represents the sample points in space, separating the classes into 2 spaces as wide as possible by means of a separation hyperplane defined as the vector between the 2 closest points of the 2 classes, to which It's called support vector. When new samples (data) are mapped to that model, depending on the spaces to which they belong, they can be classified into one or the other class. svm02 [17]

The SVM seeks a hyperplane that optimally separates the points of one class from that of another. In this concept of "optimal separation" is where the fundamental characteristic of the SVM resides, they are also known as maximum margin classifiers, because they look for the hyperplane that has the maximum distance (margin) with the points that are closest to it. [13]

The SVM has adjustment parameters that are:

2.4.1 Nucleus

Hyperplane learning in linear SVM is done by transforming the problem using some linear algebra. This is where the nucleus plays a role.

2.4.2 Regularization

The Regularization parameter (often called parameter C in the python sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a hyperplane with a lower margin if that hyperplane does a better job of correctly classifying all training points. On the contrary, a very small value of C will cause the optimizer to look for a separation margin of greater margin, even if that hyperplane mistakenly classifies more points. The images below (same as image 1 and image 2 in section 2) are examples of two different regularization parameters. The left one has an erroneous classification due to the lower regularization value. A higher value leads to results as correct.

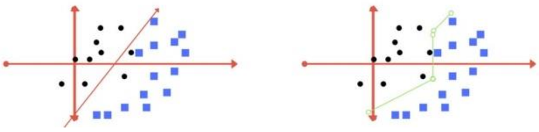


Figure 1: Left: low regularization value, right: high regularization value
Source: self made.

2.4.3 Gamma

The gamma parameter defines how far the influence of a single training example goes, with low values that mean "far" and high values that mean "near." In other words, with low gamma, the points away from the plausible separation line are considered in the calculation of the separation line. Where as high gamma means that the points near the plausible line are considered in the calculation.[13]

Figure 2: Gamma types, high and low Source: self made.

2.4.4 Margin

And finally, the last but very important feature of the SVM classifier. SVM to core tries to achieve a good margin. A margin is a line separation to the nearest class points. A good margin is one in which this separation is greater for both classes. The images below give a visual example of good and bad margins. A good margin allows points to be in their respective classes without crossing to another class..[13]

Figure 3: Types of margins, good margin and bad margin Source: self made.

2.5 Gammification

Gamification has been defined as a process for improving services with (motivational) possibilities to invoke gaming experiences and additional behavioral outcomes [6] [5]. In defining gamification, Huotari and Hamari [31] highlight the role of gamification in invoking the same psychological experiences as games (in general). Deterding et al. deterding2011game, on the other hand, emphasize that the possibilities implemented in gamification have to be the same as those used in games, regardless of the results. However, it is not clear what possibilities are exclusive to the games, as well as what psychological results can be strictly considered as a result of the games. From the perspective of these definitions, there is room for a wide variety of studies that could be framed as gamification.[13]

3. Proposal

3.1 Data Model

Table 2 shows the data model used as a proposal in the computational thinking recommendation system. A brief explanation of the model will be given below. A basic entry for data models is the tuple < Topic, Difficulty >, indicates the value of the difficulty in each topic. In our research, the themes have been selected based on the Curriculum Mesh based on the teaching methodology STEM.

The level of skills for different subjects was modeled on a 3-point scale (grades). The tuple < topic, difficulty > has the relationship between a topic and its degree of difficulty, because according to the curriculum mesh depending on the grade of the student will determine its difficulty. It also has another meaning in the case of the student's ability with a subject. For example: a user's skill level is described in a topic in their user profile (year in which it is found), and on the other hand, it indicates the required skill of a topic in levels for the video game. The user profile in relation to computational thinking includes skills and their learning goals, where they have to learn. The source of information would be (a) Curriculum: it is what gives us the basis to see what the student needs to learn and to what level of difficulty in relation to their year, (b) Evaluation: how it is obtained from the evaluation that makes the video game. The goal is to see if the student learns or what is missing to finish learning the subject, for this, the results of the video game are used and processed by the recommendation system. For the current paper, we have considered the following main aspect for the application that are required: (a) the levels assigned to the user. In order to have an attainable goal, the recommendation system needs to use the STEM curriculum to have exactly what goals the student needs.

