Artificial intelligence for choosing an agile method

Soukaina Merzouk[1], Soukaina Bouhsissin[2], Touria Hamim[2], Nawal Sael[2], Abdelaziz Marzak[2]
[1]Polydisciplinary Faculty of Sidi Bennour, Chouaib Doukkali University, El Jadida, Morocco
[2]Information Technology and Modeling Laboratory, Faculty of Sciences Ben M'sik, Université Hassan II de Casablanca, Casablanca, Morocco

Article Info

Article history:

Received May 5, 2023
Revised Oct 31, 2023
Accepted Dec 14, 2023

Keywords:

Agile method
Artificial intelligence
Chatbot
Decision tree
Machine learning

ABSTRACT

Agile methods are widely known in different companies, including information technology (IT) companies. They appeared intending to solve the problems of traditional methods while proposing an iterative and incremental cycle. These methods consist of four values and the twelve principles agreed upon in 2001 in a Manifesto. However, each method holds singularities from which it is difficult to choose one to adopt in different project cases. The selection of the method to adopt positively or negatively affects the final product following the criteria of the project and the personnel. Project experts must research and compare methods manually to make a choice, a thing that drains time, which is a key factor in project realization. Currently, there is no intelligent system or model that allows choosing the agile method to adopt for such a project. For this purpose, artificial intelligence (AI) techniques will be used to develop a Chatbot that allows reaching the aim. This Chatbot will be developed based on a decision tree model that will be proposed from an experimental study.

Corresponding Author:

Soukaina Merzouk
Polydisciplinary Faculty of Sidi Bennour, Chouaib Doukkali University
ElJadida, Morocco
Email: merzouk.soukaina@gmail.com

1. INTRODUCTION

Managing a project is a major challenge for any company. Selecting a project management method is not a simple matter. It affects positively or negatively the final product according to the personnel and project criteria [1]. Multiple agile methods exist, although they all adhere to the same set of values and principles. However, there are differences between them, in the process for instance. Moreover, they are not flexible in all project situations. The state-of-the-art study as stated in [2] shows that each method has it is singularities in terms of project/team size, iteration length, and changing. For this purpose, the project managers handle searching for and choosing the appropriate method to manage the project. However, there is no platform or assistant tool to simplify this task for managers. As a result, managers must do the research themselves, either by reading and making comparisons or by analyzing comparative studies or annual reports on the use of management methods, which affects the time required to carry out the project.

Nowadays, there are limited works available on the development of models or systems for project management method selection. In what follows, the research works that are considered interesting will be presented. Alqudah *et al.* [3] interviewed 23 agile experts from 19 teams across thirteen countries to explore the selection process of agile methods. Sixteen factors were identified and categorized into five groups: project nature, team skills, constraints, customer involvement, and organizational culture. The research provides insights on utilizing these factors in agile method selection. Al-Saqqa *et al.* [4] provide a comprehensive review of agile methods, discussing their values, principles, and differences from conventional methods. They also cover popular agile methods, their life cycles, roles, advantages, and disadvantages. The research explores the implementation of agile in big data systems and cloud computing environments, as well as the importance of selecting the most suitable agile method based on task, product sensitivity, and organizational structure. The study emphasizes the need for qualitative justification of managerial decisions and highlights the benefits of agile management in responding to rapid changes and providing managerial alternatives. Rudnichenko *et al.* [5] propose using optimization models to select agile management tools based on cost minimization. It is demonstrated that with the right tools, agile management can help companies make high-quality management choices for strategic growth. Hanslo and Tanner [6] conducted a study on predicting Scrum adoption using machine learning models. They collected survey data and developed three prediction models to identify significant features for predicting Scrum adoption. Pouyandeh *et al.* [7] identified construction industry success criteria and determined the most suitable agile method using a combination of Delphi and analytic hierarchy process (AHP) methodologies. The study collected data through literature reviews and questionnaires, analyzed it using the AHP method, and prioritized agile methods and criteria for selecting the appropriate method. In agile software development, estimating the effort and costs required to complete a project is a crucial process. For this purpose, Govil and Sharma [8] proposed an extended algorithm for estimating effort and costs in agile software development. They considered 36 success factors across different dimensions and evaluated Scrum projects using a dataset of 30 projects. Their approach is cost-effective and requires less effort than existing methods.

To sum up, the use of machine learning for the selection of the agile method to adopt has not yet been implemented. Therefore, a Chatbot with machine learning algorithms is being developed to assist project managers in selecting the appropriate agile method. The Chatbot uses a decision tree model and allows interaction based on criteria and answer formats.

