



# Assessing the Effectiveness of Textual Recommendations in KoopaML: A Comparative Study on Non-Expert Users' ML Pipeline Development

Pablo Antúnez-Muiños, University Hospital of Salamanca, Spain


 <https://orcid.org/0000-0003-0573-4321>

Pablo Pérez-Sánchez, University Hospital of Salamanca, Spain

Andrea Vázquez-Ingelmo, GRIAL Research Group, University of Salamanca, Spain\*


 <https://orcid.org/0000-0002-7284-5593>

Francisco José García-Peñalvo, GRIAL Research Group, University of Salamanca, Spain


 <https://orcid.org/0000-0001-9987-5584>

Antonio Sánchez-Puente, University Hospital of Salamanca, Spain


Víctor Vicente-Palacios, Philips Clinical Science, Spain

 <https://orcid.org/0000-0003-1895-7436>

Alicia García-Holgado, GRIAL Research Group, University of Salamanca, Spain

 <https://orcid.org/0000-0001-9663-1103>

P. Ignacio Dorado-Díaz, University of Salamanca, Spain

 <https://orcid.org/0000-0002-8344-6328>

Jesús Sampedro-Gómez, CIBERCV, Spain

Ignacio Cruz-González, University Hospital of Salamanca, Spain

Pedro L. Sánchez, University of Salamanca, Spain & University Hospital of Salamanca, Spain

## ABSTRACT

Artificial intelligence (AI) integration, notably in healthcare, has been significant, yet effective implementation in critical areas requires expertise. KoopaML, a previously developed visual platform, aims at bridging this gap, enabling users with limited AI knowledge to build ML pipelines. Its core is a heuristic-based ML task recommender, offering guidance and contextual explanations. The authors compared the use of KoopaML with two non-expert groups: one with the recommender system enabled and the other without. Results showed KoopaML's intuitiveness benefits all but emphasized that textual guidance doesn't substitute for fundamental ML understanding. This underscores the need for educational components in such tools, especially in critical fields like healthcare. The paper suggests future KoopaML enhancements include educational modules, making ML accessible, and ensuring users develop a solid AI foundation. This is crucial for quality outcomes in sectors like healthcare, leveraging AI's potential through enhanced non-expert user capability.

## KEYWORDS

Artificial Intelligence, HCI, Health Platform, Information System, Medical Data Management, Think Aloud Protocol, Usability

DOI: 10.4018/IJSWIS.340727

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

When using AI to automate tasks in sensitive fields such as healthcare, if the users are unaware of these factors, they can reach wrong conclusions, originate discrimination, or even commit medical negligence (Weyerer & Langer, 2019; Ferrer et al., 2021; Hoffman, 2021; Wachter et al., 2020). For this reason, it is crucial to understand the algorithms' inputs and outputs and the implications they have in the performance of AI, especially in the medical context where AI methods are being employed for complex tasks like diagnoses, disease detection, segmentation, assessment of organ functions, etc. (González Izard et al., 2020; González Izard et al., 2018; Litjens et al., 2017). In previous work, we tackled this problem through KoopaML (García-Holgado et al., 2022; García-Peñalvo et al., 2023; Vázquez-Ingelmo et al., 2021), a platform developed to educate and support inexperienced users in developing machine learning (ML) flows. KoopaML has two goals: (1) providing a simple interface to allow the use of ML techniques visually, and (2) offering a learning experience to its users, helping them understand the rationale of these techniques. However, providing these features is not straightforward, and real-world feedback is required to look.

The novelty of our approach lies in the integration of a heuristic-based ML task recommender within KoopaML, a platform designed to demystify the complexities of machine learning for non-experts. This innovative feature is specifically tailored to guide users through the selection and application of ML techniques, fostering an educational environment absent in existing platforms. By focusing on the unique needs of medical professionals, KoopaML stands out for its user-centric design that not only simplifies the ML workflow but also embeds learning within the process. This dual focus on usability and educational value represents a significant advancement in the field, as it directly addresses the gap between the growing demand for AI in healthcare and the current barriers to its practical application by frontline medical practitioners.

In the latest iteration of our research, we conducted a user evaluation of KoopaML with a group of physicians at the University Hospital of Salamanca. This assessment was geared toward three principal aims: firstly, to evaluate the platform's heuristic-based ML task recommender—one of KoopaML's cornerstone features; secondly, to delve into the perceptions and challenges faced by medical professionals, who are non-expert users, when interacting with AI technologies; and thirdly, to explore whether textual guidance within the platform can simplify ML concepts for users unfamiliar with the field. In this context, a critical research question emerged: Can KoopaML, with its recommender system and textual guidance, make the journey into machine learning an approachable and more user-friendly experience for medical professionals?

A key component of this evaluation was to scrutinize the practicality and effectiveness of the x system within KoopaML. Assessing its impact on the user experience is essential for refining the platform's functionality and advancing our overarching vision: to create a tool that not only simplifies the design and operationalization of ML models but also imparts valuable knowledge to its users throughout the process. By undertaking this evaluation, our objective is to shed light on the specific hurdles physicians may face as they attempt to integrate ML algorithms into their clinical practice. Understanding these challenges is crucial for developing targeted improvements in KoopaML that will support medical practitioners in harnessing the power of AI, thereby enhancing patient care and advancing the healthcare industry's technological frontiers.

The rest of this work is organized as follows: "Related Works" explores related works on tools for creating ML pipelines. "Background" provides the KoopaML platform's background, outlining its features and detailing the recommendation process. "Methodology" describes the methodology followed for the platform's assessment. "Results" presents the study results, while "Discussion" discusses them both in general and from the medical point of view, including the identified limitations. Finally, "Conclusions" outlines the conclusions reached through this study.

## RELATED WORKS

A variety of tools designed to assist in machine learning processes have emerged over time. For this research, we identified and focused on three distinct types of tools. The first category includes tools for developers and data scientists, providing comprehensive programming libraries to facilitate the development of machine learning applications. TensorFlow, for example, is a machine learning framework that runs at scale and in diverse contexts (Abadi et al., 2016), assisting academics in pushing state-of-the-art models in ML and developers in simply building and deploy (ml)ing ML-powered apps. Apache Mahout, a library for scalable machine learning on distributed dataflow systems, is another example (Anil et al., 2020). In this category, we may also add Python libraries like PyTorch, Scikit-learn, or Keras.io, as well as cloud services like Google Colab, which is a serverless Jupyter notebook environment (Bisong, 2019).

Second, there are systems aimed at specialists while still providing tools for non-specialist users. These applications provide visual environments that aid in the visual development of machine learning models. Weka, for example, is a library of machine-learning techniques for data mining jobs. It features four environments, the most notable of which is Knowledge Flow, a visual interface that allows users to describe a data stream by visually linking components representing data sources, preprocessing tools, learning algorithms, assessment techniques, and visualization tools (Frank et al., 2009; Hall et al., 2020). RapidMiner Studio is a data science platform that includes tools for creating ML processes. It includes the Visual Workflow Designer tools for creating ML processes, and every step is documented for total transparency. This feature enables data source connection, automatic in-database processing, data visualization, and model validation processes (Bjaoui et al., 2020). KNIME Analytics Platform is another example. It gives tools for constructing visual data analytics workflows with a graphical interface that does not require scripting. KNIME is a modular platform that allows for the simple visual building and interactive execution of data pipelines (Berthold et al., 2009).

ML has also begun to be introduced in elementary and high schools. This has led to the creation of tools to assist non-expert users, such as children, in performing simple ML tasks using a visual interface. Two instruments can be highlighted in this category. The first is LearningML (Rodríguez-García et al., 2021), which is a platform for developing computational thinking abilities through hands-on AI projects, and second is Machine Learning for Kids (<https://machinelearningforkids.co.uk/>). Both tools are built around a primary pipeline for training models and a Scratch integration for using the trained model.

