On Building Design Guidelines For An Interactive Machine Learning Sandbox Application

Giselle Nodalo
De La Salle University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Jose Maria Santiago
De La Salle University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Jolene Valenzuela
De La Salle University
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Jordan Aiko Deja
De La Salle University
22 PUBLICATIONS **13** CITATIONS
SEE PROFILE

On Building Design Guidelines For An Interactive Machine Learning Sandbox Application

Giselle Nodalo
De La Salle University
Manila, Philippines
giselle_nodalo@dlsu.edu.ph

Jose Ma. Santiago III
De La Salle University
Manila, Philippines
jose_maria_santiago@dlsu.edu.ph

Jolene Valenzuela
De La Salle University
Manila, Philippines
jolene_valenzuela@dlsu.edu.ph

Jordan Aiko Deja
De La Salle University
Manila, Philippines
jordan.deja@dlsu.edu.ph

ABSTRACT

There are several machine learning suites, applications and platforms that are readily-available right now. However, these applications require a basic foundation in machine learning making them appear difficult to configure. We introduce the use of the Sandbox approach with the goal of designing alternative programming interfaces and interactions for machine learning tasks. A set of guidelines have been drafted supported with initial user tests to validate the proposed design framework. Ten (10) undergraduate and graduate students with novice experience in machine learning participated in structured interviews to formulate programming patterns and styles that were used to draft the guidelines based on the literary review. These insights were analyzed using UX Research techniques to form a general problem statement. Initial analysis of the user insights suggested that a visual sandbox approach similar to Scratch helps reduce the steepness of the learning curve. The preliminary design guidelines we drafted focused on the three design factors namely System Intent, Interaction, and Algorithm Visualization. From these guidelines we produced an initial version of the prototype that will be subject to further testing and validation.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User–Centered Design

Author Keywords

Interactive Machine Learning; Grounded Approach; Design Guidelines;

CHIUXID'19, March 31–April 09, 2019, Bali, Indonesia

ISBN 123-4567-24-567/08/06. . . $15.00

DOI: http://dx.doi.org/10.475/123_4

INTRODUCTION

Machine Learning (ML) algorithms and systems tend to abstract key operations during the learning process. Operations being abstracted from the user comes with its own advantages and disadvantages. Since complex operations are abstracted from the user, the risk of the user misunderstanding these operations is limited by the abstraction. Due to these abstractions, the user has limited access to customization of algorithms from the user which feeds the idea that non-experts treat ML as a fancy "black box" [13]. The "black box" effect, as seen in Figure 1, can be summarized as an object that uses an input to produce an output without revealing the process in-between. How the model is updated in every iteration is often ignored by the user because of this trend in traditional ML algorithms. This problem may be addressed however through visualization of the inner workings of the algorithm.

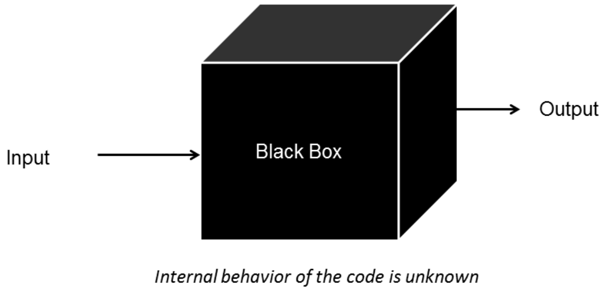


Figure 1. "Non-experts treat ML as a fancy *black box* with few convincing interpretation of the model structure and learning process." [13]. Imaged sourced from [2]

Interactive Machine Learning (iML) is a ML system that makes use of human interaction to further improve the efficiency and accuracy. According to [5], it is believed that iML can be used to enable users in creating and customizing the ML applications they work on. Since faster processing units have been introduced to the general market, the cost to practice ML has also gone down. This further enhances the capabilities of machines to handle high dimensions of data, which begs the question whether or not we can improve the

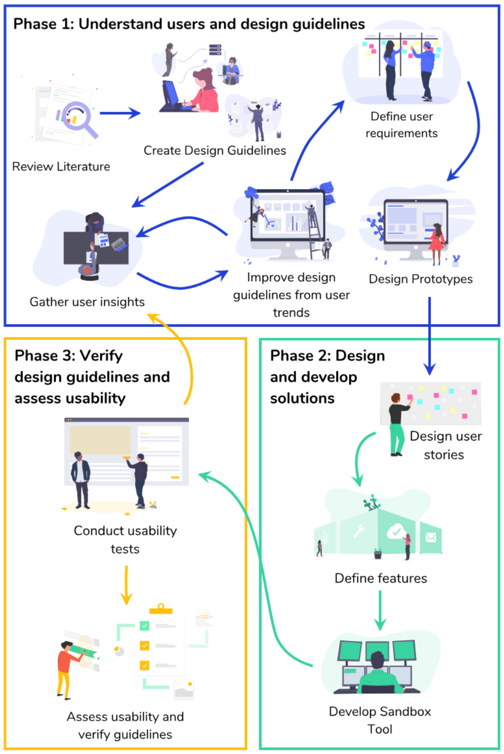

Phase 1: Understand users and design guidelines
Review Literature
Create Design Guidelines
Define user requirements
Gather user insights
Improve design guidelines from user trends
Design Prototypes
Phase 3: Verify design guidelines and assess usability
Conduct usability tests
Assess usability and verify guidelines
Phase 2: Design and develop solutions
Design user stories
Define features
Develop Sandbox Tool

