DEGREE PROJECT IN INFORMATION AND COMMUNICATION TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
*STOCKHOLM, SWEDEN 2020*

# User-centred Design for Input Interface of a Machine Learning Platform

**ADITYA GIANTO HADIWIJAYA**

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
**SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

## Abstract

Although its applications have spread beyond computer science field, the process of machine learning still has some challenges for both expert and novice users. Machine learning platform aims to automate and accelerate the delivery cycle of using machine learning techniques. The objective of this degree project is to generate a user-centred design for an input interface of a machine-learning platform. To answer the research question, there are three methods conducted sequentially: 1) interviews; 2) prototyping; and 3) design evaluation.

From the initial interview, we concluded users' problems and expectations into 11 initial design requirements that should be incorporated into our future platform. The prototype testing focused on checking and improving the functionalities, rather than the visual appearance of the product. Finally, in the design evaluation method, the research delivered design recommendations consisting of five implications: 1) start with a clear definition of the specific machine learning goal; 2) present states of machine learning with a straight-forward flow that promotes learning-opportunity; 3) enable two-way transitions between all states; 4) accommodate different users' goals with multiple scenarios; and 5) provide expert users with more control to customize the models.

## Keywords

Machine learning, machine learning platform, input interface

# Abstract

Trots att dess tillämpningar har spridit sig utöver datavetenskapliga fält, behöver utvecklingen av framgångsrik användning av maskininlärning fortfarande anspråkiga komplexa metoder. Maskininlärningsplattform syftar till att automatisera och påskynda leveranscykeln för att använda maskininlärningstekniker. Syftet med detta examensarbete är att generera en användarcentrerad design för ett ingångsgränssnitt för en maskininlärningsplattform. För att besvara forskningsfrågan finns det tre metoder som genomförs i följd: 1) intervjuer; 2) prototypning; och 3) designutvärdering.

Från den första intervjun avslutade vi användarnas problem och förväntningar i 11 ursprungliga designkrav som bör integreras av vår framtida plattform. Prototyptesten fokuserade på att kontrollera och förbättra funktionaliteterna snarare än det visuella utseendet på produkten. Avslutningsvis, i designbedömningsmetoden, levererade forskningen designrekommendationer bestående av fem implikationer: 1) börja med en tydlig definition av maskininlärningsmålet; 2) nuvarande stater med ett rakt framåtflöde som främjar inlärningsmöjligheter; 3) möjliggöra tvåvägsövergångar mellan tillstånd; 4) Rymma olika användares mål med flera scenarier; och 5) ge experter användare mer kontroll.



## Nyckelord

Maskininlärning, maskininlärningsplattform, ingångsgränssnitt

## Acknowledgement

I would like to thank Charles Windlin, my academic supervisor from KTH, and Ather Gattami, CEO of Bitynamics AB for giving me the opportunity and guidance during my struggle with the thesis. It was a great experience and I learned a lot about both interface design and machine learning. I believe the subject will continue to develop and Bitynamics will contribute significantly to the respective field.

I would also like to mention Ria Ratna Sari and Sara Hedar for becoming reliable colleagues during the early time of this project when things were still very abstract. They did an amazing job by continuously brainstorming and emotionally-supporting each other towards our final goal. EIT Digital and University of Twente also deserve my gratitude for allowing me to pursue my master degree in a major that I'm very passionate about, Human-Computer Interaction Design.

# Contents

# 1. Introduction

## 1.1. Background

Machine learning is a universal trending application of artificial intelligence by providing systems an ability to learn from given dataset and deliver a specific function from its analysis. The functions currently vary from spotting pattern in drug detection, generating a credit score, forecasting stocks, classifying images in fraud detection and many more. Although its applications have spread beyond computer science field, the process of machine learning still has some challenges for both expert and novice users. This is strongly related to its wide selection of algorithm and even parameters that users usually need to adjust iteratively before they can produce good results. Not only those who have no prior AI-knowledge think that machine learning is unreachable, but also expert users believe machine learning is challenging and tricky [1].

In machine learning, different models could perform differently based on its used algorithms, parameters and dataset, however the best-learning algorithm does not exist [1]. Thus, machine learning is a cycle of trial-and-error by iteratively training and comparing the performances of generated models. In command line-based platforms where the models are developed by typing and executing blocks of code sequentially, this iterative cycle can be overwhelming for both data scientists and business domain experts. Some more technical problems also include difficulties in setting up the working environment, lacking knowledge of development tools, multiple approaches to reach the same goal and confusions when picking up languages or libraries [1].

Following current challenges in machine learning, providing a machine learning platform has been a growing business in PaaS (Platform-as-a-Service) industry. It works as an efficient management tools throughout the lifecycle of machine learning models. Bitynamics[1] is a Stockholm-based AI development company that initiates a research for a machine-learning platform that emphasizes on simplicity, user-friendliness and affordability without leaving the performance of its generated models. This platform includes both novices and experts as target users and aims to tackle the complexities in machine learning process.

To make this idea come to reality, this degree project acts as a research that focuses only on the input interface design of the platform. As a future work, the input interface will be

[1] https://bitynamics.com/#home

integrated with other two projects exploring the model training (back-end) and data visualization (output interface). However, the integration itself is not the scope of this degree project.

## 1.2. Objective

The desired objective of this degree project is to generate a user-centred design for an input interface of a machine-learning platform.

## 1.3. Research questions

The research question is defined as **How should the input interface of a machine learning platform be designed to deliver common users' expectations and tackle their challenges in building effective machine learning systems as measured by interviews, prototyping and design evaluation method?**

## 1.4. Scope

The scope of this degree project is the input interface of Bitynamics' machine-learning platform. The input interface mainly consists of three steps that capture and pass users' input to the training process in the back-end. These steps are 1) starting a machine learning project; 2) preparing the dataset; and 3) choosing a model to train. As we aim to fulfil the expectations of users with wide range of expertise level, the interface should align to usability rules and user-centred design principles.

## 1.5. Methodology

The examination of expectations and problems mainly focus on the result of user interview following to defining the profile of our specific target users, benchmarking the competitors, and conducting literature research. From this method combined with literature study and benchmarking, we concluded initial design requirements that should be incorporated into our future platform.

Subsequently, paper prototypes were used to iteratively check, test and improve the functionalities at the early stage. The prototype testing used Wizard of Oz technique that came from a human acting as the substitution of an algorithm in linking the interfaces between different papers. During testing, participants followed a think-aloud protocol during their interaction with the prototypes to perform a set of specified tasks.

Finally, in the design evaluation method, the interactive prototype was then composed to represent the actual platform. To gain our final design recommendations, the iterative evaluations of interactive prototype required participants to deliver three tasks and again, follow a think-aloud protocol. During the testing, participants left specific comments on tasks, interface elements or interactions, in which the feedbacks were then evaluated to improve the prototype.