There are several sophisticated apps aimed at simplifying the use of ML algorithms, as well as instructional resources for comprehending these complex procedures. However, these platforms are designed to cater to a wide audience but often lack the specialized features necessary to address the unique requirements and challenges of specific domains. Against this backdrop, KoopaML emerges as a specialized visual ML platform that not only simplifies the creation and use of ML models for medical practitioners but also places a strong emphasis on the educational aspect of its use. It stands apart by integrating health-related criteria into its functionality, thus providing a more relevant and context-aware experience for users in the health domain. Moreover, KoopaML is dedicated to imparting an educational experience. It is structured to guide users through the underlying principles of ML techniques as they apply them, thereby fostering a deeper understanding of the rationale behind their choices. This contrasts with many existing solutions that may facilitate mechanical application of ML algorithms but fall short in educating users about the intricacies of model selection, data preprocessing, and algorithmic decision-making.

## BACKGROUND

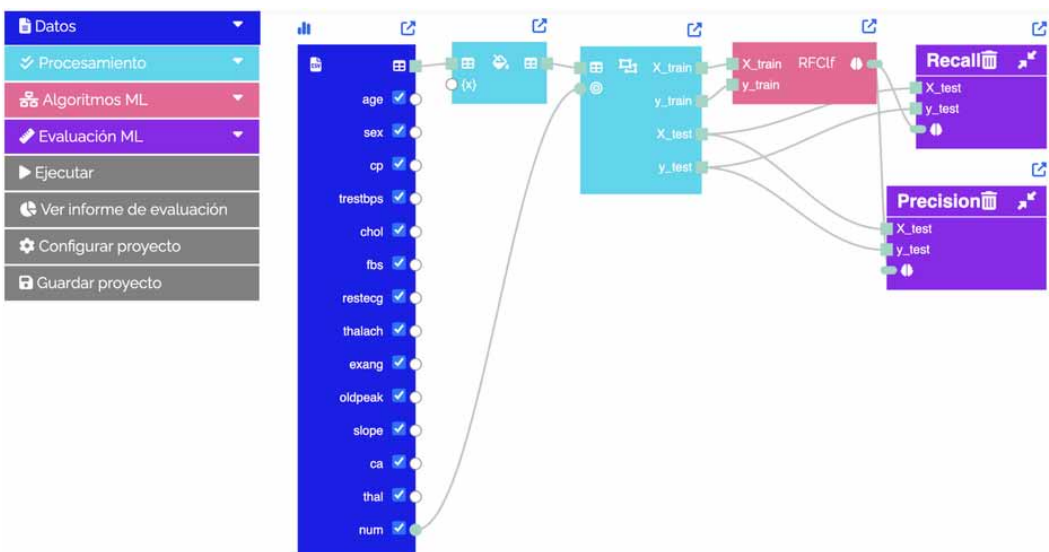
### KoopaML Description

KoopaML is a web application that allows users to create machine learning pipelines through visual nodes and simple interactions with the interface (García-Peñalvo et al., 2023). Each visual node is associated with specific tasks like data uploading processes, data cleaning functions, machine learning models, and metrics for performance evaluation.

Figure 1 displays the workspace of KoopaML, in which users can drag and drop different nodes and configure them to create the ML pipeline. On the left side of this figure, there is a menu containing the different types of nodes that can be included in the pipeline. Each node is detailed in the results section. On the right side of Figure 1, there is an ML pipeline that uploads a dataset, fills its null values, splits it into a train and a test dataset, trains a random forest classifier, and calculates the recall and precision of the model. Once users have finished their pipeline, they can execute it by clicking the “Run” button, triggering the backend to build the pipeline by connecting tasks using the algorithm presented by García-Peñalvo et al. (2023) and the SciLuigi (<https://github.com/pharmbio/sciluigi>) library. SciLuigi is a wrapper for Spotify’s Luigi Python library (<https://github.com/spotify/luigi>), and it supports dynamic workflows, avoiding hard-coded dependencies (Lampa et al., 2016; Lampa et al., 2019) and increasing flexibility and scalability of the system. Moreover, by using the SciLuigi library, users can also consult the intermediate results of each task to debug, improve, or check any issue related to the pipeline.

KoopaML has been developed through paradigms and technologies with a focus on flexibility. The emphasis on designing a flexible architecture aims at enabling expert users to expand the platform’s capabilities by adding custom algorithms, components, or novel heuristics to assist in defining ML pipelines without significant development efforts. For a complete and detailed description of the system architecture, please refer to the work by García-Peñalvo et al. (2023).

Figure 1. Example of a Workflow Developed in KoopaML



## Recommendations and Heuristics Management

One of the most interesting features of KoopaML is the heuristics management module. As explained in the introduction, KoopaML is focused on providing a learning experience while developing ML workflows. The heuristics management module has been designed and introduced in the platform to address this learning experience. This module offers an interface to define flexible and modifiable heuristics used in the workspace to yield recommendations regarding which elements are necessary in the pipeline. The heuristics are represented through the Domain-Specific Language (DSL) provided by the flowchart.js (<https://github.com/adrai/flowchart.js>) library. The flowcharts.js library allows textual and graphical representation of flow charts, which is extremely useful in this context, as it provides a visual and direct manipulation of rule-based recommendations (see Figure 2).

The heuristics module is only accessible to the system's administrators and AI experts. This module yields recommendations previously defined by using the DSL. For this reason, only users with permissions can edit the heuristics using the interface presented in Figure 3. Once the heuristics have been saved, the interface will start yielding recommendations based on the state of the pipeline and the current tasks and connections in the workspace. Whenever the user performs an action on the workspace and changes its state, the heuristics are checked to update the system's recommendation.

To sum up, the recommendations in our system are derived from the predefined heuristics, which are essentially rule-based. This approach eliminates the need for intensive computational processes like real-time data analysis or complex machine learning computations. As a result, it enables the system to provide prompt and efficient responses without the burden of heavy computational demands.

Figure 2. Excerpt of the Definition of Heuristics and Recommendations

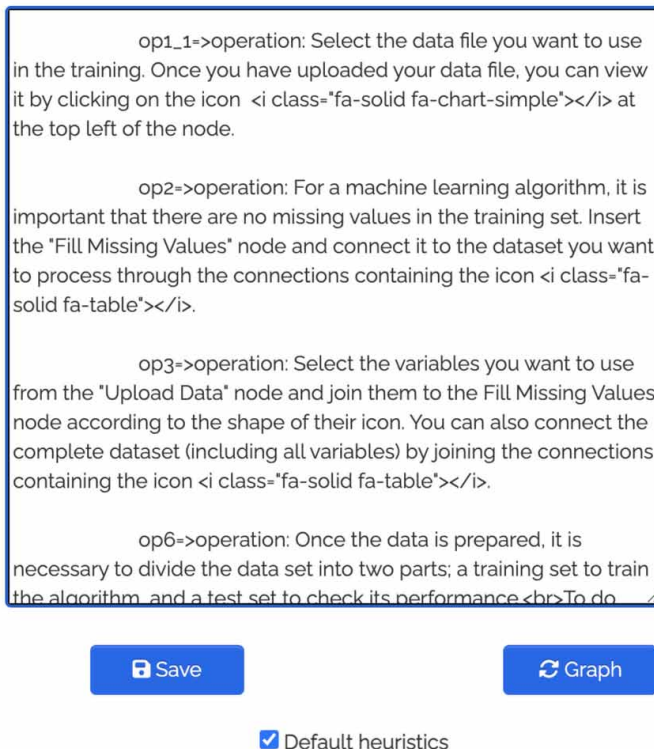
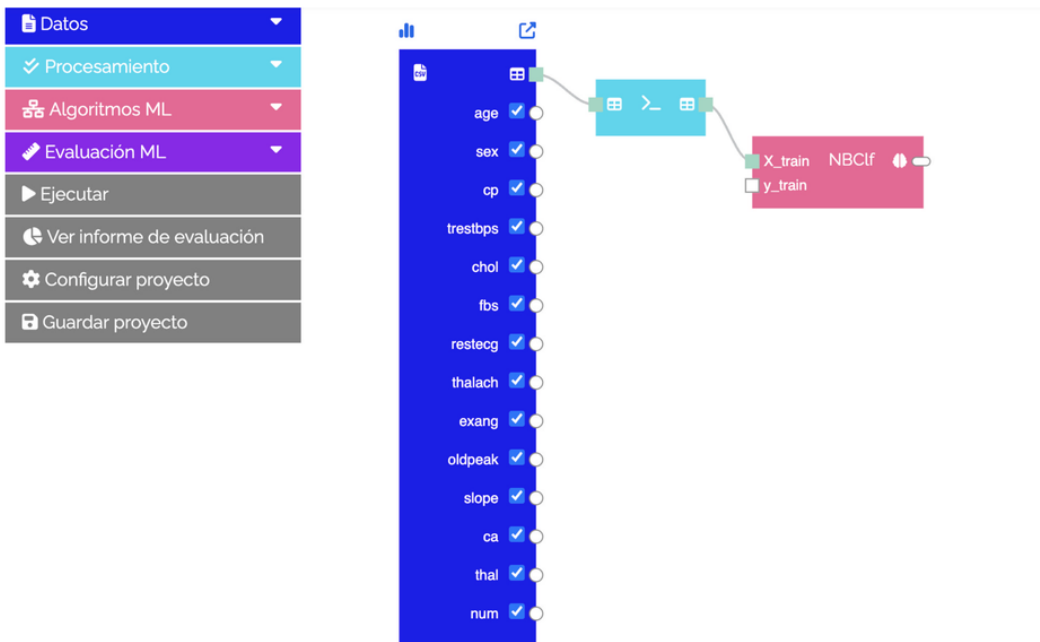