3.2 Recommendation System Architecture

The architecture of the recommendation system receives several inputs as can be seen in the figure 4, such as: Game data, Curriculum, Student Data, Levels and Difficulty. Next, each of the entries will be detailed in context:

Tuple Theme-Skill	
Topic	One or more set of STEM Curriculum Mesh
Skill	One of {1: Bad, 2: Fair, 3: Good}
User Data Model	
User-ID	Primary User Identifier
Student Name	Full Name
Year	Academic Year in which the student is
Skills	A tuple arrangement < *topic, difficulty*>
Level Model	
ID-Level	Primary Level Identifier
Description	Description of the level
Issue-Difficulty	A tuple arrangement <*theme, difficulty*>

Tuple Level-Time	
ID-Level	Primary Task Identifier
Time	Classified in seconds
Tuple Level-Scores	
ID-Level	Primary Task Identifier
Scores	Ranked the number of moves
User Assignment	
User-ID	Primary User Identifier
User Level	A tuple arrangement < *level, time* >

Table 2: Data Model. Source: Self-made.

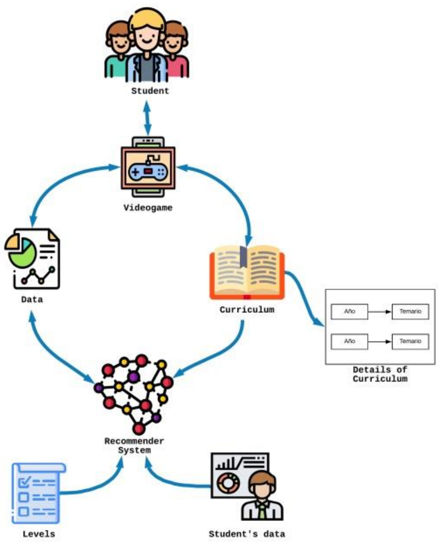


Figure 4: Video Game Architecture

- Videogame - Game data: It is the user interface, it is the one that is responsible for capturing user inputs and as it relates to the system, are the levels that the user completes and plays, each level receives what the student It has taken time to complete the level as your total time, number of movements (scores) from the starting point to the goal and will be stored in a matrix to be sent to the recommendation system.
- Curriculum: This is given by the STEM [4] methodology whose teaching chart for Computational Thinking was as follows: The proposal of this paper is focused on students in the second grade of primary school, so according to 1 it should be focused on understanding and using the basic steps in solving algorithmic problems, See how the steps work for an algorithm to work, and a simple

understanding of what an algorithm is. In addition, since its level is 1, it is of the least difficulty since it only needs to understand how it works and how it can be applied so that in the video game it will be the sequence of steps to reach the goal the way of solving the algorithm and according to your number of movements see how efficient your algorithm is.

- User data: These are the student's data such as their first name, last name, gender, and city, but their most important or interesting data to the recommendation system is in which academic year they are in to validate it with the matrix of the curriculum mesh.
- Levels: These are all the activities performed by the user in the video game and are classified according to their level of difficulty and the characteristic of computational thinking at that level.