The remainder of the paper is structured as follows: section 2 presents the background of artificial intelligence, machine learning, decision tree, and Chatbot. The research methodology followed in this study is described in section 3. Section 4 presents experimentation and results. Section 5 presents the Chatbot implementation. Section 6 concludes the paper and introduces future work.

2. BACKGROUND

2.1. Artificial intelligence

The term "artificial intelligence", according to Simmons and Chappell [9], refers to the behavior of a machine that, if a human behaves in the same way, is considered intelligent. The first efforts in artificial intelligence consisted of modeling the brain's neurons [10]. The term "artificial intelligence" was invented during a workshop of interested researchers at Dartmouth University in 1956. Artificial intelligence became a new discipline with the purpose of developing computer systems capable of learning, reacting, and making decisions in a complex and changing environment [11]. Machine learning and deep learning are among the subfields of artificial intelligence.

Machine learning deals with the construction of algorithms that, to be useful, rely on a collection of examples of a certain phenomenon. Machine learning can also be defined as the process of solving a practical problem by i) collecting a data set and ii) algorithmically building a statistical model based on that data set [12], [13]. It is used to assist machines in enhancing their data processing capabilities by learning from the data provided [14].

2.2. Decision tree

A decision tree is a predictive model that allows the decomposition of a complex decision process into a series of simpler decisions, offering a more easily interpreted solution. It is used for classification models (known as classification tree) as well as regression models (known as regression tree). Decision trees are a hierarchical model of decisions and their repercussions in operations research [15]–[17]. A decision tree consists of leaves, branches, and nodes. These latter are the positions where a decision is made. The main node is the root. Every decision tree node stands for a feature, while every edge represents possible values. Finally, the leaves represent the final decision [18]. The iterative dichotomiser 3 (ID3), based on the interpretation of Hunt's analysis, is one of the algorithms used to build a decision tree [19]. Also, many algorithms employ decisions tree like the gradient boosting-based tree (GBT) algorithm [20], and the random forest (RF) algorithm [21], [22], as both employ or combine several weak models to build more accurate ones.

2.3. Chatbot

Chatbot or conversational artificial intelligence is a computer program that processes a user's natural language input and generates intelligent, relative responses that are then returned to the user [23], [24]. Chatbots are currently powered by rule-driven or artificial intelligence engines that communicate with users mostly through a text-based interface. It comprises standalone computer applications that may be plugged into one of the several messaging services that have made application programming interface (API) available to developers, such as Facebook Messenger, Skype, and Microsoft Teams. Chatbots converse with users in a similar manner to how humans converse in their daily lives. As voice technology has advanced in recent years, companies such as Google, Apple, and Amazon have developed artificial intelligence speech assistants. Text from a Chatbot or a voice system is processed in the same manner [25]–[27].

3. RESEARCH METHODOLOGY

This section describes systematically the steps followed to develop a prediction model for a project management method as illustrated in Figure 1. The survey response data was converted into a well-defined and structured database. A pre-processing step was then performed to prepare the quality of data before adopting a machine learning model. During this phase, invalid data was eliminated, rare and missing values were replaced, and noise data was adjusted. The 10-fold cross-validation method was applied to estimate the statistical performance of the learning model. This method involves randomly partitioning the data into ten segments, with nine segments allocated for training purposes and one segment reserved for testing. This procedure is repeated 10 times each time reserving a different tenth for testing. Two types of results were obtained, weight by information gain and the decision tree. The decision tree was obtained after applying the cross-validation method, using the ID3, GBT, and RF algorithms. Finally, the evaluation was measured by the accuracy performance metric.

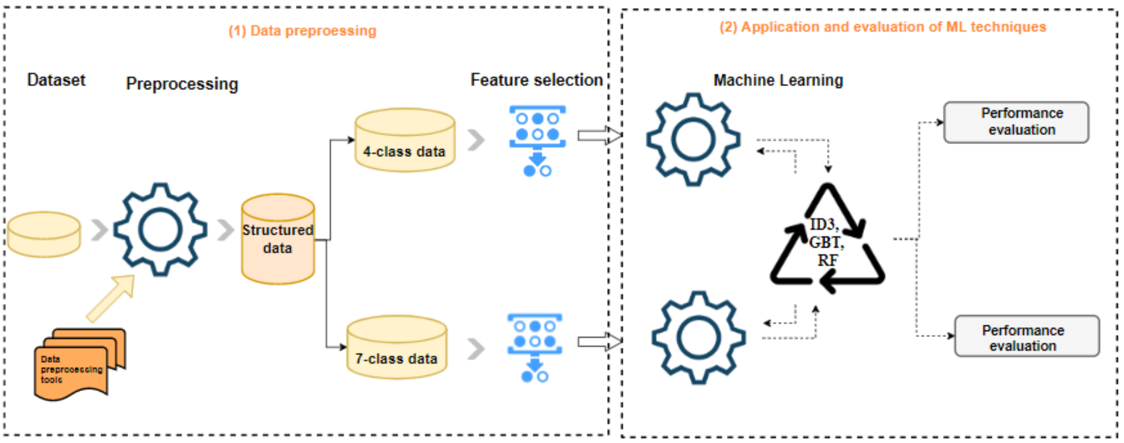


Figure 1. The methodology followed for developing a prediction model for a project management method