Figure 2. Overview of the three phases in this study. The approach follows a user-centric iterative methodology towards building a sandbox application. Phase 1 involves an in-depth and thorough analysis and study of the users in order to understand the needs of the target users. Phase 2 and 3 are linked in a way where we can build-test-deploy a minimum viable but usable product

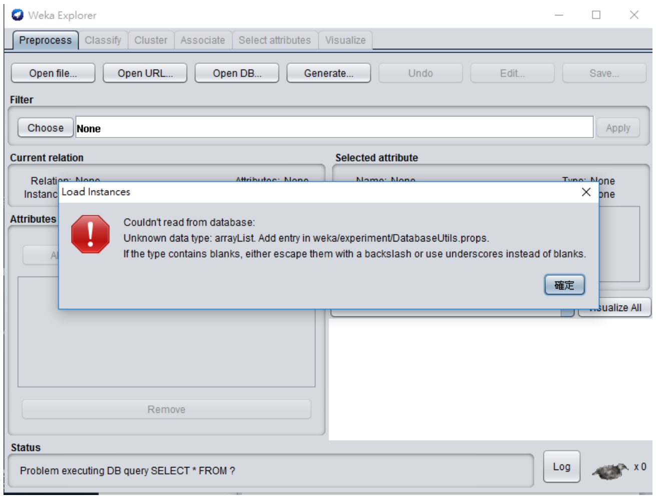
efficiency and speed of the current machine learning systems being used today.

In this paper, we will be discussing the methods, objectives and the results from the first phase of a three-phase project with the main goal of building an interactive ML sandbox application. We followed the framework approach to doing human-centric ML as seen in the works of [7, 9, 8, 24]. In these works, there is greater emphasis on the users being included in the design process of identifying and crafting the needs for a specific use case. Especially in the works of [7] and [9], the process becomes iterative, an approach that is similar to participatory design [20]. The first phase aims to make a set of guidelines for designing such applications. The second and third phases, involve an iterative design-build-test approach where users evaluate multiple iterations of the prototype towards finalizing an acceptable and usable minimum viable product, are not yet included in this paper. The phases of the project are illustrated as seen on Figure 2. Details on the interviews, study design, and protocol relevant to the first phase are described on the succeeding sections.

RELATED WORKS

We believe that Interactive Machine Learning (iML) is a possible solution to the black box effect. It can be summarized as a human-centered approach to machine learning. We have found several existing applications which provide a way to help users, both non-expert and expert, to experiment on ML algorithms through human interaction. Examples of such are Weka [11] which focuses on providing a unified tool that gives easy access to state-of-the-art ML techniques through a visual interface and RapidMiner [14] which focuses on packaging ML algorithms for data mining.

However, there are certain design limitations of RapidMiner that can affect the usability of the tool. One of the frustrations that was observed was the lack of visualization of the machine learning process. Since RapidMiner abstracts processes that limits the flexibility of the tool, it requires the data set to meet certain formatting conditions as observed in Figure 4. The use of technical labels regarding tool functions and error messages is another pain point that was observed. This was due to the limited understanding of terminologies that are displayed in error codes to the user.



None

Relation: None
Attributes: None
Instances: None

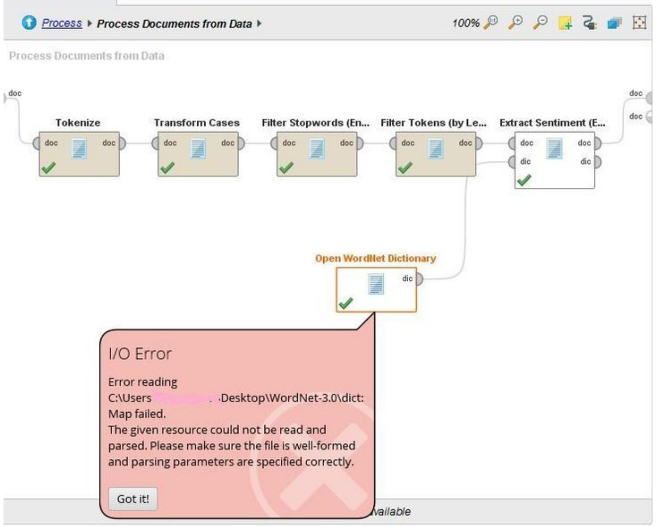
Name: None
Type: None

Load Instances

Couldn't read from database:
Unknown data type: arrayList. Add entry in weka/experiment/DatabaseUtils.props.
If the type contains blanks, either escape them with a backslash or use underscores instead of blanks.