## 1.6. Evaluation and news value

To answer the research question, the degree project delivers a set of implications for designing an input interface of a machine learning platform with systematic elaborations that involve all previous findings. Some relevant artefacts made during the degree project will be included as appendixes. This degree project mainly contributes to the input interface design principles of a machine learning platform and adds to the collection of references regarding user interface and experience design in software development.

# 2. Theoretical background

This chapter presents some literature studies conducted prior to the research. The studies focus on relevant topics that explain better about machine learning, its state-of-the-art, previous study of machine learning platforms and their design challenges.

## 2.1. Machine learning

Machine learning is a branch of computer science that focuses on building a system capable of iterative-learning over time in autonomous fashion by processing data and other relevant information from observations and real-world interactions.  Machine learning roots in statistics as its art of data extraction.
The general classifications of machine learning consist of two main types based on two problem types to solve: 1) classification that aims to separate the dataset into different categorical classes or discrete values; and 2) regression to map the input values in the dataset into continuous output values. As the expected goal, the final model of a machine learning process must solve either one of these two major problem [2].

Despite being various in applications, all machine learning techniques share similar states of work: 1) *data collection*, a process of gathering sufficient, yet relevant amount of data; 2) *data preparation*, consisting of some processes such as normalization and removal of duplicated, error or empty data to make the dataset meaningful for the model; 3) *choosing a model*, a step of determining the type and algorithms used in generating a mathematical model which satisfies the goal using provided datasets; 4) *training*, the bulk stage that uses the provide dataset to incrementally improve the performance of the model by updating its inferences, weights and biases; 5) *testing*, a performance evaluation-stage by using the generated model to unused dataset; 6) *tuning parameters*, an experimental process to refine the model by changing algorithm settings; and 7) *generating predictions*, a final stage to export the model and answer the questions [2].

Each of the technical states mentioned before needs to be executed correctly to avoid frauds. When users prepare the dataset and choose the model, different algorithms come with specific attributes to be manipulated. A fraud happens when users enter invalid attributes to the selected parameter as this action become non-executable or malfunctioned. Thus, it is very important to make sure that the dataset, selected algorithm and manipulated attributes are coherent to achieve the expected goal [3].

## 2.2. Machine learning platforms

Machine learning platform is a platform for automating and accelerating the delivery cycle of using machine learning techniques [4]. In business context, the platform is often interchangeably referred as Machine learning as a service (MLaS). Machine learning platforms makes expertise in machine learning itself not essential to the training process because less human interferences are required. An appropriate interface design for the platform should reduce users' effort by making the techniques more accessible as a human-computer interaction (HCI) task. Some popular approaches to consider are by including the intuitiveness and interactivity of the tasks to generate, inspect and correct the model. The ultimate interactivity achieved when the user and target model directly influence each other's behaviour [5].

Furthermore, the platform is a concept with promising potential to reduce overall costs as it enables resource sharing and allocation. As the platform access should be granted to multiple users, well-defined interface is an important element to provide a scalable, flexible and non-blocking platform with service-oriented architecture (SOA) [6].

The difficulties of incorporating machine learning in small-medium enterprises (SMEs), developers and research institutes are far beyond either understanding algorithm or obtaining relevant data. To face the steep learning curve of machine learning, they require computational resources to store data and models that consume impracticable storage space as well as their cost [4].

The state-of-the-art in machine learning nowadays has enabled a web user interface that aims to simplify machine learning. Both Google Cloud Platform and Microsoft Azure provides graphical assistance to their users in different stages of the input process. This concept is called as an automated machine learning that reads users' imported data and simply applies multiple combinations of algorithm and hyper-parameters to generate the best model.

## 2.3. Challenges and guidelines in designing a machine learning platform

Designing the interface for a machine learning platform has four key challenges: 1) dataset can be imprecise and inconsistent; 2) degree of users' uncertainty in determining the expected output and algorithms based on provided dataset; 3) interacting with a model is not as intuitive as information with conventional structure, such as the dataset or parameter settings; and 4) training is an open-ended process and simply impossible to be 100% accurate.

In a machine learning platform, users iteratively build and refine the mathematical model that describes the underlying concept of a problem. The involvement of human intelligence in this loop provides periodic improvements and realignment to the initial objective through multiple reviews. However, since users have to deal with the previously mentioned challenges, this behaviour triggers a paradox as it could be both useful and tricky at the same time [5].

Furthermore, the machine learning process is also very diverse in terms of complexities, goals and methods. Thus, a machine learning platform that allows the users to switch between different algorithms and model structures while maintaining similar interfaces becomes an intended goal as it could provide the flexibility and avoid the users from refamiliarization cost. With this ultimate goal in mind, an ideal machine learning platform should enable users to leverage multiple models, ensemble different functionalities and compare performance before obtaining the expected result.

After the seamless transition within multiple models and algorithms in the platform has been afforded, how to assist the involvement of users in a well-manner might come as the next concern. There are several approaches, such as minimising decision points by only including appropriate operators, providing cascading changes to the processes and making all stages reachable to enable trial-and-error [5].

In addition to this, there are three important aspects for a machine-learning platform: 1) an illustrative current state of the learned concept; 2) a clear guidance for users to provide input that improves the concept; and 3) a revision mechanism that allows users' exploration before ensuring them that the model will suffice to address a typical problem by using different datasets. [6]

## 2.4. Related algorithms in this research

Apart from the previous literatures that give a general overview of the field, this study also includes some algorithms used in the platform as they are taken into considerations when designing the interface in the later step. However, the execution of each algorithm in the training process is out of context in this degree project.

A perceptron, also called a neuron or node, is a binary classifier that has one or more weighted input connections, a transfer function that combines the inputs and an output connection. *Multilayer Perceptron* (MLP) consists of at least three fully-connected layers: an input layer, a hidden layer and an output layer. In MLP, each neuron in one layer is connected to all neurons in the next layer. The training occurs in each of the perceptrons by iteratively changing connection weights based on the amount of error between output and the expected result in supervised-learning. During the iteration, the node weights are adjusted based on corrections that minimize the error [7].

*Convolutional Neural Network* (CNN) applies some pre-processing before the conventional MLP.[8] Using the convolution, CNN breaks the raw dataset into matrices and operates matrix-multiplications to capture only the important features of the dataset [9]. To capture complex dataset, the number of layers may need to be increased for including low-levels details, yet at the cost of more computational power [10].