4. EXPERIMENTATION AND RESULTS

4.1. Data set construction

To develop our dataset, we started a questionnaire survey. We asked specific questions about the field of software engineering and that generally allows developers to choose the most suitable agile method for their specific needs. These questions were developed from our earlier comparative study [3]. Based on the latter, an in-depth analysis of criteria was performed relevant to the real choice of the management method to be applied. Next, we took a set of expert responses as a sample to confirm the proposed questions. These experts were asked about their project management experiences, more specifically, about the agile method used in the management of their projects and what are the basic criteria that help to make this choice. Our dataset has the characteristics of over 80 project members. Table 1 describes the project criteria, which include member identification (country, company name, and position in the company), project data (project name and description), and project criteria (project size, roles and responsibilities, process-centric or people-centric, high-risk mitigation or medium-risk mitigation, and daily meetings).

The number of projects in the data collected is 101, and each project is characterized by five criteria. The answers to the questionnaire come from various professionals around the world. The country that appears most often is Morocco (53%), followed by India (20%) [28]. Figure 2 shows the project management methods used in companies during the execution of their projects. Among these methods, there are Scrum, XP, Kanban, Hybrid Scrum Kanban, scaled agile framework (SAFe), and lean software development (LSD). The method that appears most often is Scrum followed by SAFe.

Table 1. Description of the project criteria used in the study

Criteria	Description
Project size	Define if the project is large or small.
Roles and responsibilities	Define if the team wants to define the roles and responsibilities of each member.
Process-centric	Define whether the team during the project is focused on the process.
People-centric	Define whether the team during the project focuses on people.
High-Risk Mitigation	Define whether the team wants high-risk mitigation during the project realization.
Medium Risk Mitigation	Define whether the team wants medium risk mitigation during the project realization.
Daily meetings	Define whether the team wants to meet daily to discuss problems encountered during the project.

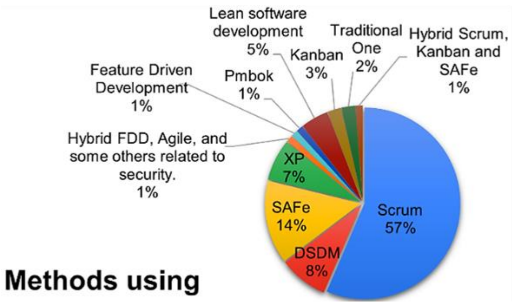


Figure 2. The percentage of methods used in the management of a project

4.2. Data pre-processing

During the first exploration of the database, the number of projects was 101, of which 19.80% were non-information technology specific or invalid projects. Our preliminary preprocessing serves to drop these invalid data. The result is a database holding 81 projects with missing values in various attributes.

Reduction of classes in the dataset: the resulting dataset holds as many as 11 methods as shown in Figure 2 that were chosen by managers; these choices present our objective (label). The first development of the model with 11 values to predict produced a very poor performance. This is mainly due to the number of classes (11) as well as to the minority of several classes (unbalanced data). To ensure a better performance while having interesting results, we went to a reduction of the number of classes based on the statistical results obtained in Figure 2. Based on the percentage of each method (1), we reduced the number of classes. Table 2 shows the percentage of each method in our dataset after the elimination of non-agile methods.

$$Percentage = \frac{absolute\ number}{81\ (number\ of\ projects)} \times 100 \quad (1)$$

Table 2. Percentage of each method in the dataset

Method (label)	Percentage
Scrum	59.3
SAFe	14.8
Dynamic systems development method (DSDM)	08.6
Extreme programming	06.2
LSD	04.9
Kanban	02.5
Hybrid Scrum, Kanban, and SAFe	01.2
Feature driven development (FDD)	01.2
Hybrid FDD, agile, and some others related to security	01.2

Based on Table 2 results, we found that the six methods with the highest percentage are Scrum (59.3%), SAFe (14.8%), DSDM (8.6%), XP (6.2%), LSD (4.9%), and Kanban (2.5%). Therefore, we decided to work with 4 classes initially and later with 7 classes. Once we determined the number of classes, we replaced the remaining methods with the “Others” class. Tables 3 and 4 display the percentages of each class after implementing the reduction method for both the 4 and 7 class cases.