Figure 3. Recommendation of the Platform After Uploading a New Dataset (See Grey Modal at the Bottom)



💡 For a machine learning algorithm, it is important that there are no missing values in the training set. Insert the "Fill Missing Values" node and connect it to the dataset you want to process through the connections containing the icon .

## MATERIALS AND METHODS

### Methodology

The think-aloud method (Gambier & Doorslaer, 2010; Someren et al., 1994) has been followed to carry out the usability test of KoopaML. The think-aloud method, employed in our usability test of KoopaML, offers several distinct advantages. Primarily, it provides real-time insights into users' thought processes, which is invaluable in identifying not only the difficulties they face but also their problem-solving strategies while interacting with the system. This method enables us to gather nuanced data on user experience, which goes beyond mere functionality and delves into user satisfaction and engagement with KoopaML. Importantly, the flexibility of the think-aloud method, which can be carried out online through video conferencing tools, aligns with the constraints faced by our target users—physicians. Their tight schedules often preclude the possibility of in-person testing, and the adaptability of this method to remote settings via video conferencing tools is a crucial factor in ensuring their participation. Finally, by using this approach, it is also possible to capture the users' interaction during the execution of the demanded tasks and map them with the users' thoughts and performance, thus obtaining a complete landscape of usability. In this sense, we have also integrated a digital data analysis tool that captures the behavior of users within websites, namely Hotjar.js (<https://www.hotjar.com/>), to track every interaction performed by the users.

A total of eight physicians (four female and four male) participated in the study. Although small, the sample is appropriate for the selected method. As Nielsen (1994) suggests, planning think-aloud

protocols with  $4 \pm 1$  subjects yields sufficient information to analyze the participants' problem-solving issues. Table 1 outlines the specialization and background in ML of each participant. A high background in ML means the participant has worked and used algorithms programmatically, while a moderate level means the participant has used ML superficially (use of graphical platforms, for example). On the other hand, participants deemed low have limited ML knowledge are without having used ML and without having a background in ML, and none indicates they do not know anything about ML.

Considering their background with ML, each participant was assigned, in a balanced way, to use one of the two variants of the platform: the baseline version, without the recommender feature, and the new version, with the recommender feature. This assignment will support comparisons between the two versions and an analysis to determine if the recommender feature is helpful for users.

### Study Protocol

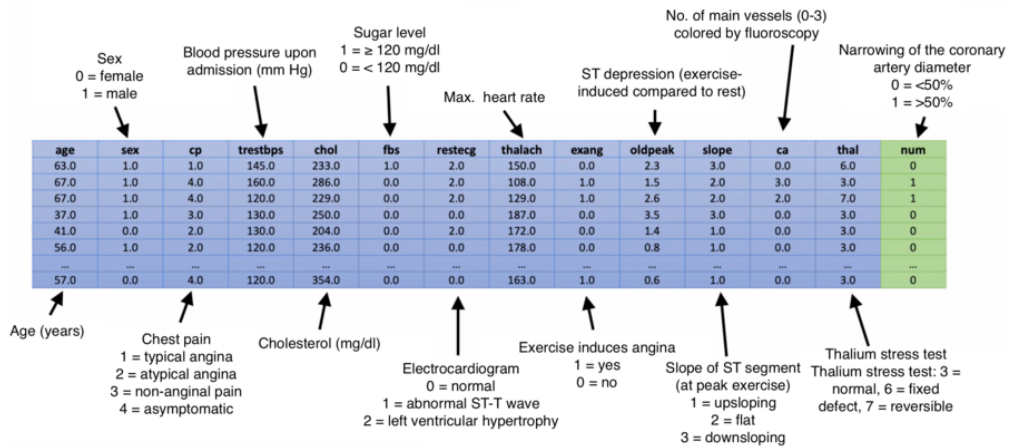
This section details the study protocol followed during the interviews. An online meeting through a videoconferencing platform was arranged for each participant once they confirmed their participation. In each meeting, the following steps were carried out:

1. The participant was introduced to the study and was asked to share their screen.
2. An introductory video<sup>1</sup> to ML and the KoopaML platform was played. The video contained the basic notions of ML and a quick tutorial on the platform's main features. Specifically, the video outlined basic concepts of ML including the structure of the input data, the notion of train and test sets, the difference between classification and regression, and the steps needed in an ML pipeline (e.g., data preparation, model training, and model validation).
3. Question-and-answer (Q&A) about the video and study protocol was conducted. The purpose of this Q&A is twofold: firstly, to resolve any ambiguities or confusion regarding the platform's functionality and features, and secondly, to clarify the objectives and procedures of the study itself.
4. The participant was given access to KoopaML. Each participant in the study was provided with their own set of credentials to access KoopaML. This approach was deliberately chosen to prevent any overlap or interference with the projects and results of other users. By ensuring each participant had a unique login, we maintained the integrity of the data collected and allowed for a more accurate analysis of user interactions and outcomes.
5. The study's database, formatted as a CSV file, was made available in the chat for participants to download and use on their computers. The dataset structure and its variables are explained in Figure 4.

**Table 1. Participants of the Study and Their Backgrounds**

ID	Area	ML Background	KoopaML Setting
P1	Medical resident	High	No recommender
P2	Medical staff	Moderate	Recommender
P3	Research nurse	Low	Recommender
P4	Medical resident	Low	No recommender
P5	Medical resident	Low	No recommender
P6	Medical staff	None	Recommender
P7	Medical resident	None	No recommender
P8	Medical resident	None	Recommender

Figure 4. Input Data Description



6. Finally, the tasks to be carried out during the session were also shared via chat. To ensure participants had constant access to their instructions and could easily reference them throughout the study, we pinned the task list in the chat. This strategic placement aimed to prevent any confusion or forgetfulness regarding the tasks at hand, thereby facilitating a smoother user experience and minimizing the need for participants to seek clarification or assistance.

The chosen database is a well-known open repository (Dua & Graff, 2019). This database has been used in multiple studies to create ML models (Detrano et al., 1989; Gennari et al., 1989). It includes 303 patients with suspected ischemic heart disease. Up to 14 features, including baseline characteristics like demographic or cardiovascular risk factors, were included. Indeed, results from the non-invasive tests were collected. Finally, results from the coronary angiography were used to confirm the presence or absence of coronary artery disease.

Regarding the activities to be carried out by the participants, two main tasks were indicated (participants could switch to second task (T2) if they got stuck with the first one): The first task (T1) included the following sub-tasks:

- **T1.1.** Load the data.
- **T1.2.** Visualize the data.
- **T1.3.** Split the dataset to use 80% of the observations as a training set and 20% as a test set (train 80%/test 20%).
- **T1.4.** Train a logistic regression model.
- **T1.5.** Obtain ROC AUC results of the trained model.