As opposed to MLP and CNN that use feedforward neural network in which the connections between nodes do not form a cycle, Long Short-term Memory (LSTM) uses recurrent neural network that enable temporal dynamic behaviour and feedback connections. Thus, a LSTM network suffices to process time-series datasets, such as voice or video recognition. A common LSTM architecture is composed of a *cell*, the memory part of an LSTM network that keeps track of the dependencies between the elements. The concept of LSTM bases on the gates that learn which information is relevant to keep or forget during training [11].

# 3. Methodology

This degree project relies on an inductive approach of three qualitative empirical research methods. This chapter only includes elaboration of each method while the result is presented on the next chapter “Results”.

## 3.1. Initial interview

### 3.1.1. Interview design and procedure

In this degree project, semi-structured interviews [26] were used at the beginning to obtain participants’ current thoughts in the machine learning context. The specific goal of this method was to understand their problem and expectations. The interviews were conducted to six participants aged 24-34 years old with different expertise levels through face to face interviews. In order to guarantee the representativeness of the sample, two participants were recruited from Bitynamics’ recommendation who did not have any affiliations to the company, one participant was Bitynamics’ intern, while three were students in KTH. All participants did not receive any rewards from doing the interviews. To indicate their expertise level, participants answered three introductory questions regarding their previous experiences related to machine learning.

In accordance with the expected information, the interviews consisted of three semi-structured parts: 1) introductory (answered by all participants); 2) previous interactions with machine learning platform (answered by users with prior machine-learning experiences); and 3) expected aspects that a machine learning platform should incorporate (answered by all participants). The interviews were conducted separately for 20-30 minutes each participant.

### 3.1.2. Interview questions

In the introductory part, each of the participants informed their previous experiences in machine learning field with three introductory questions: 1) if they have built an ML model at least once; 2) if they have used ML in a professional context; and 3) if they have completed an education level in ML-related fields.

We began the second part by asking users their mostly-used platform. Following this, we asked a guided question regarding the stages of machine learning that the platform covers. As the available options, we familiarized participants with the general terminologies for seven stages of machine learning mentioned in Chapter 2. Apart from giving us some referred platforms to benchmark, this question also briefly triggered participants’ memory with machine learning.

To gain meaningful insights, we then asked what users like and dislike from the platform, also how they currently deal with what they dislike.

The third part of these interviews focused on uttering users' expectation when they use a machine learning platform. We also provided some keywords that were taken from aspects that a computer software generally considers important, such as automation, ease-of-control, explicit instructions, task overview, recommendations of actions and performance comparison. For expert users, we added some domain-specific questions regarding how some particular stages could be improved.

From the user interviews combined with benchmarking and literature study, we concluded list of functionalities, features and guidelines that our future platform should incorporate to help our users reach their ultimate goal: building a machine learning model.

## 3.2. Prototyping

### 3.2.1. Paper prototype design

Prototypes are tangible expressions of design intent. We used paper prototypes in this stage. Paper prototype is a crucial part in the early stage of the design process as it works as an early sample to test if our idea is conceptually correct [28]. Paper prototype is designed to be made, evaluated and improved in a short, yet fast cycle. In this degree project, paper prototype would be used as low-fidelity prototype. This stage focused on checking, testing and improving the functionalities, rather than the visual appearance of the product. The interactivity in this paper prototype used Wizard of Oz technique that came from a human acting as the substitution of algorithm in linking the interfaces between different papers [24]. To systematically visualize the list of functionalities obtained from our paper prototype iterations, we provide you with a sitemap as a deliverable.

### 3.2.2. Paper prototype testing

The prototype was tested to a novice, an expert user and the CEO of Bitynamics AB as the main stakeholder of this project. The paper prototype worked as a simplified representation to communicate how the platform would work and trigger conversations that captured specific feedbacks from our target users. During the testing, participants interacted with the prototypes to perform a set of specified tasks: 1) create new project; 2) upload dataset; and 3) choosing models. Participants had to follow think-aloud protocol that obliged them to say whatever inside their mind as they completed actions. This included what they were looking at, thinking, doing, and feeling as we were interested to observe participants' cognitive processes rather than only their final decision to act. To prevent losing remarkable thoughts, we only took notes without attempting to interpret and assist. This project conducted three sessions of low-fidelity testing before generating its final version.

## 3.3. Design evaluation

### 3.3.1. Interactive prototype design

As opposed to a paper prototype, visual design matters significantly in an interactive prototype.[28] In this project, the interactive prototype was a high-fidelity medium that appeared and functioned as similar as possible to our future machine learning platform. To resemble participants' interaction with the actual platform, we used the final interactive prototype instead. As we already built solid ground regarding the functionalities, interactive prototype included design details, such as elements, spacings, colour palette, icons, graphics, typography and placement in each screen interfaces of the prototype. In addition to this, the prototype previewed content that would appear in the actual platform.

### 3.3.2. Interactive prototype testing

The evaluation of our interactive prototype was conducted to six people aged 22-40 years old during the appeal for social distancing in Stockholm, Sweden. Thus, our testing was done and recorded via Zoom[2] application with all participants' consents. In the session, each of six participants interacted with the interactive prototype[3] made and shared with Figma[4] platform. The interactive prototype had similar appearance and functionality to the actual platform. During the testing, the participants were asked to think-aloud or say their thoughts, such as impressions or reasons why they took such actions. While they were working with the task, researcher was not allowed to interrupt, suggest certain actions or answer participants' questions. This rule happened to maintain the context authenticity when actual users use the platform without any supervision. At the end of each task, the participants had to answer relevant questions regarding their general satisfaction, comments and feedbacks. To obtain meaningful opinions, some questions were followed up by elaborative inquiries. In the

[2] https://zoom.us/
[3] https://bit.ly/figma-prototype-input-bitynamics/
[4] https://www.figma.com/

evaluation, participants could also comment on specific interface elements or interactions, such as affordance of icons or animations, in which the feedbacks were then elaborated and analysed. In the evaluation, participants could also comment on specific interface elements or interactions, such as affordance of icons or animations, in which the feedbacks were then elaborated and analysed.

Similar to the paper prototype testing, participants were obliged to deliver three tasks and follow think-aloud protocol. The interaction was expected to be more natural as if they were interacting with the final platform. To ensure the testing covers holistic goals of the platform, participants were asked to perform the tasks based on their expertise level following the scenarios in Table 1. As three of the participants were considered as expert users based on their answers regarding previous experience in machine learning field, they were assigned for expert users' task set while the rests did the novices' set.

| Task | Instructions for novice users | Instructions for expert users |
|---|---|---|
| Task 1 | Create new machine learning project using this platform. | |
| Task 2 | Create an image classification project and upload a dataset. | |
| Task 3 | Change the project into a tabular classification project and start training the model. | Change the project type into a tabular classification project and train the model with Multilayer Perceptron algorithm. |

*Table 1: Different task instructions during the interactive prototype testing with novice and expert users*

Each of the performed tasks allowed users to interact with different parts of the prototype and thus, consecutively they covered all elements that belong to our research scope. As a final deliverable that answers to the research question, we composed a set of recommendations in the form of design implications that guide the design process of input interface of a machine learning platform.