Replacement of missing values: this is the first step to be done after obtaining the dataset. The dataset contains the previously mentioned missing values. To address this issue, a decision was made to replace the missing values with the characters “U” or the term “Undefined” to signify their undefined status.

Verification of the consistency of the data: after replacing rare and missing values, this step involves treating values that have the same meanings but are written in multiple ways, causing the system to perceive them as different. This step focuses on addressing these variations and ensuring consistency in the representation of these values. By standardizing the expression of similar meanings, the system can accurately interpret and analyze the data without any confusion or discrepancies.

Once this preprocessing was done, missing values were successfully reduced, noisy values were treated, and the projects resulted in the presence of several missing data. The study's main objective is to predict the most appropriate agile management method for a specific project. The selection process entails utilizing a prediction method, such as ID3, RF, or GBT, following the implementation of feature engineering for scenarios involving 4 and 7 classes.

Table 3. Percentage of each class after reduction to 4 classes

Method (Label)	Percentage
Scrum	59.3
Other	17.3
SAFe	14.8
DSDM	08.6

Table 4. Percentage of each class after reduction to 7 classes

Method (Label)	Percentage
Scrum	59.3
SAFe	14.8
DSDM	08.6
Extreme Programming	06.2
LSD	04.9
Other	03.7
Kanban	02.5

4.3. Feature engineering: weight through information gain

To calculate the weight of the attributes with respect to the class attribute, we used the information gain (2) according to Shannon entropy (3) [29], [30]. The attribute with the highest weight is considered the most relevant. Table 5 displays the calculated weights for each attribute in the scenario of 4 and 7 classes, showing that the “centric” attribute holds the highest significance in both cases.

$$Gain(S, A) = E(S) - \sum^{v} \left(\frac{|S_v|}{|S|} * E(S_v) \right) \tag{2}$$

Where S is a training set, A is the target attribute, S_v is the subset of elements whose attribute A value is v, $|S_v|$=the number of elements in S_v, and $|S|$=the number of elements in S.

$$E(S) = -\sum_{j=1}^{|S|} p(j)\, log_2\, p(j) \tag{3}$$

Where p(j) is the probability of having an element of characteristic j in the set S.

Table 5. The weight of the attributes in relation to the class attribute “method used”

Attribute	4 classes case		7 classes case	
	Weight	Approximate weight value	Weight	Approximate weight value
Roles & responsabilities	0.02092693192021855	0.021	0.12554480211152708	0.126
Daily meetings	0.04351192127755099	0.044	0.07302253270425707	0.073
Project size	0.0967436571845599	0.097	0.1252529219884546	0.125
Risk mitigation	0.17704992460790092	0.177	0.18290236270044402	0.183
Centric	0.191454575584443	0.191	0.24668972972636904	0.247

4.4. Decision tree models for choosing an agile management method for a project

In this study, all available attributes are initially selected for prediction. The optimal number of variables is not known as a priori. A feature selection process will allow for minimizing the system response time, thus improving the model performance. Once the dataset has been prepared by applying feature selection, the next step is the generation of decision tree models by applying the ID3, GBT, and RF algorithms in the case of 4 and 7 classes. Table 6 presents the parameters used for each model.

Table 6. Decision tree algorithm parameters

Algorithm	ID3	RF	GBT
Parameters	Information Gain Minimum size of split: 5 Minimum leaf size: 15 Minimum gain: 0.5	Gain ratio Number of trees: 100 Maximal dept: 10	Maximal depth: 5 Number of trees: 50 Learning rate: 0.01 Number of bins: 20 Min rows: 10 Split improvement: 1.0E-5

4.5. Results analysis and discussion

The dataset resulting from this study represents the different agile management methods currently used in companies. Nevertheless, the data is unbalanced, which leads to a weak performance during the first development of the model to be predicted with all the methods included in the dataset (label). Based on the statistical results obtained, i.e., the frequency of each method, we decided to reduce the number of classes into 7 classes at first and then into 4 classes in order to ensure better performance with attractive results. Table 7 shows the performance obtained after the application of ID3, GBT, and RF algorithms for both 4 and 7 classes cases.