Problem executing DB query SELECT * FROM ?

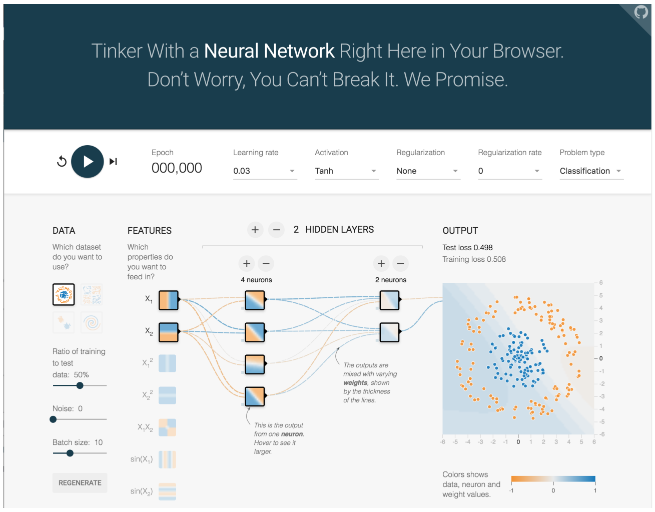
Figure 3. A screenshot of Weka encountering a Load Instance error due to an unknown data-type in the database. Image sourced from [25]



I/O Error

Error reading
C:\Users [redacted] Desktop\WordNet-3.0\dict:
Map failed.
The given resource could not be read and parsed. Please make sure the file is well-formed and parsing parameters are specified correctly.

Figure 4. A screenshot of RapidMiner encountering an error reading an incompatible resource file. Image sourced from [12]



Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Epoch 000,000
Learning rate 0.03
Activation Tanh
Regularization None
Regularization rate 0
Problem type Classification

DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10

FEATURES
Which properties do you want to feed in?

2 HIDDEN LAYERS

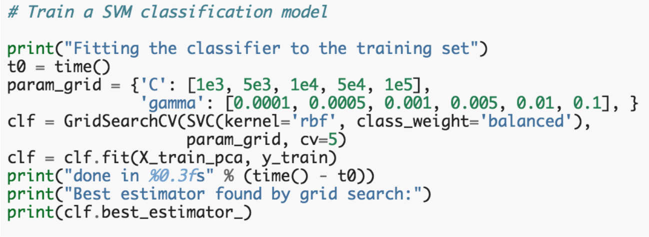
OUTPUT
Test loss 0.498
Training loss 0.508

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

Figure 5. A screenshot of Tensorflow Playground which is a visualization tool for Neural Networks. Image sourced from [22]



```
# Train a SVM classification model

print("Fitting the classifier to the training set")
t0 = time()
param_grid = {'C': [1e3, 5e3, 1e4, 5e4, 1e5],
              'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1], }
clf = GridSearchCV(SVC(kernel='rbf', class_weight='balanced'),
                   param_grid, cv=5)
clf = clf.fit(X_train_pca, y_train)
print("done in %0.3fs" % (time() - t0))
print("Best estimator found by grid search:")
print(clf.best_estimator_)
```

Figure 6. A code sample from Scikit-Learn using the *Fit Function*. Image sourced from [1]

Similar to RapidMiner, WEKA suffers from data incompatibility errors presented through Java Exception Errors as seen in Figure 3. It was observed that this tends to confuse the user that was using the tool as a standalone application. While WEKA visualizes the machine learning and data mining processes, frustrations regarding the complicated visualization for these processes were also observed. WEKA also uses technical terminologies which limits user feedback for users that are not well-versed in machine learning.

Such tools are still not at par with the functionality of ML libraries such as Scikit-learn, TensorFlow, and PyTorch, which may require a steep learning curve for those who are not familiar with Python or programming in general. Scikit-learn, as seen in Figure 6, is a Python module that focuses on bringing machine learning to users through a general-purpose high-level language [17]. However, the pain points observed in Scikit-learn are different compared to the ones mentioned in previous tools. While the functions used in creating machine learning systems were flexible, it lacked the visualization for the process.