# 4. Results and discussions

## 4.1. Initial interview

From the first part of each interview, participants were asked to indicate their profile as seen in Table 2. This includes their expertise level and general purpose when they worked with machine learning (ML). The interview involved representative samples for our target group, from novice to expert ML users.

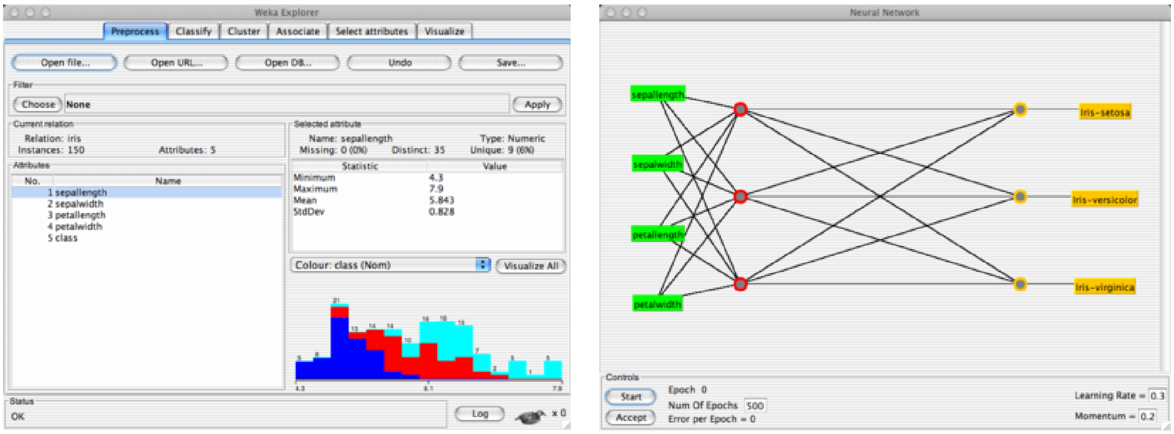
| No. | Participant | Built an ML model at least once | Used ML in professional context | Completed education in a ML-related field |
|---|---|---|---|---|
| 1. | Participant A | No | No | No |
| 2. | Participant B | Yes | No | No |
| 3. | Participant C | Yes | Yes | No |
| 4. | Participant D | Yes | Yes | Yes |
| 5. | Participant E | Yes | Yes | Yes |
| 6. | Participant F | Yes | Yes | Yes |

*Table 2: Obtained profiles of interview participants*

### 4.1.1. Prior experiences with machine-learning platforms

Going to the second part, five of our participants who had prior experiences to build an ML model shared their previous interactions with the respective platforms. The question was then followed up with more detailed inquiries based on each of their answers: Which stages of machine learning did the platform provide assistive functionality? What do you like? What do you dislike from the platform? The participants were also provided with relevant keywords regarding the ML stages to trigger their memory. The answers contributed some benchmarking insights and enabled participants to put themselves in relevant contexts when they were trying to deliver ML-related tasks with other platforms.

Jupyter Notebook was mentioned by all five experienced participants. Jupyter[5] is an open-source, interactive web tool that enables users to type lines of code, see computational output, explanatory text and multimedia resource in a single document. It is compatible to three top programming languages: Julia, Python and R. As a cloud-based platform, it facilitates access to remote data that might not be feasible to keep in local storage. The main advantage of Jupyter Notebook is its computational narrative or ability to let scientists supplement their code and data with analysis, hypotheses and conjectures at the same file. This premise was approved by all five participants although Participant C, D and E added remarkable notes regarding its slow debugging process since they had to evaluate each of the lines. Furthermore, Jupyter's behaviour to reuse modules (sets of code lines) tend to make users run code cells out of order and generate unexpected error [12]. To summarize, Jupyter provides flexibility to customize the algorithm that generates the expected machine learning model at the expense of prone-to-error actions. By allowing users to write anything, the platform lacks of assistance to guide users on which code is executable or not.




(a) Explorer window with six tabs as the main entrance point

(b) Customizing algorithm can be done by changing some parameter values

*Figure 1: Graphical User Interfaces of WEKA*

Waikato Environment for Knowledge Analysis (WEKA) is a free software that allows users to execute various visualization tools and algorithms with a simplifying graphical user interfaces (GUI). Although the GUI currently looks obsolete, Participant D mentioned WEKA was a

[5] https://jupyter.org/

helpful, transparent platform when they were firstly introduced to ML. It displays not only the final result but also the process. WEKA provides an Explorer window with six tabs that represent different functionalities for ML: pre-process, classifiers, clusters, associations, attribute selections and visualizations. WEKA also provides limited interactivity in which Participant D once found it useful to manually customize one of the algorithm by setting certain parameter values. Participant D explicitly stated, “I would like to have this interactivity for all algorithm since it significantly helped me to experiment with the process.”

Cloud AutoML is part of Google Cloud Platform that provides ML at the click of buttons and was mentioned by three participants. Participant B and C suggested that the platform enabled them who had limited ML expertise to generate sufficient models for their business-oriented goals. Participant D added that working with AutoML was obviously less time-consuming than Jupyter. Basically, users only need to upload proper number of samples and run the training. This straight-forward flow addresses common pain points by reducing both human errors and time for doing research and learning about the best practices. Despite these advantages, Participant D claimed the pitfall was that he had no idea about the process when he finally managed to get a well-performing model. As an ML expert, he felt clueless and lacking sense of control about his own work because his only involvement was providing the dataset and clicking ‘Train’ button.