Table 7. Performance comparison for 4 and 7 classes cases using the ID3, GBT, and RF algorithms

	4 classes	7 classes
ID3	60.42% (micro average: 60.49%)	55.56% (micro average: 55.56%)
GBT	60.42% (micro average: 60.49%)	54.44% (micro average: 54.32%)
RF	54.31% (micro average: 54.32%)	54.31% (micro average: 54.32%)

Distinct types of tree-based models were employed for both the 4-class and 7-class. The ID3 model served as a simple decision, while GBT encompassed a range of methods. RF, on the other hand, offered an enhancement by generating a set of decision trees and incorporating feature randomization. When applying the ID3 algorithm in the case of 4 classes, we found that the model gave an accuracy of 60.42% (micro average: 60.49%). We notice that Scrum is the most predictable class (accuracy of 75.61%) compared to the other methods. Otherwise, in the case of 7 classes, the result shows that the third model gives an accuracy of 55.56% (micro average: 55.56%). We notice that Scrum is the most predictable class (accuracy of 70.21%) compared to the other methods.

When applying the GBT algorithm in the case of 4 classes, we found that the fourth model gives an accuracy of 60.42% (micro average: 60.49%). We notice that Scrum is the most predictable class (accuracy of 66.67%) compared to the other methods. Otherwise, in the case of 7 classes, the result shows that the model gives an accuracy of 54.44% (micro average: 54.32%). We notice that Scrum is the most predictable class (accuracy of 61.54%) compared to the other methods.

When applying the RF algorithm in the case of 4 classes, we obtained that the model gives an accuracy of 54.31% (micro average: 54.32%). It shows that Scrum is the most predictable class (65.96% accuracy) compared to the other methods. Otherwise, in the case of 7 classes, the result shows that the model gives an accuracy of 54.31% (micro average: 54.32%). We notice that Scrum is the most predictable class (62.96% accuracy) compared to the other methods.

5. THE CHATBOT IMPLEMENTATION

5.1. Construction of the data set based on the decision tree model

To develop our dataset, we used the decision tree model improved by the results obtained in the earlier section. We, therefore, held on to the resulting decision tree in the case of 4 classes by applying the ID3 algorithm. This tree stands for a set of rules that we later transformed into a dataset shown in Table 8. To build a Chatbot, it is necessary to follow several steps, including identification of the intent and entity in a single user statement, implementation of a question-answer dialogue for a recognized intent, adding contextual information to responses when an entity is recognized, and implementing a multi-question process flows to satisfy a user’s query.

Table 8. Dataset of rules for choosing an agile project management method

Centric	Risk mitigation	Project size	Roles & responsabilities	Daily meetings	Method used
Process	Medium				Other
Process	High	Small			DSDM
Process	High	Large	U		Scrum
Process	High	Large	D	Defined	SAFe
Process	High	Large	D	Undefined	SAFe
People	Medium				Scrum
People	High	Small			Scrum
People	High	Large		Defined	Scrum
People	High	Large		Undefined	SAFe

Rasa stack is an open-source conversational artificial intelligence platform solution and machine learning framework that is widely used in large companies all over the world to supply the infrastructure to create Chatbots and virtual assistants [31]. Rasa is robust and flexible in enabling natural language understanding (NLU) and dialogue management. It allows for transparency, control, and the ability to integrate with existing systems.

The main components of Rasa are NLU which manages the classification of intentions, determining what comes next in a conversation, and Rasa core which enables the developing of a probability model that decides the set of actions to be performed [31]. NLU is a subset of natural language processing (NLP), which is considered the brain of Chatbots, as it processes raw data, analyzes it, cleans it, and then prepares to take the appropriate action [32]. Rasa uses long short-term memory (LSTM) which can be defined as a modified recurrent neural network (RNN) architecture that addresses the problem of evanescent and explosive gradients and solves the problem of training over long sequences and memory retention [33], [34].

The software architecture: with the aim of setting up our Chatbot solution for choosing the agile method to implement in a project. We used Python as a programming language. Rasa as an open-source machine learning framework for creating virtual assistants or Chatbots that use pipelines and NLP architecture. Flask as an open-source web development framework in Python.

5.2. Chatbot implementation

We started with the training data that is needed for a Chatbot. This is a list holding a set of messages that the user is still waiting for one of the relevant messages to receive from the bot. Furthermore, this data is annotated with "intent" and the entities that Rasa NLU needs to learn to extract. In our project, we created a dataset holding our decision tree model data for training. The general architecture of the project is shown in Figure 3 where the file named DATA_DT.xlsx contains the data we worked on.

Then, we need to configure the Rasa Pipeline after the training data is ready. The components and policies of our decision tree model to predict depending on user input are defined in the 'config.yml file'. The "language" and "pipeline" keys indicate the components used into the model to provide NLU predictions. The "policies" key specifies the policies that the model uses to predict the next action. To effectively reply to user messages, it is important to create "stories". Rasa stories are a form of training data used to train the dialogue management models of Rasa core.