3.3 Methodology

This paper aims to investigate and develop a knowledge-based recommendation system [18], with core an algorithm of Super Vector Machine (SVM) [9] supervised learning whose objective is to indicate whether the student learns or not based on a training data and then analyze it in the recommendation system to decide whether the student, in the case he did not learn, recommend a help text or Tell you some other similar activity that you can do to better understand the subject in question. Next, the proposal will be shown, the architectures that comprise the recommendation system and how it interacts with the entries of both the user and an entry that is the curricular mesh, where all the skills that are necessary will be, for this case it will be developed in an application mobile. The paper proposal is a learning method through a videogame for teaching computational thinking, defined in a knowledge-based recommendation system for personalized teaching for each student. Students who wish to learn about computational thinking can do so through a video game developed at Unity, but with guided teaching, this is where the recommendation system comes in to see their progress and analyze according to their input variables and give a recommendation through a prediction. The details are:

- Students will enter the mobile video game developed at Unity and register to have their progress and see, according to their year, what they have to learn in relation to the curriculum.
- Once identified, they will be given the level (subject) to complete with their degree of difficulty.
- Students will be able to interact through the video game that captures the numbers of movements (scores) and the time it takes.
- The records stored by the video game will be part of the entry for the recommendation system, along with the curriculum.
- Through an SVM the recommendation system will evaluate the input data resulting in whether the student learned or did not learn.
- Subsequently, the recommendation system based on the results will dictate that the student needs to learn and if he did not learn, he needs to reinforce so that the issue is clear with his respective difficulty.

3.4 Recommendation Algorithm

The recommendation system is based on the theoretical framework of the STEM curriculum. Student data and video game data (the time it takes to complete the level, the total number of movements, what level and how difficult it is) are the input data for the recommendation system that first passes through the SVM algorithm.

The SVM algorithm has as input data or dimensions the degree, level, difficulty, scores, time and as an objective variable the result (which would be whether it learned or not). The algorithm disaggregates based on a line of separation between the data that is necessary to define whether the student learned or did not learn.

For the training part of the algorithm a dataset of 435403 records containing the input data was used, of which 70 % was collected for training data, and the remaining 30 % for test data. In addition, cross-validation will be implemented so that the algorithm does not fit the data in the event that the dataset has repeated consecutive rows to improve its training.

Then you will want to see its accuracy of recommendation to see if, based on the 30 % that we separated for testing, it did a good job of recommendation and to finish, the whole procedure of the algorithm will be stored in a pickle to be used in the repository that we will use for the videogame to connect with the operation and indicate its student results with the recommendation system based on the curriculum.

A student should be recommended to other levels (or difficulties) according to their results that suits their current situation. This should not be too difficult or too easy for him. Therefore, the usefulness of a learning content with respect to the current knowledge of the user is determined by the current knowledge of the user in the subject and the nature of the learning object and its level of difficulty by adding its number of movements (scores) and the time it took to do it.

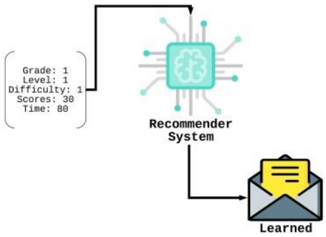


Figure 6: Example of the operation of the recommendation system

3.5 Implementation

To perform the tests, an application was developed on the Unity platform called CGAME, which has two games called MST and TorresH, which help us obtain data for our recommendation system.

3.5.1 CGame

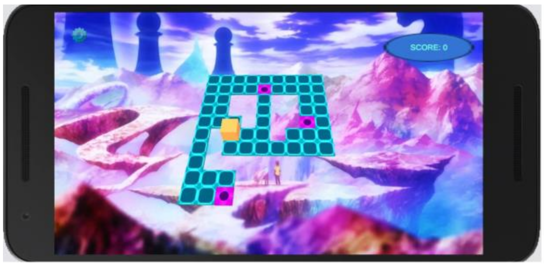
The application has a Login view, in which you have to enter a valid email and password, you also have the registration view in case you do not have a valid account. Once inside the application it will show us the games that will help us to carry out the tests.