Scratch, a visual programming language seen in Figure 7, was developed to address the difficulty of learning programming for novice users, especially young users [3]. According to [16], using Scratch, programming concepts and fundamentals were grasped by high school students without worrying about syntax. In addition, according to [21] algorithm visualization tools have already been developed to help users understand how certain algorithms work, however older tools became inaccessible as operating systems improved. An improved way to learn algorithms was developed by [10], combining user interactions, conversion to pseudocode, and animation. It also eliminated the problem of system compatibility and lack of visualization consistency. These applications and systems apply the known sandbox approach. According to [18] the sandbox approach is an isolated computing environment which a program can run without affecting the application it runs in. Such approaches are known to help users abstract information, visualize complex instructions thus reducing cognitive loading and learning curves [6].

Table 1. A summary and comparison of machine learning tools currently available.

Tools	Uses Interactive Visualizations	Abstracts Algorithm Code	User Feedback
RapidMiner	×	✓	Limited
WEKA 3	×	✓	Limited
Scikit Learn	×	×	N/A

Algorithm visualizations were already attempted in the domain of machine learning, but are usually static and still have a feeling of the "black box" effect. Other tools attempted a more responsive approach by visually showing the effect of the user's changes to the training model accuracy. An example of which is [19], a tool made using spreadsheets, a familiar end-user interface. There has also been a tool made for TensorFlow as illustrated in Figure 5 designed for users, especially non-experts, to experiment with ML algorithms and visualize the model simultaneously with user manipulation using the sandbox approach [23]. iML visualizations, however, do not guarantee that end-users primarily learn from the feedback of the system. Explanatory user feedback is needed to provide the user with context about the model visualization and how changes made by the user affected the output visually.

In that regard, we believe that the use of Scratch can be explored to help teach ML to novice users, but the development of such a tool must be governed by design guidelines first to ensure that end-user goals are met. The Scratch programming tool makes use of the sandbox approach to help young people learn to think creatively, reason systematically, and work collaboratively [3]. The goal with using a sandbox approach is to introduce an environment that lets the user explore and create while minimizing the risk of a system error.

The work of [4] created a list of design considerations to create an effective iML system. These design factors, coupled together with the insights gathered from our interviews with ML users, we created design guidelines which will help in building an interactive sandbox application for learning ML. We believe that the creation of such guidelines will help solve the problem on the lack of design standards for iML systems.

METHODS

The first phase of this study included a set of activities that is aimed towards better understanding users and their tasks

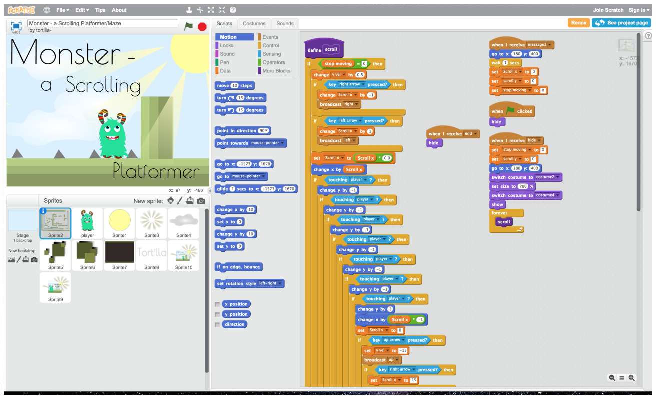


Figure 7. MIT Scratch Studio Interface. Image sourced from [3]

at hand. These activities include User Interviews, Defining User Requirements, Literature Review, Creating and Mapping of Design Guidelines, Gathering User Insights, and Initial Prototyping.

Participants

Ten (10) computing students and professionals aged 19-25 were recruited through a purposive sampling method. Note that these participants are divided into three categories: novice ML users that have only used ML tools, novice ML users that have both taken a ML course and used ML tools, and expert ML users that proactively use ML in their jobs.

Each participant was interviewed about their experience with ML and the various tools they have encountered. There were two sets of interview questions depending on whether the participant has taken a ML course or not. For participants that have taken a ML course, in depth questions about ML processes and setups were asked to gain further insights into their techniques and pain points. These questions can be seen in the study design.

Study Design

Prior to the interview, a one week schedule was given to contact interview participants. The participants were given the options to specify the interview time and location that is convenient for them. Prior to the interview, an informed consent form was given to the participant to read and sign. The consent form contained the purpose of the interview and the rights and responsibilities of the participant including the permission to audio record their interview. Finally, the participant was briefed about the interview flow before proceeding with the interview.