On the other hand, the second task (T2) was composed of the following sub-tasks:

- **T2.1.** Load the data.
- **T2.2.** Visualize the data.
- **T2.3.** Connect an input to a “Missing Values” node.
- **T2.4.** Split the dataset to use 80% of the observations as a training set and 20% as a test set (train 80%/test 20%).
- **T2.5.** Train a random forest model.
- **T2.6.** Obtain recall and precision results of the trained model.



Firstly, all the physicians tested were working at that time in the Cardiology Department of the University Hospital of Salamanca, Spain. As most of the participants had barely any knowledge about AI or ML, the selected database permitted them to use at least familiar features that would facilitate the tasks scheduled.

The study aims to analyze the effect of KoopaML's recommendations on the understandability of ML concepts. For this purpose, the protocol explained above was carried out. The use of the database in the study was conditioned by the study participants. The participants were working at the time in the cardiology department. Therefore, the chosen database presents a prediction problem in the field of cardiology. Thus, the familiarity of the variables to be studied with the participants did not pose any extra problems for the understanding of the database. On the other hand, different ML models for prediction or regression were integrated into the KoopaML platform. In the explanatory video before the study, all participants were told they had to use prediction models. Given the participants' bare knowledge of ML models, the algorithm to be used was explicitly indicated in each task, so participants did not have to analyze the problem to choose the best algorithm (out of the scope of this evaluation). Logistic regression and random forest models were chosen, as they are widely used in classification tasks. To calculate the predictive power of the algorithms in each exercise, participants were also asked for common metrics in the health domain. Metrics such as ROC AUC, recall, or precision are used in medicine for other techniques such as survival analysis, so the participants were familiar with these concepts.

The study aims to validate the use of the recommendations in the KoopaML platform. For this purpose, the protocol explained above was carried out. At the time of the study, all participants were part of a cardiology department in a Spanish public hospital. To facilitate familiarization with the task, a simple database was chosen, by the clinical knowledge of the group studied. The familiarity with the variables to be studied with the participants did not pose any extra problems for the understanding of the database. On the other hand, the KoopaML platform integrates different ML prediction and regression models. Before the study, participants were informed they should use a prediction model to develop the tasks. The choice of two specific algorithms was aimed at observing how user-friendly it was for participants to locate the algorithms on the platform. The choice of specific metrics for the evaluation of the models was chosen because of their use in medicine for other techniques such as survival analysis.

Finally, the screen and voice were recorded for each participant during the whole study to analyze their actions and comments subsequently.

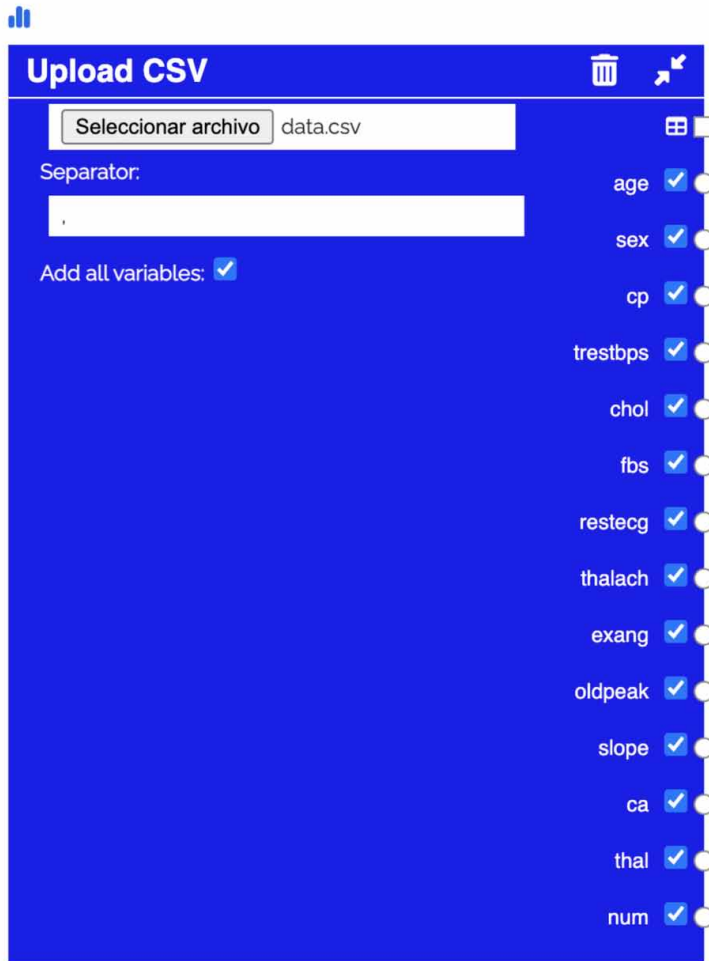
## RESULTS

The results are structured according to the tasks outlined in the previous section. The sub-tasks covered are data uploading and visualization, train/test partitioning, algorithm, and validation. These sub-tasks correspond with the available visual nodes in the KoopaML workspace interface and are outlined in each sub-section. Finally, other aspects of the tasks, such as the connections between nodes or the filling in of missing data, among others, are analyzed.

### Data Uploading and Visualization

The data uploading node (see Figure 5) allows users to upload and visualize data. The node contains an input to explore the file system and upload CSV files. Once the dataset is uploaded, users can configure certain aspects, like the separator of the CSV file, and determine if all the variables should be considered in the dataset (they can be ignored by unchecking them). The columns of the CSV file are displayed in the rightmost part of the node, and they can be connected to other nodes individually (through the circular sockets or through the square socket on top). In addition, the dataset can be inspected through an information dashboard by clicking on the icon in the top-left corner.

Figure 5. Data Uploading Node

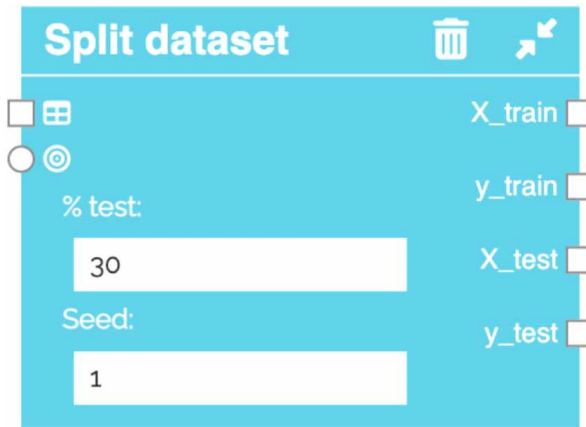


All participants were able to upload the training data properly, completing T1.1 successfully. Although this sub-task was executed correctly, some participants had problems finding and adding the “data upload” node, spending significant time during this sub-task. On the contrary, most of the participants did not complete T1.2. Some participants (3 out of 8) managed to visualize data through the dashboard provided by the platform, but even visualizing the data, all participants had problems understanding the displayed information. T2 had similar results as task T1. All participants correctly performed the data upload (T2.1). In this task, T2.2 was only achieved by the participants who managed to accomplish it also in T1.

### Train/Test Partitioning

The data partitioning node (see Figure 6) is focused on obtaining test and training datasets to prepare the data for the subsequent ML algorithms. This is done to evaluate the model’s performance objectively. The model learns from the training set and is then tested on the unseen data of the test set to gauge its accuracy and generalizability. This node has two inputs (on the left side): the original dataset (on

Figure 6. Data Partitioning Node



top), and the target variable (at the bottom). The percentage of rows employed for the test dataset can be configured through the input “% test.” A seed for this process can also be configured.

The “split dataset” node provides four outputs (on the right side of the node) that can be connected to subsequent nodes: the training and test features ( $X_{train}$  and  $X_{test}$ , respectively) and the training and test labels ( $y_{train}$  and  $y_{test}$ , respectively) taking into account the dataset and target variable specified in the inputs.

During T1, data partitioning (T1.3) was successfully performed by 5 of the 8 participants. One of the participants skipped this section when performing the task, going directly to the algorithm task. The rest of the participants completed this task but with different difficulties:

- Two of the participants added the node correctly but were not able to link the variables properly. After several attempts, they managed to do it.
- One of the participants was not able to find the node. Eventually, they got it on the dashboard but could not locate it in the workspace. Once found, they were able to link the variables correctly.
- Two of the participants forgot to add this node and went directly to the classification model. Later, they realized they did not add the data division node and could then add it and connect it properly.