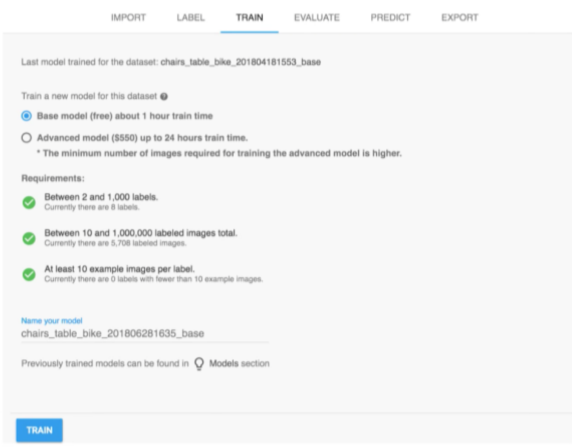



*Figure 2: Graphical User Interface of AutoML*

#### 4.1.2. General expectations

The third part of interviews required participants to utter their expectations. To trigger contextual responses, the interview was initiated by providing choices and asking participants to pick the aspects that an ideal platform should include. Table 3 shows participants’ preferences. Since each of the keywords were vaguely defined, we followed up with elaborative questions to understand what participants meant for their selected aspects.

| Aspects | Participants | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| Automation | **v** | **v** | | | | |
| Explicit instructions | **v** | **v** | **v** | **v** | **v** | **v** |
| Ease-of-control | **v** | | **v** | **v** | **v** | **v** |
| Recommendations of action | **v** | **v** | **v** | **v** | **v** | **v** |
| Task overview | | | | | | **v** |

*Table 3: Interview participants' preferences toward expected aspects in an ML platform*

Participant A used to act as a domain-expert who worked closely with a data scientist to generate a business-oriented ML model. Participant A had no technical knowledge and realized that the extraction process of a large-scale dataset had more challenges than recognizing useful patterns and insights. From the teaming-up experience, Participant A understood that there were an overwhelming number of ways, techniques and methods to train the dataset and all of them would still result on reasonable models. As argued by Participant B, the preliminary researches to decide them always relied on the best practices of similar use cases. Additionally, Participant B revealed that many models shared fixed pipeline that also works generally. Apart from providing the correct dataset, they proposed that the automation could have minimized human involvement in deciding and executing the most suitable techniques.

All participants agreed to include explicit instructions in the platform. The actual design suggestions may vary but it is vital to have clear user guides. Participant E added that the instruction should also be non-obtrusive and allows users to focus on their ML goal. Explicit instructions avoid ambiguity for novice users who still cannot make inferences from their current circumstances and let all users put least effort on following the correct path. Furthermore, having step-by-step directions from an ML platform had helped Participant B to learn its general process.

Ease-of-control consisted of two components: ease to minimize the difficulty level of tasks and control that enables users to feel power. Participants suggested that providing the process with customization and explanation are elements of control that provide them with sense of giving direct influences. To add a remarkable note, explorative actions that allow executions of different command lines made Participant F still prefer Jupyter Notebook with complexity at stakes. Participant F found the idea to incorporate 'ease' factor in a flexible platform could have improved the performance significantly.

Recommendations of actions mainly had two different reasons to be included as an important aspect. Participant A expected recommendations could serve as a practical assistance to sequence of steps that delivers the whole task in a good manner. Participant A believed that having recommendations was crucial for novice users' learning process and their success level of utilizing the platform to reach the goal. This interpretation overlapped with the explicit instructions as they both shared mutual characteristics. On the contrary, Participant B, C, D, E and F considered recommendations as a complementary element that supports them in improving the quality of models. In addition to this, recommendations could minimize options of decision by showing prioritization or eliminating irrelevant scenarios. As opposed to a mandatory guidance, Participant D and F claimed "recommendations is more a 'nice-to-have' thing" that they might have decided to follow or not based on the circumstances. Despite this diversity of motive, recommendations of actions were added to the list by all participants.

Task overview was proposed by Participant F to illustrate clarity toward the states of process. With this aspect, users might have got better picture on more than just their current situation, but also what a state means as part of the process journey. Participant F argued that the importance of task overview is even more significant with business-oriented approach since it simplifies users' task to plan and allocate resources.

### 4.1.3. Initial design requirements

During the design process, I worked closely with two other students whose responsibilities were to explore the ML process and the output interface respectively. To avoid overlapping, the scope of input interface was defined as platform functionalities that support data collection, data preparation and choosing the model before a training starts. Table 4 concluded the findings from our initial interviews into 10 initial design guidelines that our subsequent design process should follow.

| No. | Design guidelines | Description |
|---|---|---|
| 1. | Provide cloud-based data storage | It enables access to big data that is not feasible to store in local disk. |
| 2. | Declare explicit instructions | Clear user guides reduce users' workload and support a learning-by-doing experience. |
| 3. | Use graphic user interface (GUI) | In general, it eases users to deliver the tasks. Additionally, this increases affordability to users without any programming skills. |
| 4. | Include transparency | Transparency helps users learn the process. |
| 5. | Promote straight-forward flow | It is less time-consuming and minimizes possibility of human-errors. |
| 6. | Provide interactivity | It triggers user to experiment and explore. |
| 7. | Enable flexibility to customize the model | It adds sense of control and satisfaction for users with sufficient expertise level. |
| 8. | Avoid prone-to-error actions | In a qualitative method like interview, each sample represents variation in our target group. Summarizing the interview result, we realized that automation is a suitable approach for novice users to reach this design goal. |
| 9. | Recommend actions for improving models | It overcomes users' problem with picking options to improve the model performance, such as choosing more explainable algorithms, benchmarking to best-practices or using the complexity level of the |

| | | |
|---|---|---|
| | | dataset. |
| 10. | Illustrate current state of the process | To bring task overview in a more general term, helping users to understand their current state is beneficial to enrich their perspective and planning. |

*Table 4: Preliminary design guidelines from participants' problems and expectations*

Before further development into a paper prototype, a sitemap was made to effectively manage the content and functionality according to the technical requirements of a machine learning process and our design guidelines. Using sitemap as a blueprint has the advantages to promote straight-forward workflow that simultaneously also avoid users from doing prone-to-error actions.