During the interview, each participant was asked three (3) sets of questions with at least two (2) questions per set. The goal of the first set was to identify the current stand of the participant regarding their experiences in learning about ML and the various tools they have used. The goal of the second set was to identify the various problem areas that the participant associated with based on their experience using current ML tools. The third set was used to identify improvements regarding usability of the tool, as pointed out by the participant. The questions that were used are as follows:

- Q1 - Could you briefly explain your introduction to Intelligent Systems or Machine Learning course experience?
- Q2 - Have you used Machine Learning tools such as WEKA, RapidMiner, or Scikit-learn? What do you like or dislike about them?
- Q3 - In relation to the Machine Learning algorithms you have used, how important was the visualization when learning about these algorithms?
- Q4* - Based on your experience, have you encountered problems in these areas when learning Machine Learning:
  - Using software needed to test or create Machine Learning systems.
  - Understanding the impact or severity of each individual function in the system.
  - Visualizing the process of data fitting and prediction.
  - Pre-processing the data before fitting.
  - Establishing and connecting libraries and determining which functions to use for a specific task.
  - Others.
- Q5* - Which code libraries or software tools do you use to visualize Machine Learning Algorithms and how well do you understand the output of these tools?
- Q6 - How would you react to a tool that allows you to visualize the structure and process of a Machine Learning algorithm through code blocks? How do you think this tool will change the way people learn Machine Learning algorithms?

* *These questions were only asked to participants that have taken a Machine Learning Course.*

The preliminary design guidelines were introduced to the participants as a survey where they were asked to give rating using a 4 point likert scale with 1 being Strongly Disagree and 4 being Strongly Agree. The results in this survey can be seen in Table 2. Each participant was free to ask questions and give insights to help improve the design guidelines.

Data Analysis

After the interviews, the user insights from both the interview and additional comments from the survey were gathered and analyzed through an affinity mapping session where user verbatim was extracted and grouped into similar user statements until a general problem statement was formed. For the survey, the rating results were extracted and were analyzed to find a pattern in each participant preference for each guideline in order to identify the relevance of the guideline.

RESULTS

This section contains the insights from the interviews that we have conducted and the design guidelines that we have formulated based on the insights and survey results.

User Insights

According to our participants, studying machine learning concepts was interesting in application based on their experience with ML tools. However, the novice participants stated that learning the mathematical theories and concepts were confusing to understand and that most of their ML classes resulted in experimenting with the libraries and documentations they could find. In addition, expert participants say that it is up to the learner to dive deeper in understanding ML theory and practice.

RapidMiner was praised by participants for the visual interface that was able to show the process of the ML system. However, most of the participants expressed frustrations regarding features in RapidMiner as a tool which were similar to those observed in related works. One of these frustrations was about the restrictions with certain data sets that does not meet certain formatting conditions, as expressed by the participants. Another frustration that the participants have mentioned was about the use of confusing technical labels and terminologies regarding tool functions and error messages. These frustrations were due to the limited feedback of the tool which affected the understanding of the user.

Similar pain points were also observed by the participants when using WEKA. It suffers from data incompatibility errors which limited the kind of data sets that were usable in the application. While WEKA visualizes the ML and data mining processes, frustrations regarding the complicated visualization for these processes were also observed by the participants. Similar frustrations with RapidMiner include problems with technical terminologies and limited user feedback.

Scikit-learn is another tool that the participants have associated with or used in their respective fields. Some participants mentioned that functions that they used when creating ML algorithms were flexible but lacks the visualization. This visualization would be useful when trying to understand the purpose of the function and how it affects the data set. It was pointed out that the documentation was one of the features of Scikit-learn that helped them understand the various functions of the library. They mentioned in the interview that a clear and exhaustive documentation is needed when defining the various functions of the tool to help the user understand these functions and how to apply them properly.

Regarding visualizations for understanding machine learning algorithms, most of the participants mentioned that visualization would help at first by introducing how the algorithm works and would help build the intuition for understanding the processes of the algorithm. Additionally, some of the novice participants mentioned that algorithm visualizations are helpful since diving into code can sometimes be difficult especially for those that are visual learners. Furthermore, the participants that took a ML course state that sometimes they have difficulty understanding the outputs of the visualization tools or libraries they use. Concepts also become clearer when they are accompanied by a visualization and it helps them appreciate understanding the algorithm.

The participants were asked to give their opinions on the possibility of having a sandbox Scratch-based tool that will help visualize the logic and structure of a ML algorithm. There were mixed reactions from the participants with some stating that it would help visual learners understand ML better and that it would lower the barrier of entry for non-experts. In addition, it would also enhance the learner's appreciation for ML through experimentation especially for those that get overwhelmed by code.

On the other hand, some of the participants that prefer a coding approach state that although the concept seems helpful for visual learners, user may rely too heavily on the blocks. In addition, they may ignore the code and it may seem more like spoon-feeding users rather than helping them understand ML. They also claim since there can be many ways to code ML tasks, users will not have the freedom and flexibility to edit code using code blocks.

For the feedback on the guidelines, most of the participants agreed with the need for guidelines usually suggesting specific outcomes they hope the guidelines will help future iML systems to achieve. Some participants were very particular about the fifth guideline, stating that the error recovery from the sandbox tool should not mention directly how to recover from the error since it feels like the system is spoon feeding the user. Instead, the error recovery should be enough to help guide the user in exploring their recovery options.