During T2 (T2.4), there was a considerable improvement compared to the first one:

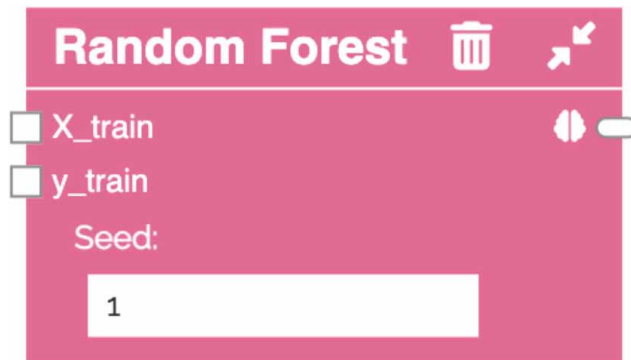
- Seven of the participants succeeded in adding the node.
- Three of the participants performed this step correctly.
- Two of the participants made the connection successfully but removed the selection of the variables in the data uploading node.

## Algorithm

The nodes related to ML algorithms have a common structure (see Figure 7). In this case, the inputs are the previously obtained training features ( $X_{train}$ ) and labels ( $y_{train}$ ). The result of executing this node is the trained ML model, which can be downloaded as a pickle object and subsequently connected to evaluation nodes. As with the data partitioning node, a seed can be configured.

In T1, participants are asked to fit a logistic regression (T1.4). However, although it had been indicated that a classification task would be performed in both tasks, most participants (6 out of 8)

Figure 7. ML Algorithm Node for a Random Forest Classifier



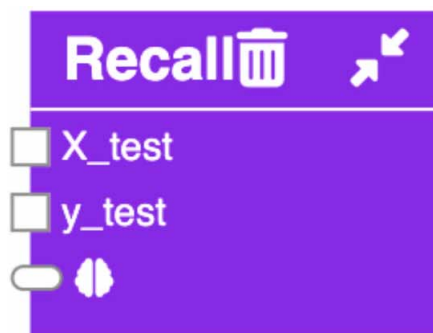
had problems understanding how to tackle this activity. The initial intuition of most of the participants who failed on this point (5 out of 6) was to search for the node in the regression section. This was a misconception compounded by the repetition of a specific term, in this case, “regression.” In this sub-task, there was no difference between the group that received recommendations and the group that did not. In contrast, in T2 (T2.5), all participants managed to find the random forest node without much difficulty. However, while participants ended up finding the right node for the sub-tasks T1.4 and T2.5, they were not able to train the models. During the think-aloud evaluation, we detected their confusion about what they needed to do to “use” the model, even from those participants who relied on the system’s recommendations.

### Validation

Finally, validation nodes also have the same structure (see Figure 8); they only differ in the metric computed. These nodes receive the training features and labels ( $X_{test}$  and  $y_{test}$ ) and the trained model obtained from a previous algorithm node (represented by the brain icon, at the bottom of the node) as input.

The validation task of both tasks involving validation (T1.5 and T2.6) was only performed correctly by one participant (1 out of 8). However, the reasons for this are different across participants. In most cases (5 out of 8), errors accumulated in previous steps led to the validation sub-tasks not being completed. The rest of the errors were caused by difficulties in finding the specific validation nodes (2 out of 8) or by problems in connecting the preceding nodes to the validation node (1 out of 8). These errors were related to the misunderstanding of participants about the metrics and lack of

Figure 8. Validation Node to Compute the Recall of a Classification Model



knowledge regarding the validation processes, as they explicitly verbalized during the execution of these validation sub-tasks.

Finally, like in T1.4 and T2.5, no notable differences were found between the groups that received recommendations and the groups that did not.

### Final Comments and Summary

The results displayed in Table 2 and Figure 9 underscore the nuanced relationship between the presence of a recommender system and the proficiency of participants with differing levels of machine learning knowledge. A closer examination of task completion rates and issues encountered reveals some patterns and outliers. Firstly, participants without access to the textual recommender (NR) exhibited varied performance, with some completing all tasks successfully and others struggling significantly. Notably, participant P1, with no recommender, achieved a perfect task completion rate, which could be influenced by their high level of ML background. Conversely, participants P4, P5, and P7, also without a recommender, had lower completion rates, particularly for Task 1 (T1), indicating potential areas where the platform’s usability could be enhanced or where additional instructional support may be needed.

Among those with the recommender system (R), participant P2’s improvement from T1 to T2 suggests the recommender may have a delayed benefit, possibly aiding in learning or adaptation over time. However, this pattern was not consistent across all participants with access to the recommender, as P6 and P8’s performances did not show the same level of improvement.

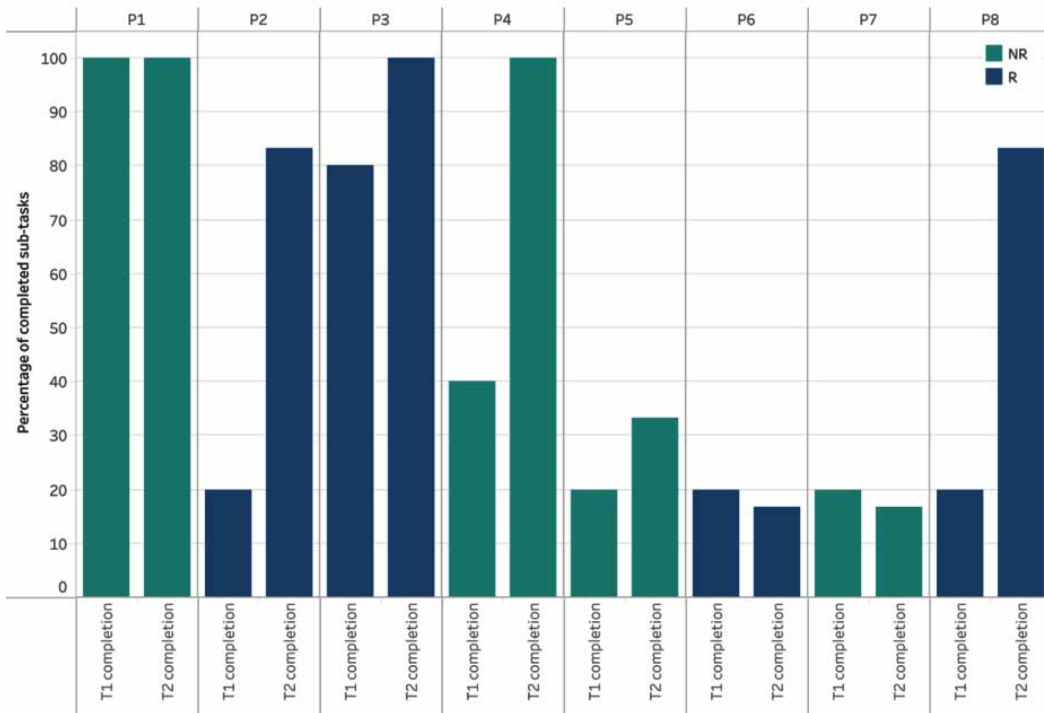
The issues column brings attention to specific sub-tasks where participants encountered difficulties. The recurring issue at T1.3, across both NR and R settings, indicates a common stumbling block in the pipeline creation process, which warrants further investigation. The additional challenges faced by participants P5, P6, P7, and P8 during T2.4 suggest this sub-task may be particularly complex or insufficiently supported by the current platform design or the recommender system. In light of these findings, it is apparent that the recommender system’s influence on task performance is not straightforward and may be affected by individual user differences, such as prior ML knowledge. Moreover, the overall platform design and user interface may play a significant role in the successful completion of tasks, irrespective of the presence of a recommender system.