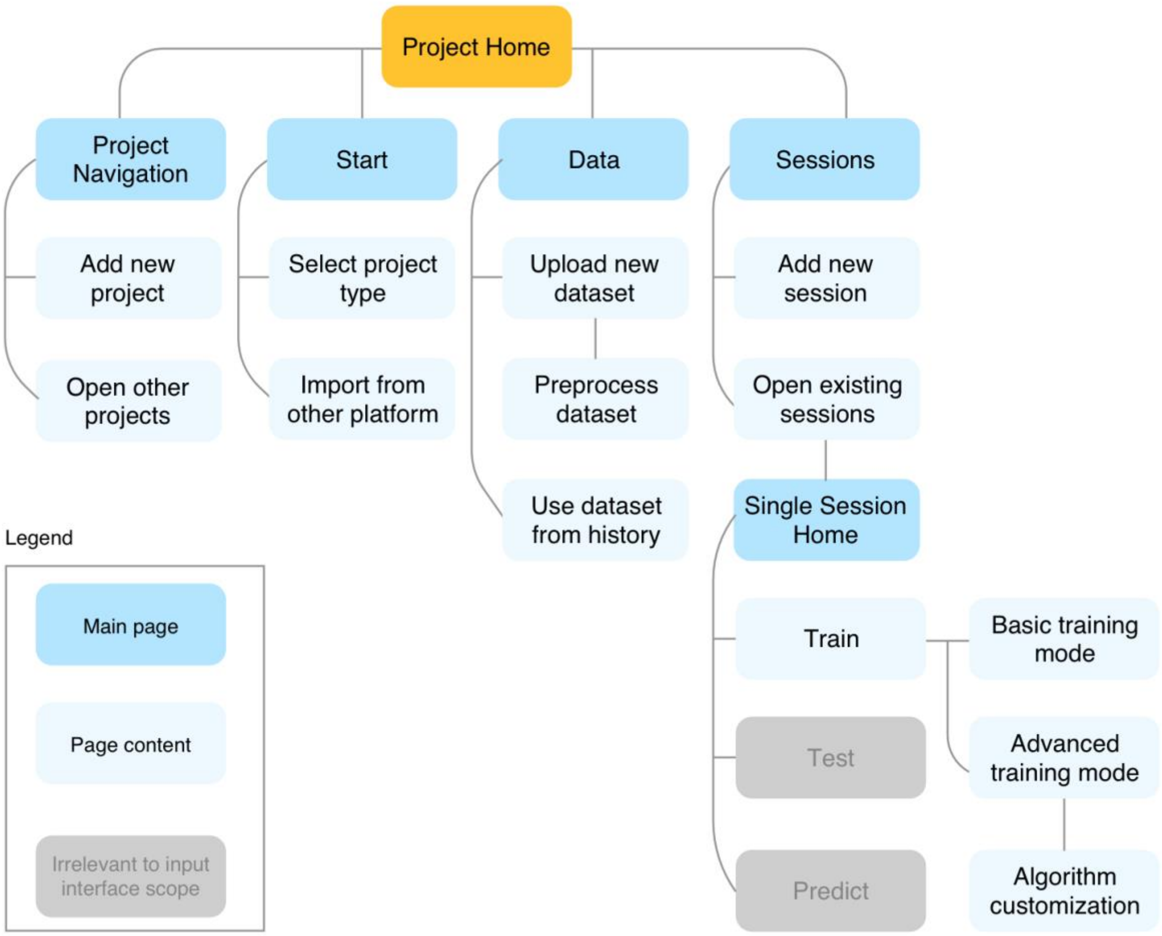



*Figure 3: The sitemap of our designed input interface*

As seen in Figure 3, the platform would organize users' tasks as Projects. To add, edit and open projects, users can simply access the menus under Project Navigation page. The Start page acts as a home screen where users can select a project type. Based on the input and output types, there are five available options for project types: tabular classification, tabular regression, image classification, time-series classification and time-series regression. Each project type is a different template that has been assigned with the relevant sets of ML algorithms, parameters and other customized actions. Using template avoids user from a non-executable command that may lead to an error. As a complementary feature that aims to attract expert users by providing simple migration tasks, our platform is able to import their previous ML works generated by Jupyter Notebook or other command-line based platforms.

Subsequently, the Data page is supposed to be users' control panel for collecting and preparing the dataset. Users are allowed to choose between uploading a new dataset or using one that has been stored in the platform. Based on the selected project type, the platform will recommend relevant pre-processing actions that enable the platform to read the dataset as an input. This maintains transparency and sense of control as it will always ask users' preference and confirmation before any actions are carried out.

In the platform, Sessions are defined as the subordinate components of Projects that holds single Train, Evaluate and Export pages together. Inside a Project, users can generate multiple Sessions that use identical dataset and generate same output type. This organization optimizes the benefit of a cloud-based data storage as same project and dataset settings become reusable. If users want to improve current models' performance, they can skip the previous steps, create a new session and experiment by tuning some parameters.

The Train page serves as an interface for choosing and building a desired model. Here, the design guidelines that elaborate the opposing ideas of flexibility for expert users and automation for novices should be manifested proportionally. To accommodate these contradiction, the platform uses segmentation strategy by providing two modes: Basic and Advanced training mode. Basic mode provides an automated service that applies several relevant algorithms to train the data based on the best practices and returns a final model with highest accuracy. Primarily designed for novice users, the Basic mode avoids prone-to-error actions and reduces the workload to conduct prior researches, a time-consuming effort that expert users might need to do as well. The Advanced training mode, on the contrary, provides users with access to customize the relevant parameters, such as algorithm type, architecture and other training settings. This mode aims to maintain flexibility, transparency and a sense of control that advanced users used to own in command line-based platforms. At the end of each mode selection, the users are then set to start training the models.

## 4.2. Prototyping

### 4.2.1. Paper prototype design

From the sitemaps, we got clear structure of our web-based platform. Thus, we could now design each of our pages. Following the design guidelines, GUI are used to simplify the tasks and increase affordability of machine learning to broader user group. To provide interactivity, we design an interface that enables user to easily access other pages with a seamless transition as argued in the principal design for correction-interfaces [13].

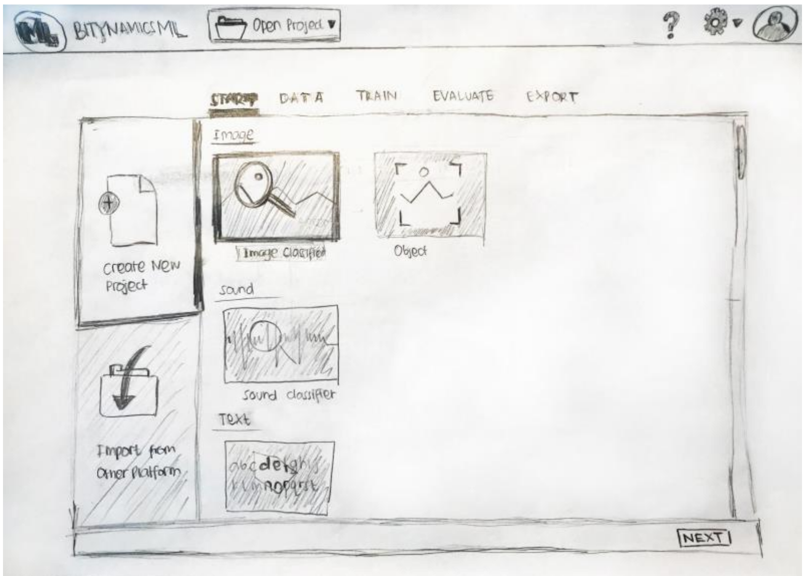
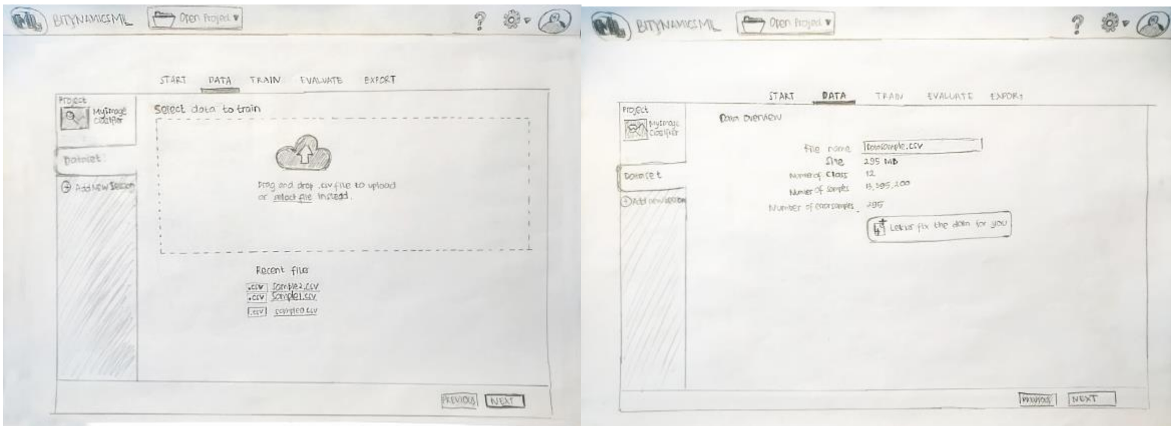