Design Guidelines

From the user insights extracted above, we formulated a set of guidelines considering the factors and constraints described and similar to that of the work of [4]. These were formulated by combining the insights we collected and the inputs from the literature review. The initial resulting guidelines are described below:

1. The system should indicate to the user its state of change with every interaction.

2. The components of the interface should be laid out in an organized manner and should be explicitly labelled.

3. Visualizations should be simple enough to understand and they should not overload the user.

4. The intent of the system should be clear to the user upon interaction.

5. The system should indicate explicitly when and why errors occur and how users can recover from them.

Along with mapping the constraints and factors in the work of [4] with the results of the initial interviews, we also asked the participants to score and rate their perception on these formulated guidelines. A scale of one (1) to four (4), namely, 1 as strongly disagree, 2 as disagree, 3 as agree and 4 as strongly agree, was given to the participants. With this, they scored each guideline in order for us to validate if these were formulated correctly and as needed. These guidelines will be used as defining standards in the development of a prototype that applies the sandbox approach in the proposed iML tool.

Table 2. Perception of the Participants on the Proposed Design Guidelines. Each respondent in the initial user research gave a rating to each guideline depending on how they agree to such design constraint and factor.

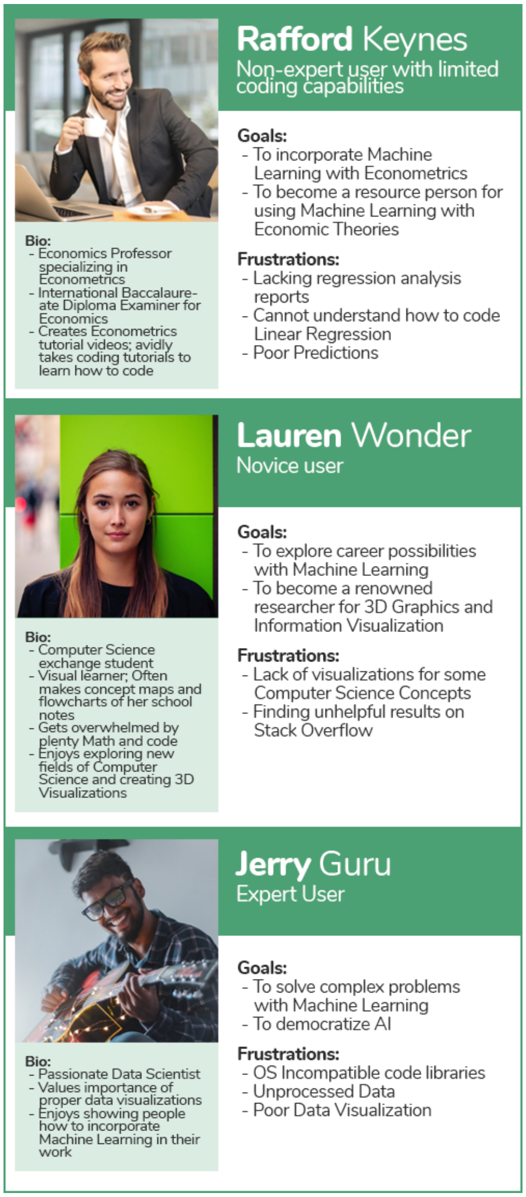
Guideline	1	2	3	4
1. The system should indicate to user its state of change with every interaction.	0%	0%	10%	90%
2. The components of the interface should be laid out in an organized manner and explicitly labelled.	0%	0%	10%	90%
3. The visualizations should be simple enough to understand. They should not overload the user.	0%	0%	70%	30%
4. The intent of the system should be clear to the user upon interacting with the system.	0%	0%	10%	90%
5. Indicate explicitly when and why errors occur and how users can recover from them.	0%	10%	30%	60%

From the results as seen in Table 2, it can be observed that most of the guidelines provided positive feedback as seen in the scores belonging to the Somewhat-agree (3) and Strongly-Agree (4) scales. Nine (9) out of ten (10) respondents agreed and believed that in order for the system to be interactive and usable, it should indicate the user its state of change with every interaction. The same number of respondents agreed that the interface must be laid out in an organized and properly-labelled manner. Upon careful review of these guidelines, it can be noticed that these do align from the Heuristics for Evaluating Interfaces as written by [15]; specifically "Provide feedback" and "Show system status and visibility". While these were somehow intentional but not disclosed to the respondents, it was interesting to find out how the guidelines do agree with the said standards defined by [15]. Interestingly, the results of the fifth (5th) guideline appear to be not totally inclined with the rest.