**Table 2. Summary of Task Completion Rates and Issues (Participants Are Ordered by Knowledge/Experience in Machine Learning)**

ID	Setting	T1 Completion	T2 completion	Issue
P1	NR	5/5	6/6	-
P2	R	1/5	5/6	T1.3
P3	R	4/5	6/6	T1.3
P4	NR	2/5	6/6	T1.3
P5	NR	1/5	2/6	T1.3 – T2.4
P6	R	1/5	1/6	T1.3 – T2.4
P7	NR	1/5	1/6	T1.3 – T2.4
P8	R	1/5	5/6	T1.3 – T2.4

Note. The “Setting” column refers to the recommender setting; NR = No recommender; R = Recommender. The completion of the tasks (“T1 and T2 completion”) refers to the number of nodes correctly executed in the pipeline and not only to the number of nodes correctly added to the workspace. The “Issue” column refers to the sub-task in which participants were not able to continue.

Figure 9. Completion Rate of Participants in the Two Proposed Tasks (The Color Encodes the Recommender Setting: NR = No Recommender; R = Recommender)



## DISCUSSION

The results are structured according to the tasks outlined in the previous section. The sub-tasks covered are data uploading and visualization, train/test partitioning, algorithm, and validation. These sub-tasks correspond with the available visual nodes in the KoopaML workspace interface and will be outlined in each sub-section.

### General Discussion

The goal of this study was focused on understanding the problems and difficulties users with different ML expertise present when introduced to ML and to test if textual recommendations could alleviate the complexity of ML concepts to non-expert users. Following the obtained results, as noted in Table 2, where participants with higher expertise performed better, it is clear to conclude that facing ML without a strong background is highly complex. Although some sub-tasks were accomplished, most of the participants were not able to complete the two proposed tasks successfully. Although half of the participants had support from textual recommendations to guide the pipeline construction, there were no significant differences in performance between the two groups. Moreover, some participants asked how to close the “dialog box at the bottom of the screen.” While these textual recommendations provided all the necessary actions to carry out the two proposed tasks successfully, participants ended up ignoring them. This could mean that once read, the recommendation was not useful for them anymore, and they were not aware the dialog would update with every change in the workspace. Participants did read the errors shown in the dialog box displayed after running the pipeline in case of an unsuccessful execution because it drew more attention than the recommendations dialog box. In Figure 10, which shows the interaction heatmap of one of the participants with the

Figure 10. Heatmap Indicating Interactions of One Participant in Regard to Text Recommendation Support (Warmer Colors Indicate Areas in Which the Interaction Activity Was High). Obtained From Hotjar.



textual recommendations enabled, it is possible to identify interactions with the dialog box at the bottom (which holds the textual recommendations). In this case, the user clicked on this box, trying to close it unsuccessfully. One of the potential explanations for this phenomenon could be that the textual recommendations employed technical language out of the scope of the participants' level of understanding, so they may have assumed these hints were not aimed at them, and thus ignored them.

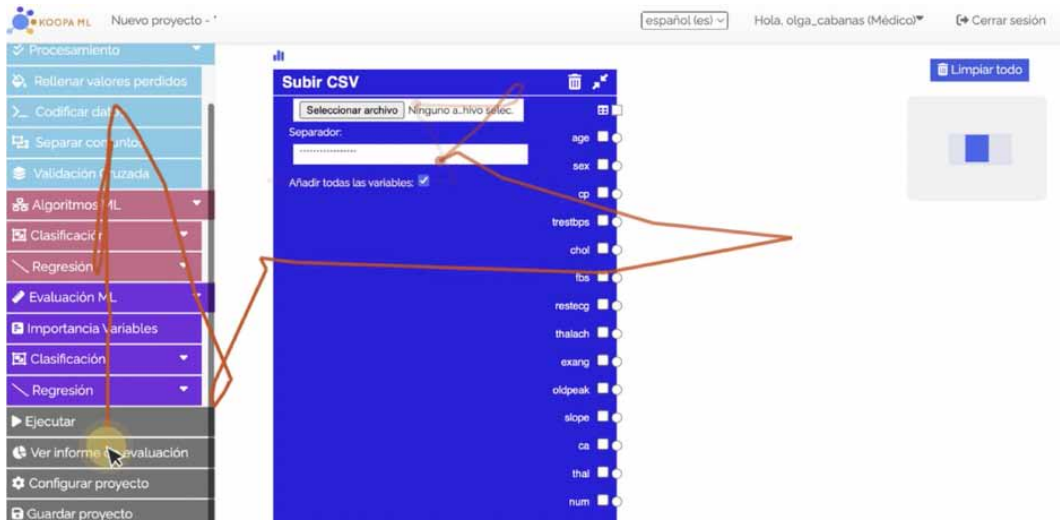
From these interactions, we conclude, in this context, that textual recommendations are not as useful as they may seem to be. Participants verbalized they were overwhelmed by these "hints" and started ignoring them, which led to similar task completion rates as the participants who did not rely on the system's assistance. In this sense, we pretend to modify the workspace to follow an onboarding approach with visual recommendations and automated assistance instead of textual recommendations.

Another interesting insight reached via the user evaluation is through the problems they experienced while finding the right nodes that were requested by the tasks. Figure 8 also shows the user spent most of the time clicking through the side menu, opening and closing the dropdowns that categorize the different ML nodes that can be included in the pipeline. In general, this pattern can be identified in every participant, which could indicate the menu should be redesigned (and even removed) to avoid users getting lost in it. From this behavior, it is concluded that showing too many options to non-expert users can overwhelm them, especially in such a complex context as AI-powered applications. In this sense, the language needs to be simplified and guidance should be the responsibility of the system, which should not be relegated to occasional assistance, as is the case of textual recommendations (in which the final responsibility is put on the user).

Figure 11 displays the path (in red) of another participant's mouse while they were searching for the "Split dataset" node. Although the participant hovered over the right node, they could not fully understand it was the node that was being requested by the task and went through the whole menu several times unsuccessfully. This is not only a usability problem but a deeper misunderstanding of the concepts involved in the ML context.

Finally, other minor usability issues were detected throughout the evaluation. For example, nodes take up a lot of space on the screen, and if several nodes are added, it is very difficult to manage them

Figure 11. Path of One Participant's Mouse While Searching the Next Node Requested in T1 (The Red Path Indicates the Real Trajectory of the Cursor Made by the User During the Study). Obtained From Hotjar.



properly. Moreover, on several occasions, users dragged the mouse to connect two nodes by their respective sockets, which caused usability issues because two nodes' sockets are meant to be connected by clicking first on one of them and then on the other. However, although most participants were not able to finish the tasks properly, they commented at the end of the study they were not experts in ML, which caused confusion regarding the goal of the tasks to be achieved. Most of the participants pointed out the system seemed very intuitive to them, and their difficulties in using KoopaML were mostly related to their lack of expertise in ML. The participant who had some expertise and experience in ML was able to achieve both tasks efficiently through our system.

As seen in Table 2, previous knowledge was a more influential factor in completing the proposed tasks. This observation leads to several important considerations regarding the role of textual recommendations in building ML pipelines using a visual platform. Firstly, the data suggests prior knowledge of ML plays a more critical role in completing tasks than the assistance provided by textual recommendations. This could be attributed to the nature of the recommendations themselves, as perhaps they were too generic or not sufficiently tailored to the varying expertise levels of the users. It is also possible the visual platform's design may have overshadowed the utility of the textual recommendations, especially for more experienced users who might rely more on their expertise than on external guidance. Furthermore, the lack of significant impact from the recommendations on participants' performances indicates a need to reassess how these recommendations are integrated into the visual platform. As the interaction data suggest, they weren't noticeable by the users, nor did they provide actionable, context-specific advice that users found valuable.

On the other hand, the obtained results prompt a deeper exploration into how KoopaML can be optimized for users with varying expertise levels. For instance, it may be beneficial to consider adaptive recommender systems that tailor suggestions based on user proficiency or to enhance the platform's interface to alleviate common issues experienced by participants. Further, the integration of more interactive and responsive error messaging could assist users in overcoming obstacles more effectively. In conclusion, while the recommender system did not uniformly enhance performance across the participant group, there is evidence to suggest small hints, whether through textual recommendations or execution error messages, have the potential to improve user engagement and task completion in KoopaML but not in their current form. This insight lays the groundwork for iterative platform



enhancements and future studies could focus on personalized user support mechanisms. Addressing and validating the identified issues from the user evaluation will be imperative in the development of the next version of KoopaML. This will involve either avoiding or redesigning the interface elements that presented challenges for the participants.