*Figure 4: Start page as the main entrance point of our designed interface*

The Start page in Figure 4 shows the embodiment of our previous concepts in a visual low-fidelity representation. To switch easily within the platform, there are two main functionalities: 1) Project Navigation; and 2) Stage Indicator that contains five buttons (Start, Data, Train, Evaluate and Export) to let users open different stages in a project. You will soon notice that those functionalities remain in an absolute position on the top area of other pages. Below the Stage Indicator, users begin the process by determining their project types based on their output and input. Having this clear project type as an expected final goal helps user to include their intention with the platform from the very early stage. Besides, getting well-informed regarding users’ determined intents provide the platform with sufficient base to decide which assistances are toward their satisfaction [14]. As selecting the project type is our main task in this page, we represent each project type in a universal visual language with recognizable icons that draw interests faster. To ensure good elaboration, we keep the textual name of each project type as labels below the icons.

Subsequently, the interface design for Data page in Figure 5 enables user to choose which dataset should be processed in the further steps. Apart from the two navigation functionalities that we keep from the Start page to maintain consistency, there is now another sidebar that provides users with project name, datasets and sessions. As a project might hold multiple sessions, users can navigate by switching session here. By default for a new project, the page shows an upload area with a relevant icon and instruction on how users can use it. If users decide to upload a new file, the GUI reads a drag-and-drop interaction by device gestures that virtually grabs and releases a supported file into the upload area.

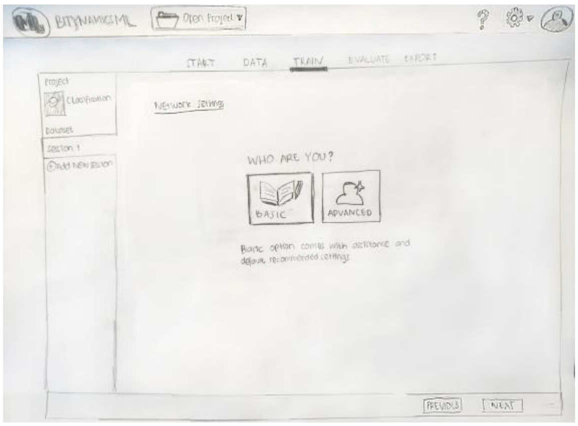



(a) Users initially start with selecting the dataset.

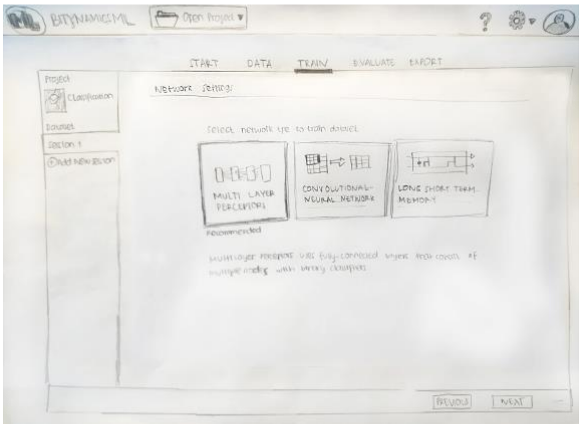
(b) Once a dataset is selected, users need to prepare it through a pre-processing.

*Figure 5: Data page acts as the dataset dashboard in the project*

When the platform manages to read and upload the file to our cloud service, the page shows summarizing statistics of the file. In order to proceed, the project should have no error datasets. If the uploaded file does not comply the dataset requirements for respective project type, the overview statistics return Number of Error Samples that users should consider to fix. As our platform has a functionality to simplify the data pre-processing, users are provided with an option to allow us taking care of them. Once the platform clean all errors in the uploaded dataset, it is ready to be trained in the next stage.




(a) Users can select between two modes based on their expertise levels.




(b) If an Advanced mode is selected, users can adjust various training parameters.

*Figure 6: Train page is hierarchically located under a Session and used to manipulate the training settings.*

A Session consists of single Train, Evaluate and Export processes. In this degree project, only Train reflects relevance to the scope of input interface. When users open the Train Page for the first time after selecting a dataset, the project automatically creates a new Session in which they can select either Basic or Advanced mode. To present the two different modes, the main area of the page shows the icons, names and explanations of both modes. The basic mode is a simple setting that allows users to run the training without adjusting any parameters as the platform handles all the required information based on the best practices. On the contrary, the Advanced mode allows users to use a GUI to switch and adjust between different settings for training models. The GUI-based mode aims to bridge between the error-tolerance and flexibility by maintaining simple interaction that eliminates possibility of entering non-executable command line and allows users to customize the parameters at the same time.