Upon consultation with the respondents concerned, we were able to extract and discover that some novice users prefer not to be "spoon-fed" with resolutions to errors and would actually prefer that they be given hint so they can discover on their own the solutions to these errors. While the work of Nielsen tells us in one of his heuristics that "error messages should be good, clear and simple and natural dialogue", some users prefer that these could be guiding and not to totally give away solutions. We believe that this may be related to the task at hand of learning ML. Such responses were also seen from the expert users who prefer that they be guided and not be given with the exact solution. We believe that this adds an intuitive and discoverable feature but these will have yet to be validated further.

Persona Design

After formulating a set of design guidelines, we came up with user personas based on the insights from the user interviews as seen on Figure 8.



Rafford Keynes
Non-expert user with limited coding capabilities

Bio:
- Economics Professor specializing in Econometrics
- International Baccalaureate Diploma Examiner for Economics
- Creates Econometrics tutorial videos; avidly takes coding tutorials to learn how to code

Goals:
- To incorporate Machine Learning with Econometrics
- To become a resource person for using Machine Learning with Economic Theories

Frustrations:
- Lacking regression analysis reports
- Cannot understand how to code Linear Regression
- Poor Predictions

Lauren Wonder
Novice user

Bio:
- Computer Science exchange student
- Visual learner; Often makes concept maps and flowcharts of her school notes
- Gets overwhelmed by plenty Math and code
- Enjoys exploring new fields of Computer Science and creating 3D Visualizations

Goals:
- To explore career possibilities with Machine Learning
- To become a renowned researcher for 3D Graphics and Information Visualization

Frustrations:
- Lack of visualizations for some Computer Science Concepts
- Finding unhelpful results on Stack Overflow

Jerry Guru
Expert User

Bio:
- Passionate Data Scientist
- Values importance of proper data visualizations
- Enjoys showing people how to incorporate Machine Learning in their work

Goals:
- To solve complex problems with Machine Learning
- To democratize AI

Frustrations:
- OS Incompatible code libraries
- Unprocessed Data
- Poor Data Visualization

Figure 8. Personas synthesizing the needs, goals, and behavior patterns observed from the participants during the user interviews. The personas mainly represent a non-expert user with minimal coding knowledge, a novice user, and an expert user of ML.

CONCLUSION AND FUTURE WORK

We have introduced design guidelines for an Interactive Machine Learning Sandbox Application. These were extracted and created by combining the factors and constraints defined by [4] and the results of the initial user research. To confirm these guidelines, we gathered insights from students who have novice and expert experience in ML. We also noted prior works on providing a graphical interface for ML such as WEKA and Rapidminer; such suites have a "black box" effect due to too much abstraction of processes and lack of versatility. The case is totally different and opposite as compared to some Python-based ML libraries that may require a steep learning curve especially for non-coders and those who are unfamiliar with Python.

However, the main purpose of the project is not to teach ML but to provide a safe exploratory environment for novice or non-expert users to create ML experiments. This is to reduce the difficulty of using the application and to help build the user's intuition for creating an ML algorithm. In addition, the application aims to visualize the ML algorithm to provide the user more context of how the data is manipulated. The visualization will be semi-dynamic to encourage user interaction by allowing the user to preview past samples of the experiment.

For our future work in this project, our second phase will involve designing and developing the sandbox application, which will provide a visual interface for novice learners of ML, and for the third phase we will be verifying the design guidelines we proposed in this paper by assessing the usability of our sandbox application. We have already began the design of the prototype, a screenshot can be seen in Figure 9. This initial concept has been made considering the sandbox approach and the guidelines from the results in this phase of the study.

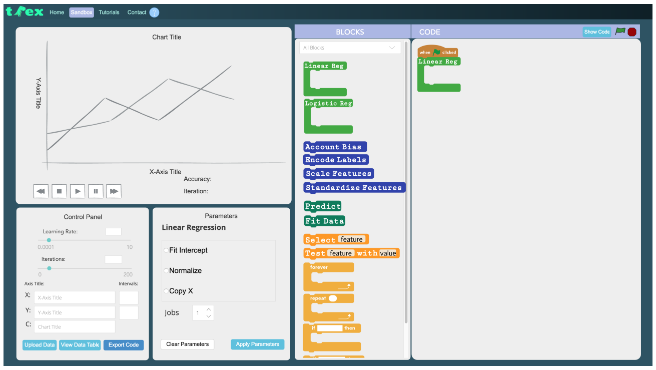


Figure 9. Screenshot of our Interactive Machine Learning Sandbox Application prototype.

ACKNOWLEDGMENTS

The authors would like to thank the research participants that contributed to gaining valuable insights for the development of the guidelines and the for serving as the inspiration behind the personas.