Finally, some limitations of this study include the small sample size employed. Although the think-aloud protocol provides reliable results for small samples, we were not able to perform an in-depth quantitative analysis. This study primarily employs a qualitative approach to capture users' thoughts and experiences while using the platform. We believe this approach provides valuable insights into user behavior and interaction with the recommender system, which may not be fully apparent through quantitative metrics alone. However, we acknowledge the importance of performance metrics in understanding the efficiency and accuracy of ML pipelines. To address this, future research would also focus on expanding the sample size to enable a more robust quantitative analysis.

Furthermore, another potential concern of using the think-aloud protocol is that verbalizing thoughts might alter the way users interact with the system, potentially leading to atypical behavior or decision-making processes. To mitigate this, participants were briefed thoroughly on the purpose of the think-aloud protocol and were encouraged to behave as naturally as possible.

## Discussion From the Medical Point of View

In this study, we evaluated how machine learning platforms such as KoopaML could help physicians. Artificial intelligence (AI) has emerged as an important tool in medical research. However, although there are many publications in the literature, AI in medicine is not yet well established. Furthermore, the potential applications of machine learning models can improve clinical diagnoses, risk stratification, and patient-specific management. This will promote more personalized medicine and can prompt better exploitation of medical resources (MacEachern & Forkert, 2021). In this regard, KoopaML was created to facilitate the work of physicians in building predictive models with machine learning and to promote AI education among healthcare professionals. Building machine learning prediction models requires a minimum of AI knowledge (Rajkomar et al., 2019). Not only are the fundamentals of machine learning important, but also a minimum knowledge of calculus, statistics, or computer programming must be present. Likewise, preprocessing data is crucial in machine learning. For these reasons, the participants experienced the various difficulties mentioned above during the exercises. Furthermore, although no notable differences were found between the groups, the participants who received recommendations during the exercises completed the proposed tasks easier and faster than the others. This probably highlights the need for further training in machine learning among physician participants. The participants also emphasized the exercises could not have been completed without these recommendations.

One of the most remarkable disadvantages of applying AI in the medical context is the lack of AI training in physicians (Kolachalama, 2022), and this study proves it. This is also corroborated by other studies (Kolachalama & Garg, 2018). Therefore, efforts should be made to help medical professionals acquire AI knowledge (Handelman et al., 2018). One of the possible reasons for this lacuna is the lack of mentoring in this field during medical school (Kolachalama, 2022). When training physicians in AI, different aspects should be assessed; not only should the technical aspects of the ML methods be explored, but also the potential applications this technology offers to solve different medical problems should be incorporated into research. In this study, we found the basic knowledge of physicians related to AI is scarce, and better education on AI and ML should probably be included in medical school curriculum.

Identifying those areas where ML could be useful is crucial to expanding its use in medical practitioners. For example, teaching how predictive models can be helpful in the diagnosis, treatment, or evaluation of a patient's prognosis could be an important goal to engage physicians with ML methodologies. Moreover, knowledge of basic ML model concepts is required to use and apply them. Physicians must understand the process of defining endpoints and selecting inputs and outputs. Also,

data literacy is a cornerstone in this context. This study shows most of the mistakes encountered during the proposed tasks were specifically related to a lack of understanding. In general, we believe educating physicians in ML would not only benefit them by allowing them to carry out complex analyses with these techniques but also to create quality databases that can be used to train better models.

One of the highlights of this study is that most of the participants emphasized the simplicity of the platform. In addition, it was observed, as they progressed through the exercises, the time spent on each task was shorter than the time spent on previous tasks. This might imply that just a short training with the application will able physicians to create self-made machine learning models. Recommendations also facilitated the development of the tasks; therefore, KoopaML appears to have a relevant role in future medical studies. In conclusion, this study highlights the necessity of increasing the AI literacy of physicians to promote the applicability of these models in daily medical practice.

## CONCLUSION

This work details a user evaluation of KoopaML, a platform to support non-expert users in the creation and execution of ML pipelines. The main goal of the user evaluation was to determine the usefulness of textual recommendations aimed at non-expert users during the process of designing an ML pipeline graphically. The results show textual recommendations did not make a significant difference in performance compared to the group of participants who did not have any support from textual recommendations. These results seem to be related to the fact that the participants were mostly non-expert users of ML. However, some interesting insights about the interface elements have been reached after analyzing the interaction patterns of the participants during the study. Most of the issues related to the use of technical language in the recommendations and menus and to the arrangement of the interface elements on the screen. In this sense, we suggest incorporating an onboarding approach to guide users during their first steps on the platform. In conclusion, this study's motivation was to unravel the complexities faced by non-expert users when engaging with ML concepts and to assess whether textual recommendations could simplify their learning curve.

The challenges identified, such as the limited impact of recommendations due to technical language and the overshadowing of visual elements, highlight a gap between user expertise and the platform's usability. To overcome these challenges, the prescribed approach involves refining KoopaML to better cater to users' varying levels of expertise. This could include simplifying the language of the recommendations, making them more context-specific, and enhancing the platform's interactivity. By addressing these issues, KoopaML aims to not only make ML more accessible to a broader audience but also to ensure users can more effectively engage with and understand ML concepts, leading to better outcomes in their respective fields.

Future research lines will involve the redesign of the KoopaML workspace to address the issues found during the presented case study, as well as to validate the changes with additional user evaluations. This redesign aims to enhance user experience and interface intuitiveness, particularly for those with limited ML expertise. Efforts will be made to simplify the technical language of the recommendations, improve the visibility and relevance of guidance cues, and streamline the overall workflow to reduce cognitive overload for users. Additionally, the research will incorporate the development of more adaptive and user-friendly features, potentially including a more responsive error feedback system and interactive tutorials tailored to varying levels of user proficiency. We also plan to conduct more qualitative studies to gain deeper insights into properly training physicians in AI techniques, as well as explore how different attributes, such as the specialty of physicians, influence their understanding of ML. We also intend to conduct long-term studies to evaluate the acceptance and proficiency of the enhanced version of KoopaML once it is deployed in a production environment.

## REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., & Wicke, M. (2016). *TensorFlow: A system for large-scale machine learning*. <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- Almomani, A., Alauthman, M., Shatnawi, M. T., Alweshah, M., Alrosan, A., Alomoush, W., Gupta, B. B., Gupta, B. B., & Gupta, B. B. (2022). Phishing Website Detection With Semantic Features Based on Machine Learning Classifiers: A Comparative Study. *International Journal on Semantic Web and Information Systems*, 18(1), 1–24. doi:10.4018/IJSWIS.297032
- Anil, R., Capan, G., Drost-Fromm, I., Dunning, T., Friedman, E., Grant, T., Quinn, S., Ranjan, P., Schelter, S., & Yilmazel, Ö. (2020). Apache mahout: Machine learning on distributed dataflow systems. *Journal of Machine Learning Research*, 21(127), 1–6. <https://jmlr.org/papers/v21/18-800.html>
- Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., & Wiswedel, B. (2009). KNIME - the Konstanz information miner. *SIGKDD Explorations*, 11(1), 26–31. doi:10.1145/1656274.1656280
- Bisong, E. (2019). Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 59–64. 10.1007/978-1-4842-4470-8\_7
- Bjaoui, M., Sakly, H., Said, M., Kraiem, N., & Bouhleb, M. S. (2020). Depth insight for data scientist with RapidMiner “an innovative tool for AI and big data towards medical applications.” *Proceedings of the 2nd International Conference on Digital Tools & Uses Congress*, (21). doi:10.1145/3423603.3424059
- Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *IEEE Access*, 8, 75264–75278. 10.1109/ACCESS.2020.2988510
- Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., Guppy, K. H., Lee, S., & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5), 304–310. doi:10.1016/0002-9149(89)90524-9 PMID:2756873
- Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository*. University of California, School of Information and Computer Science.
- Dwivedi, R. K., Kumar, R., & Buyya, R. (2021). Gaussian Distribution-Based Machine Learning Scheme for Anomaly Detection in Healthcare Sensor Cloud. *International Journal of Cloud Applications and Computing*, 11(1), 52–72. doi:10.4018/IJCAC.2021010103
- Fayoumi, A. G., & Hajjar, A. F. (2020). Advanced learning analytics in academic education. *International Journal on Semantic Web and Information Systems*, 16(3), 70–87. doi:10.4018/IJSWIS.2020070105
- Ferrer, X., van Nuenen, T., Such, J. M., Coté, M., & Criado, N. (2021). Bias and discrimination in AI: A cross-disciplinary perspective. *IEEE Technology and Society Magazine*, 40(2), 72–80. doi:10.1109/MTS.2021.3056293
- Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. (2009). Weka-A machine learning workbench for data mining. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 1269–1277). doi:10.1007/978-0-387-09823-4\_66
- Gambier, Y., & Doorslaer, L. v. (Eds.). (2010). *Handbook of Translation Studies*. John Benjamins Publishing Company. doi:10.1075/hts.1
- García-Holgado, A., Vázquez-Ingelmo, A., Alonso-Sánchez, J., García-Peñalvo, F. J., Therón, R., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P. I., & Sánchez, P. L. (2022). KoopaML, a machine learning platform for medical data analysis. *Journal on Interactive Systems*, 13(1), 154–165. doi:10.5753/jis.2022.2574
- García-Peñalvo, F., & Vázquez-Ingelmo, A. (2023). What Do We Mean by GenAI? A Systematic Mapping of The Evolution, Trends, and Techniques Involved in Generative AI. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(4), 7–16. doi:10.9781/ijimai.2023.07.006