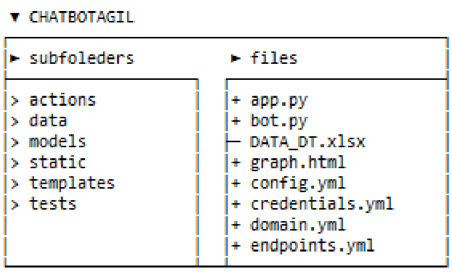


Figure 3. The project architecture

Once we have prepared the stories, it is time to prepare the Chatbot actions and the domain. The relevant files for this are 'actions.py' and 'domain.yml'. The first model has all the actions that the Chatbot must do, i.e., predict, after each message from the user, an action that the assistant must perform next. The second model specifies the intentions, entities, slots, responses, forms, and actions that the bot should be aware of. Also, it specifies a setting for conversation sessions.

Finally, after preparing everything for our Chatbot. It is necessary to start the training of our training data. This step is done using the command prompt. It allows the chatbot to learn and understand the messages received from the user (intentions) so that it can respond to them.

Our Chatbot is now able to understand the messages received from users and respond accordingly, utilizing our decision tree model. In Figure 4, the Chatbot poses a question to the user, inquiring about the desired type of process to follow and the level of risk mitigation. Users have the option to choose between medium and high levels, allowing for personalized interactions. Then, the Chatbot asks the question about the project size, large or small, and then depending on the user's choice, it chooses the agile method to follow.

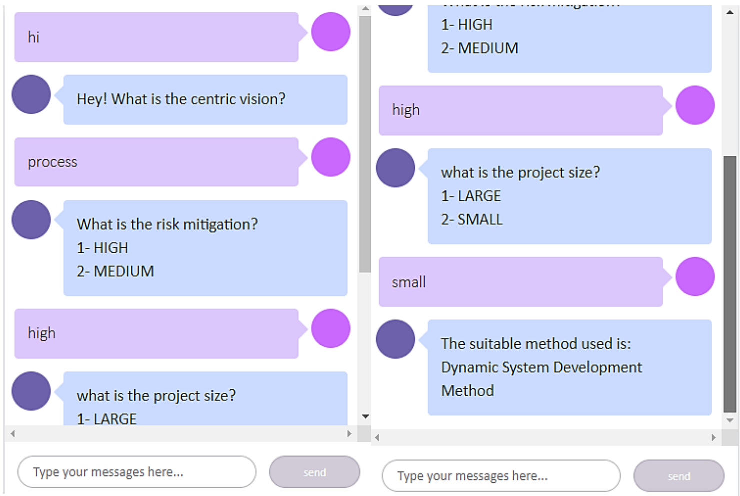


Figure 4. Chatbot's final response

6. CONCLUSION

The right choice of an agile method to manage a project positively affects the quality of the project, which involves customer satisfaction. This choice is based on a set of criteria extracted during the pre-project phase. Even though it is not easy to make this choice in the absence of a model that does this job as well as in the lack of the dataset. Therefore, our task consists of building our dataset using a questionnaire intended for experts in the field. Then this dataset was used to develop the decision tree model. The latter was developed by applying the ID3, RF, and GBT algorithms for cases of 4 and 7 classes due to the diversity of classes and the size of the dataset. Then, we made a comparison between these two cases in terms of performance. Eventually, we were able to develop an intelligent Chatbot that carries out our decision tree model for the 4-class case. This choice is based on a comparison of the performance. It can react to different questions in text form. In upcoming works, we plan to enhance our Chatbot by adding support for voice-based questions and improving the model's performance. This will be achieved by expanding the dataset size, enabling us to explore advanced deep-learning techniques. To validate our approach, we intend to collaborate with industry partners and conduct case studies. Furthermore, we aim to improve the different questions asked in order to consider other criteria that may be important in project management.

REFERENCES

[1] C. A. T. Romero, J. H. Ortiz, O. I. Khalaf, and W. M. Ortega, "Software architecture for planning educational scenarios by applying an agile methodology," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 8, pp. 132-144, Apr. 2021, doi: 10.3991/ijet.v16i08.20603.

[2] S. Merzouk, S. Elhadi, A. Cherkaoui, A. Marzak, and N. Sael, "Agile software development: comparative study," in *The Second International Conference on Smart Application and Data Analysis for Smart Cities*, 2018, pp. 1-6, doi: 10.2139/ssrn.3186323.

[3] M. K. Alqudah, R. Razali, and M. Kasim, "Agile methods selection model: a grounded theory study," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 357-366, 2019, doi: 10.14569/IJACSA.2019.0100749.

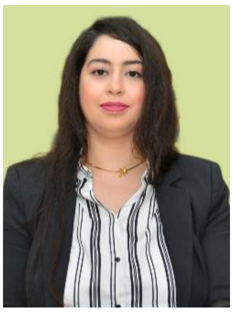
[4] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.