Figure 7: CGame - Login

3.5.2 MST

This game consists of moving a cube through all the positions that mark you in the game, to successfully overcome the game, one must use the minimum number of possible movements, these types of problems are solved using algorithms such as the Minimum Expansion Tree (Minimum Spanning tree). This game has three levels (easy, intermediate and difficult), once each level is completed, you get data such as the number of movements and the time it took to overcome that level. The data obtained are sent to the recommendation system which returns additional information to strengthen their knowledge and skills in case they have not satisfactorily exceeded the level.

MST that will teach the student in relation to computational thinking is pattern recognition and algorithm resolution, since to complete the game, the student must move the cube from an initial point to an end point (marked the purple quadrant) and the less movements generated, the more optimal the resolution of the level will be and follow a sequence of steps to complete the level, so that you understand and use the basic steps in solving problems, by steps and therefore understand a compression Simple operation of an algorithm.



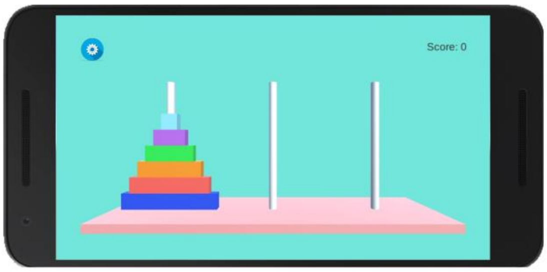


Figure 8: MST - Intermediate Level / TorresH – Advanced Level

3.5.3 TorresH

This game is a simulation of the mathematical game of the Towers of Hanoi, it consists of a number of perforated discs of increasing radius that are stacked inserted into one of the three posts fixed to a board. The objective of the game is to transfer the stack to another of the posts following certain rules, such as that a larger disc cannot be placed on top of a smaller disc. This problem can be solved by recursion and iteratively. In the same way as the previous game, we obtain the number of movements and time, and pass them to the recommendation system.

TorresH has the operation of a tower of Hanoi that, if we compare it with computational thinking, we would cover what is the decomposition, abstraction and resolution of algorithms, since to complete the level, the student has to move The whole tower of blocks from one side to the other without a larger block being on top of the other, only a smaller block can be on top of another. As a decomposition we have the tower divided into blocks of different sizes, the abstraction as a solution to the problem is that, as we can move the blocks between one of smaller size and another and as resolution of algorithms, we have the steps to follow to complete the level, and So understand the definition of an algorithm as a sequence of instructions that can be processed by a computer and explain how the sequence, selection, interaction and recursion in our videogame are building blocks of algorithms.

Once the student's results are obtained in the MSTo in TorresH, an HttpRequest request will be sent to the service where the Recommendation System will be hosted and through its body a JSON will be sent with the following format: {user: 20 , level: 1, difficulty: 2, time: 100, scores: 10, grade: 2} and the answer that the request will give us HttpRequest will be if the student learned or what is missing to comply with what is indicated according to his academic year. As you can see in the following example: {result: 'Didn't learn', help: 'https: //csunplugged.org/en/topics/kidbots/integrations/finding-shapes/'}

4. Results

Synthetic data was tested, that is, the evaluation was done with a methodology expert and data generated randomly from the curriculum using a program to generate the data with a total of about 435403 records, implementing cross-validation or crossvalidation, which is a technique used to evaluate The results of a statistical analysis and ensure that they are independent of the partition between training and test data. It consists of repeating and calculating the arithmetic mean obtained from the evaluation measures on different partitions. It is used in environments where the main objective is prediction and you want to estimate the accuracy of a model that will be implemented. [3] It is a technique widely used in artificial intelligence projects to validate generated models. Using cross validation gave an accuracy of 96 % using 70 % of the data for the test side and the remaining 30 % for simulation.

A graph of precision recall was made, which is defined as the fraction of all the relevant instances divided by the obtained instances. Recovery is the fraction of relevant instances that have been obtained over the total number of relevant instances. Both accuracy and recovery are based on an understanding and measure of relevance. [10] Which indicated that its average accuracy is 92 %.