- García-Peñalvo, F., Vázquez-Ingelmo, A., García-Holgado, A., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P. I., & Sánchez, P. L. (2023). KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals. *International Journal of Interactive Multimedia and Artificial Intelligence*. doi:10.9781/ijimai.2023.01.006
- Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, 40(1-3), 11–61. doi:10.1016/0004-3702(89)90046-5
- George, G., & Lal, A. M. (2021). A personalized approach to course recommendation in higher education. *International Journal on Semantic Web and Information Systems*, 17(2), 100–114. doi:10.4018/IJSWIS.2021040106
- González Izard, S., Juanes, J. A., García Peñalvo, F. J., Estella, J. M. G., Ledesma, M. J. S., & Ruisoto, P. (2018). Virtual reality as an educational and training tool for medicine. *Journal of Medical Systems*, 42(3), 50. Advance online publication. doi:10.1007/s10916-018-0900-2 PMID:29392522
- González Izard, S., Sánchez Torres, R., Alonso Plaza, Ó., Juanes Méndez, J. A., & García-Peñalvo, F. J. (2020). Nextmed: Automatic imaging segmentation, 3D reconstruction, and 3D model visualization platform using augmented and virtual reality. *Sensors (Basel)*, 20(10), 2962. doi:10.3390/s20102962 PMID:32456194
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2022). The WEKA data mining software. *SIGKDD Explorations*, 11(1), 10–18. doi:10.1145/1656274.1656278
- Hoffman, S. (2021). The emerging hazard of AI-Related health care discrimination. *The Hastings Center Report*, 51(1), 8–9. Advance online publication. doi:10.1002/hast.1203 PMID:33315263
- Karthika, R., & Jegatha, D. L. (2021). Enhanced learning experiences based on regulatory fit theory using affective state detection. *International Journal on Semantic Web and Information Systems*, 17(4), 37–55. doi:10.4018/IJSWIS.2021100103
- Kolachalama, V. B. (2022). Machine learning and pre-medical education. *Artificial Intelligence in Medicine*, 129, 102313. doi:10.1016/j.artmed.2022.102313 PMID:35659392
- Kolachalama, V. B., & Garg, P. S. (2018). Machine learning and medical education. *npj. Digital Medicine*, 1(1), 54. doi:10.1038/s41746-018-0061-1 PMID:31304333
- Kulkarni, R. H., & Padmanabham, P. (2017). Integration of artificial intelligence activities in software development processes and measuring effectiveness of integration. *IET Software*, 11(1), 18–26. doi:10.1049/iet-sen.2016.0095
- Lampa, S., Alvarsson, J., & Spjuth, O. (2016). Towards agile large-scale predictive modelling in drug discovery with flow-based programming design principles. *Journal of Cheminformatics*, 8(1), 67. Advance online publication. doi:10.1186/s13321-016-0179-6 PMID:27942268
- Lampa, S., Dahlö, M., Alvarsson, J., & Spjuth, O. (2019). SciPipe: A workflow library for agile development of complex and dynamic bioinformatics pipelines. *GigaScience*, 8(5). Advance online publication. doi:10.1093/gigascience/giz044 PMID:31029061
- Li, S., Qin, D., Wu, X., Li, J., Li, B., & Han, W. (2022). False Alert Detection Based on Deep Learning and Machine Learning. *International Journal on Semantic Web and Information Systems*, 18(1), 1–21. doi:10.4018/IJSWIS.313190
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. doi:10.1016/j.media.2017.07.005 PMID:28778026
- MacEachern, S. J., & Forkert, N. D. (2021). Machine learning for precision medicine. *Genome*, 64(4), 416–425. doi:10.1139/gen-2020-0131 PMID:33091314
- Malik, C. P., Prabha, C., Soni, P., Arya, V., Alhalabi, W. A., Gupta, B. B., Albeshri, A. A., & Almomani, A. (2023). Machine learning-based automatic litter detection and classification using neural networks in smart cities. *International Journal on Semantic Web and Information Systems*, 19(1), 1–20. doi:10.4018/IJSWIS.324105
- Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test. *International Journal of Human-Computer Studies*, 41(3), 385–397. doi:10.1006/ijhc.1994.1065

- Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *The New England Journal of Medicine*, 380(14), 1347–1358. doi:10.1056/NEJMra1814259 PMID:30943338
- Rodríguez-García, J., Jesús Moreno-León, Román-González, M., & Robles, G. (2021). Evaluation of an online intervention to teach artificial intelligence with learningml to 10-16-year-old students. *Technical Symposium on Computer Science Education*. doi:10.1145/3408877.3432393
- Someren, M., Barnard, Y., & Sandberg, J. (1994). *The Think Aloud Method – A Practical Guide to Modelling Cognitive Processes*. Academic Press. [https://www.researchgate.net/publication/215439100\\_The\\_Think\\_Aloud\\_Method\\_-\\_A\\_Practical\\_Guide\\_to\\_Modelling\\_CognitiveProcesses](https://www.researchgate.net/publication/215439100_The_Think_Aloud_Method_-_A_Practical_Guide_to_Modelling_CognitiveProcesses)
- Vázquez-Ingelmo, A., Alonso-Sánchez, J., García-Holgado, A., José, F., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Ignacio Dorado-Díaz, P., & Sánchez, P. L. (2021). Bringing machine learning closer to non-experts: Proposal of a user-friendly machine learning tool in the healthcare domain. *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*. doi:10.1145/3486011.3486469
- Wachter, S., Mittelstadt, B., & Russell, C. (2020). Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. *SSRN Electronic Journal*, 41(2021), 105567. 10.2139/ssrn.3547922
- Weyerer, J. C., & Langer, P. F. (2019). Garbage in, garbage out. *Proceedings of the 20th Annual International Conference on Digital Government Research*. doi:10.1145/3325112.3328220
- Wu, A., Wang, Y., Shu, X., Moritz, D., Cui, W., Zhang, H., Zhang, D., & Qu, H. (2021). AI4VIS: Survey on artificial intelligence approaches for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(12), 5049–5070. doi:10.1109/TVCG.2021.3099002 PMID:34310306
- Yu, K.-H., Beam, A. L., & Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nature Biomedical Engineering*, 2(10), 719–731. doi:10.1038/s41551-018-0305-z PMID:31015651

## ENDNOTE

- <sup>1</sup> <https://www.youtube.com/watch?v=JeQrz2I20TY> (contents in Spanish)