[5] Y. Rudnichenko, L. Liubokhynets, N. Havlovska, O. Illiashenko, and N. Avanesova, "Qualitative justification of strategic management decisions in choosing agile management methodologies," *International Journal for Quality Research*, vol. 15, no. 1, pp. 209–224, 2021, doi: 10.24874/IJQR15.01-12.

[6] R. Hanslo and M. Tanner, "Machine learning models to predict agile methodology adoption," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, Sep. 2020, pp. 697–704, doi: 10.15439/2020F214.

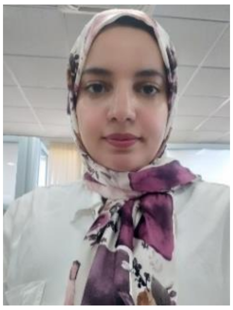
[7] F. Pouyandeh, M. Golabchi, and K. Taghizadeh, “Providing a model to choose the most appropriate agile method in construction projects,” *Proceedings of the Institution of Civil Engineers-Management, Procurement and Law*, vol. 176, no. 1, pp. 14–27, Jan. 2023, doi: 10.1680/jmapl.21.00033.
[8] N. Govil and A. Sharma, “Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors,” *Advances in Engineering Software*, vol. 172, Oct. 2022, doi: 10.1016/j.advengsoft.2022.103209.
[9] A. B. Simmons and S. G. Chappell, “Artificial intelligence-definition and practice,” *IEEE Journal of Oceanic Engineering*, vol. 13, no. 2, pp. 14–42, Apr. 1988, doi: 10.1109/48.551.
[10] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
[11] R. E. Neapolitan and X. Jiang, *Artificial intelligence with an introduction to machine learning*. Boca Raton: CRC Press, 2018.
[12] A. Burkov, *The hundred-page machine learning book*, Quebec City, Canada: Andriy Burkov, 2019.
[13] T. W. Edgar and D. O. Manz, “Machine learning,” in *Research Methods for Cyber Security*, Cambridge: Elsevier, 2017, pp. 153–173.
[14] B. Mahesh, “Machine learning algorithms-a review,” *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020.
[15] O. Z. Maimon and L. Rokach, *Data mining with decision trees: theory and applications*, Singapore: World Scientific Publishing, vol. 39, no. 3. 2015.
[16] T. Hamim, F. Benabbou, and N. Sael, “Survey of machine learning techniques for student profile modeling,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 4, pp. 136-151, Feb. 2021, doi: 10.3991/ijet.v16i04.18643.
[17] M. Thamarai and S. P. Malarvizhi, “House price prediction modeling using machine learning,” *International Journal of Information Engineering and Electronic Business*, vol. 12, no. 2, pp. 15–20, Apr. 2020, doi: 10.5815/ijieeb.2020.02.03.
[18] A. So, W. So, Z. Nagy, and O. M. C. Safari, “The applied artificial intelligence workshop,” Birmingham: Packt Publishing, 2020.
[19] F. J. M. Shamrat, R. Ranjan, K. M. Hasib, A. Yadav, and A. H. Siddique, “Performance evaluation among ID3, C4. 5, and CART decision tree algorithms,” in *Pervasive Computing and Social Networking: Proceedings of ICPCSN 2021*, pp. 127–142, 2022.
[20] C. Krauss, X. A. Do, and N. Huck, “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500,” *European Journal of Operational Research*, vol. 259, no. 2, pp. 689–702, Jun. 2017, doi: 10.1016/j.ejor.2016.10.031.
[21] P. Probst, M. N. Wright, and A. Boulesteix, “Hyperparameters and tuning strategies for random forest,” *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 3, pp. 1-19, May 2019, doi: 10.1002/widm.1301.
[22] K. B. Dasari and N. Devarakonda, “Detection of DDoS attacks using machine learning classification algorithms,” *International Journal of Computer Network and Information Security*, vol. 14, no. 6, pp. 89–97, Dec. 2022, doi: 10.5815/ijcnis.2022.06.07.
[23] F. O. Isinkaye, I. G. AbiodunBabs, and M. T. Paul, “Development of a mobile-based hostel location and recommendation Chatbot system,” *International Journal of Information Technology and Computer Science*, vol. 14, no. 3, pp. 23–33, Jun. 2022, doi: 10.5815/ijitcs.2022.03.03.
[24] B. P. Wicaksono and A. Zahra, “Design of the use of chatbot as a virtual assistant in banking services in Indonesia,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, pp. 23–33, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp23-33.
[25] R. Khan and A. Das, “Introduction to Chatbots,” in *Build Better Chatbots*, Berkeley, CA: Apress, 2018, pp. 1–11.
[26] N. K. Manaswi, “Developing Chatbots,” in *Deep Learning with Applications Using Python*, Berkeley, CA: Apress, 2018, pp. 145–170.
[27] S. Singh and H. Beniwal, “A survey on near-human conversational agents,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 8852–8866, Nov. 2022, doi: 10.1016/j.jksuci.2021.10.013.
[28] S. Merzouk, R. Gandoul, A. Marzak, and N. Sael, “Toward new data for IT and IoT project management method prediction,” *Mathematical Modeling and Computing*, vol. 10, no. 2, pp. 557–565, 2023, doi: 10.23939/mmc2023.02.557.
[29] B. Azhagusundari and A. S. Thanamani, “Feature selection based on information gain,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 2, pp. 18–21, 2013.
[30] C. Yin and J. Xi, “Maximum entropy model for mobile text classification in cloud computing using improved information gain algorithm,” *Multimedia Tools and Applications*, vol. 76, no. 16, pp. 16875–16891, Aug. 2017, doi: 10.1007/s11042-016-3545-5.
[31] M. Burtsev *et al.*, “DeepPavlov: open-source library for dialogue systems,” in *Proceedings of ACL 2018, System Demonstrations*, 2018, pp. 122–127, doi: 10.18653/v1/P18-4021.
[32] S. Raj, “Natural language processing for Chatbots,” in *Building Chatbots with Python*, Berkeley, CA: Apress, pp. 29–61, 2019.
[33] A. Nasser and H. AL-Khazraji, “A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 721–730, Feb. 2022, doi: 10.11591/ijece.v12i1.pp721-730.
[34] M. Tarsi, S. Douzi, and A. Marzak, “Predicting stock price using LSTM and social media dataset,” in *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, May 2023, pp. 1–4, doi: 10.1109/IRASET57153.2023.10152930.