REFERENCES

1. 2007. Putting it all together. (2007). https://scikit-learn.org/stable/tutorial/statistical_inference/putting_together.html
2. 2015. The Aggravation with Conventional Black Box Testing. (2015). https://www.360logica.com/blog/the-aggravation-with-conventional-black-box-testing/the-aggravation-with-conventional-black-box-testing/
3. 2018. Scratch - About. (2018). https://scratch.mit.edu/about
4. Saleema Amershi. 2011. Designing for effective end-user interaction with machine learning. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*. ACM, 47–50.
5. Francisco Bernardo, Michael Zbyszynski, Rebecca Fiebrink, Mick Grierson, and others. 2016. Interactive Machine Learning for End-User Innovation. American Association for Artificial Intelligence (AAAI).
6. Michelle Patrick Cook. 2006. Visual representations in science education: The influence of prior knowledge and cognitive load theory on instructional design principles. *Science education* 90, 6 (2006), 1073–1091.
7. Jordan Aiko Deja, Patrick Arceo, Darren Goldwin David, Patrick Lawrence Gan, and Ryan Christopher Roque. 2018a. MyoSL: A Framework for Measuring Usability of Two-Arm Gestural Electromyography for Sign Language. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 146–159.
8. Jordan Aiko Deja, Rafael Cabredo, and Toni-Jan Keith Monserrat. 2018b. On Building an Emotion-based Music Composition Companion. In *Proceedings of the Asian HCI Symposium'18 on Emerging Research Collection (Asian HCI Symposium'18)*. ACM, New York, NY, USA, 25–28. DOI:http://dx.doi.org/10.1145/3205851.3205859
9. Jordan Aiko Deja, Kevin Gray Chan, Migo Andres Dancel, Allen Vincent Gonzales, and John Patrick Tobias. 2018c. Flow: A Musical Composition Tool Using Gesture Interactions and Musical Metacreation. In *International Conference on Human-Computer Interaction*. Springer, 169–176.
10. Steven Halim, Zi Chun Koh, Victor Bo Huai LOH, and Felix Halim. 2012. Learning Algorithms with Unified and Interactive Web-Based Visualization. *Olympiads in Informatics* 6 (2012).
11. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. DOI: http://dx.doi.org/10.1145/1656274.1656278
12. Jabra. 2018. use wordnet and word2vec on dataset. (Apr 2018). https://community.rapidminer.com/t5/Getting-Started-Forum/use-wordnet-and-word2vec-on-dataset/td-p/48671
13. Teng Lee, James Johnson, and Steve Cheng. 2016. An Interactive Machine Learning Framework. *arXiv preprint arXiv:1610.05463* (2016).
14. Jovanovic Milǒs, Vukicevic Milan, Delibăsic Boris, and Suknovic Milija. 2013. Using RapidMiner for Research: Experimental Evaluation of Learners. *RapidMiner: Data Mining Use Cases and Business Analytics Applications* (2013), 439.
15. Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 249–256.

16. Ibrahim Ouahbi, Fatiha Kaddari, Hassane Darhmaoui, Abdelrhani Elachqar, and Soufiane Lahmine. 2015. Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences* 191 (2015), 1479–1482.

17. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830. `http://dl.acm.org/citation.cfm?id=1953048.2078195`

18. Margaret Rouse. 2005. What is sandbox? - Definition from WhatIs.com. (Sep 2005). `https://searchsecurity.techtarget.com/definition/sandbox`

19. Advait Sarkar, Mateja Jamnik, Alan F Blackwell, and Martin Spott. 2015. Interactive visual machine learning in spreadsheets. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on*. IEEE, 159–163.

20. Douglas Schuler and Aki Namioka. 1993. *Participatory design: Principles and practices*. CRC Press.

21. Clifford A Shaffer, Matthew L Cooper, Alexander Joel D Alon, Monika Akbar, Michael Stewart, Sean Ponce, and Stephen H Edwards. 2010. Algorithm visualization: The state of the field. *ACM Transactions on Computing Education (TOCE)* 10, 3 (2010), 9.

22. Daniel Smilkov and Shan Carter. 2016. Tensorflow - Neural Network Playground. (Apr 2016). `https://playground.tensorflow.org/`

23. Daniel Smilkov, Shan Carter, D Sculley, Fernanda B Viégas, and Martin Wattenberg. 2017. Direct-manipulation visualization of deep networks. *arXiv preprint arXiv:1708.03788* (2017).

24. John Edel Tamani, Jan Christian Blaise Cruz, Joshua Raphaelle Cruzada, Jolene Valenzuela, Kevin Gray Chan, and Jordan Aiko Deja. 2018. Building Guitar Strum Models for an Interactive Air Guitar Prototype. In *Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia, CHIuXiD '18 (CHIuXiD '18)*. ACM, New York, NY, USA, 18–22. DOI: `http://dx.doi.org/10.1145/3205946.3205972`

25. Waikato. 2017. Json format display error. (Mar 2017). `http://weka.8497.n7.nabble.com/Json-format-display-error-td39863.html`