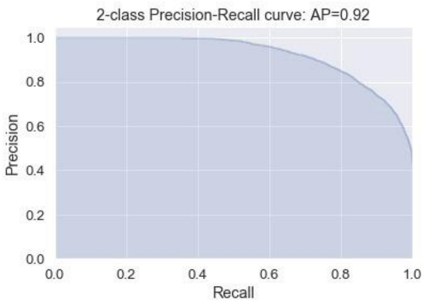


Figure 10: Precision recall.

5. Conclusion

In this paper there was an experimentation and interpretation of the results obtained using a knowledge-based recommendation system, as you can see the experiment was done with synthetic data on second grade elementary students who want to learn about computational thinking but in a dynamic way and through a video game that helps them understand how computers work in a simple way. In the paper you can see the different entries to the system by users such as their data, the curriculum and video game data such as time, scores and number of movements. In the results section we can see that the system has a good prediction for the recommendation of contents since it reaches a 96 % accuracy.

References

[1] K. Brennan and M. Resnick. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, volume 1, page 25, 2012.

[2] D. BRIDGE, M. H. GOKER, L. McGINTY, and B. SMYTH. Case-¨ based recommender systems. *The Knowledge Engineering Review*, 20(3):315–320, 2005.

[3] P. A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice hall, 1982.

[4] S.-C. Fan and K.-C. Yu. How an integrative stem curriculum can benefit students in engineering design practices. *International Journal of Technology and Design Education*, 27(1):107–129, Mar 2017.

[5] J. Hamari. Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service. *Electronic commerce research and applications*, 12(4):236–245, 2013.

[6] K. Huotari and J. Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th international academic MindTrek conference*, pages 17–22. ACM, 2012.

[7] M. S. Khine. *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights*. Springer, 2018.

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[9] T. Mahmood and F. Ricci. Towards learning user-adaptive state models in a conversational recommender system. In *LWA*, pages 373–378.
Citeseer, 2007.

[10] G. G. Meza. Precision and recall, 2018.

[11] G. O. Rebollo, I. M. R. Albaladejo, F. G. Cuadra, and A. C. Sanchez.´ Implantacion de la nueva asignatura "rob´ otica" en ense´ nanza secun-˜ daria y bachillerato. *Investigacion en la escuela´* , (87):65–79, 2015.

[12] F. Ricci, L. Rokach, and B. Shapira. *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011.

[13] J. A. Rodrigo. Support vector machines. https://rpubs.com/Joaquin$_A$*R/*267926,2017.

[14] R. E. Satchwell and F. L. Loepp. Designing and implementing an integrated mathematics, science, and technology curriculum for the middle school. *Journal of Industrial Teacher Education*, 39(3):41–66, 2002.

[15] B. Schafer, J. Konstan, and J. Riedl. E-commerce recommendation applications, data mining and knowledge discovery, vol 5 (1–2), 2001.

[16] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[17] N. C. SHUJUN HUANG. Applications of support vector machine (svm) learning in cancer genomics. *Cancer Genomics Proteomics*, Dec 2017.

[18] J. K. Tarus, Z. Niu, and G. Mustafa. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50(1):21–48, Jun 2018.

[19] A. Yadav, S. Gretter, J. Good, and T. McLean. Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking*, pages 205–220. Springer, 2017.

[20] Villalba-Condori, K. O., Cuba-Sayco, S. E. C., Chávez, E. P. G., Deco, C., & Bender, C. (2018, October). Approaches of learning and computational thinking in students that get into the computer sciences career. In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality (pp. 36-40). ACM.

[21] Espinoza-Suarez, S., Falen, J. M. M., Chipana, S. S., Villalba-Condori, K. O., & Castro-Cuba-Sayco, S.(2019) Project-oriented learning for the achievement of educational outcomes.

[22] Silveira, I. F., & Villalba-Condori, K. O. (2018). An Open Perspective for Educational Games. Journal of Information Technology Research (JITR), 11(1), 18-28.