BIOGRAPHIES OF AUTHORS



Soukaina Merzouk received her master’s degree in Information Science and Engineering from Hassan II University-Casablanca Faculty of Science Ben M'sik, Casablanca, Morocco, in 2016, and her Ph.D. degree in Computer Science from Hassan II University-Casablanca, Faculty of Science Ben M'sik, Casablanca, Morocco, in 2022. Currently, she is a professor of Computer Science at the Polydisciplinary Faculty of Sidi Bennour, Chouaib Doukkali University El Jadida, Morocco. Her research interests include agile software development, metamodeling, the internet of things, artificial intelligence applications, data mining techniques, big data, and machine learning. She can be contacted at email: merzouk.soukaina@gmail.com.

Soukaina Bouhsissin received a B.Sc. degree in Mathematical Sciences and Computer Science from the Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca Morocco in 2018 and an M.Sc. degree in data science and big data from Hassan II University, Morocco, in 2020. She is currently pursuing a Ph.D. in computer science. Her research interests include driver behavior classification, intelligent transport systems, machine learning, deep learning, image classification, and temporal series. She can be contacted at email: bouhsissin.soukaina@gmail.com.



Touria Hamim earned her Bachelor's degree in Engineering from the National School of Applied Sciences of El-Jadida in 2017. Following her passion for computer sciences, Touria pursued a Ph.D. in Computer Sciences at Hassan II University-Casablanca Faculty of Science Ben M'sik, Casablanca, Morocco, in 2021. Currently, Touria holds a key position as a responsible data scientist in a prominent telecommunications company, where she applies her expertise in machine learning, big data, and other advanced data science techniques to solve industry-specific challenges. She can be contacted at email: hamimxtouria@gmail.com.

Nawal Sael received her Ph.D. in Computer Science from Hassan II University-Casablanca Faculty of Science Ben M'sik, Casablanca, Morocco, 2013, and her engineer degree in software engineering from ENSIAS, Morocco, in 2002. Currently, she is a professor of Computer Science and a member of the Information Technology and Modeling Laboratory at Faculty of Science Ben M'sik, Casablanca, Morocco. Her research interests include software engineering, data mining, educational data mining, machine learning, the internet of things, and artificial intelligence. She can be contacted at email: saelnawal@hotmail.com.

Abdelaziz Marzak received his Ph.D. in Computer Science from Mohammed V University-Rabat, National School of Computer Science and Systems Analysis, Rabat, Morocco, 2000. Currently, he is a professor of Computer Science and director of the Information Technology and Modeling Laboratory at the Faculty of Science Ben M'sik, Casablanca, Morocco. His research interests include agile software development, metamodeling, the applications of artificial intelligence, software engineering, data mining, educational data mining, machine learning, and the internet of things. He can be contacted at email: marzak@hotmail.com.