#### 4.2.2. Testing and analysis

Using the Wizard of Oz method, we asked the participants to interact with our paper prototype and deliver a machine-learning model. The paper prototype focused to deliver the idea the workflow and functionalities. The participants were asked to articulate their feelings, impressions or reasoning rather than just their final actions using a think-aloud protocol. We then obtained several feedbacks in Table 5 that we used as iterative improvements to build our interactive prototype.

| No. | Elements | Feedbacks |
|---|---|---|
| 1. | General | GUI-based interaction was easily understood.<br>Participants captured the intention of each page despite the doubts in some details. Generally, participants prefer the utilization of icons, layout and workflow that follows the convention in common web-based application. |
| | | Project-Session organization was intuitive, but sometimes malfunctioned.<br>All participants easily grasped that they could generate multiple Sessions in a Project. However, some participants did not realize the possibility to reuse same project and dataset settings to train different model by making another Session. |
| 2. | Project Navigation | Items in navigation bars gave clarity of current state.<br>Novice participant managed to learn the general state of ML process by recognizing the Stage Indicator and concluding each of the steps' names. All participants agreed that these provided reachability of all states by simply navigating within the Stage Indicator. |
| | | Multiple navigation bars led to confusion.<br>Participants struggled when they had to pick a navigation using the Stage Indicator or Sidebar. Although they practically serve different function, participants suggested to merge both functionality as some stages are actually wrapped under a Session that must be navigated using Sidebar. |
| 3. | Start | "Import from Other Platform" was out-of-specifications.<br>Our stakeholders suggested to temporarily remove the possibility to migrate from other platform as the team currently focuses on building an independent ML platform that assists users to build a model from scratch. |

| | | |
|---|---|---|
| 3. | Data | Data pre-processing lacked of customizations.<br>From the perspective of expert users, they were willing to have more flexibility. They expected to be able to adjust how the pre-processing can be done based on the project type and dataset itself. |
| 4. | Train | Basic and Advanced modes sufficed users' expectation but were not explained well.<br>All participants supported the idea of separating the approaches for different user target based on the expertise level. Our novice participant delivered the task by using the automation that Basic mode offered, while the experts were satisfied to have control over different parameters. However, some participants required more relevant explanations about the differences between the modes, especially regarding the actions they could do after a particular mode is selected. |

*Table 5: Participants' feedbacks in each of the elements presented in the paper prototype*

## 4.3. Design evaluation

### 4.3.1. Interactive prototype design

The design evaluation was conducted by testing the interaction between participants and an interactive prototype. While the sitemap remains the same, this part elaborates how the iterative improvements were made to each particular components from our previous design requirements.

#### 4.3.1.1. Project type selector

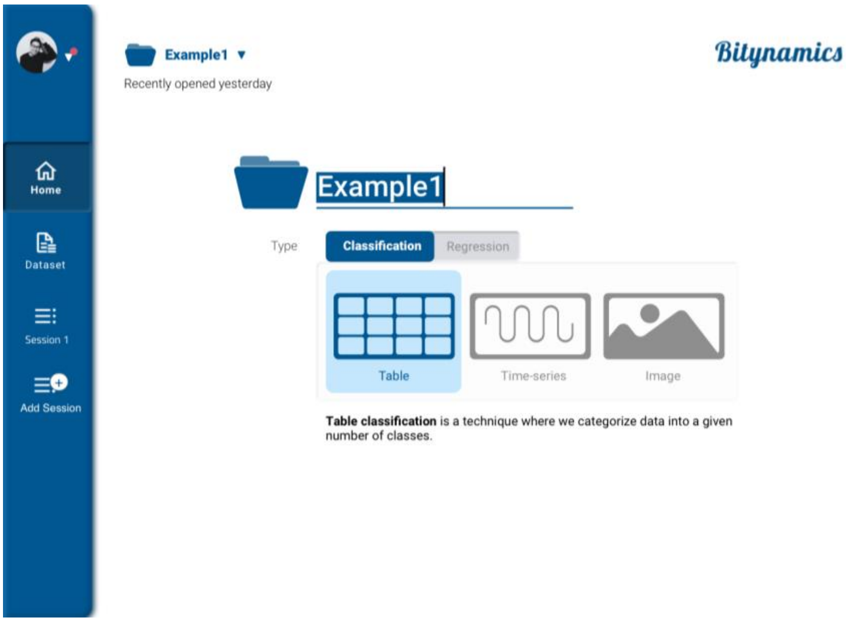



*Figure 7: Using the selector, users tell their expected output and input as a project type. Each of the available project types represent combination of output and input types.*

The project type selector is located at the Start page which serves as the home screen once users enter the ML platform console. The selector is located at the very early stage in order to include users' intent as a starting point. Establishing clear task goals is an important part especially for non-experts considering the process is largely user-driven. The pursuit of clear goal helps collaborative actions between platforms and users to determine the correct strategies as well as constraints in the process [5]. Additionally, users are often imprecise and inconsistent as ML training process is open-ended. It is also essential to understand goals and constraints since the interaction with an ML model is different from more conventional computer interactions where users generally can see the direct impact over their actions. [14] An application that is less responsive to user input violates the principles of direct manipulation and may causes frustration. Thus, providing the selector as soon as they begin the project can aid the focus, persistence and sense of control from users' perspective.

Using the selector, users start by telling the platform their expected output or expectedly-solved problems: classification or regression. Subsequently, the selector shows available choices of project type for the selected output type. Users are then able to pick based on which dataset or input type they currently have. Each project type determines the possible manipulations that users can do in the following stages and eliminates options of irrelevant decisions.

The use of icons to represent different project types aims to catch quick identifiability and memorable attention with its abundant visual forms. The icons combine basic image feature and text that contain obvious figures and texts to build meaningful understanding for users [16]. While the icons highlight and emphasise the differences among project types, the selector is also equipped with literal, concise definitions just below each of the icons. The definitions clarifies an elaborative explanations of the project type itself.

#### 4.3.1.2. Navigation sidebar

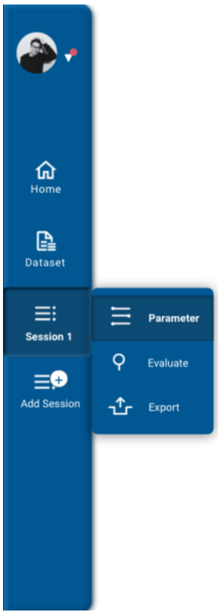



*Figure 8: Navigation sidebar helps users to easily switch into other states within the project.*

The platform initially includes the navigation functionality to provide reachability of all states and enable interactive inspections that improve the output quality. To eliminate confusion resulting from multiple navigation bars as suggested previously by participants' feedback in paper prototype testing, the State Indicator is now merged inside the Navigation Sidebar. This also follows the golden ratio for general web design regarding the width of navigation area that should not exceed 38% of the whole screen to maintain users' focus on the main task [17]. The elements of our new navigation sidebar consist of 1) User Settings; 2) Home to open the Start page; 3) Dataset to select and pre-process the dataset; 4) Session to expand single Session page; and 5) Add Session. To maintain the functionality that State Indicator used to serve, there are three sub-elements that can be accessed under the Session which each of them represents sequential state during a process: Parameters, Evaluate and Export.

The navigation sidebar is designed to be explicit and independent to the main screen area for enabling users' familiarity among different states of the process. As users easily navigate by clicking different button elements in the sidebar, they should also be able to maintain similar interactions and minimise refamiliarization cost. In HCI domain, consistency within the whole system generates higher convenience level by requiring users to spend less effort for relatively-small changes.

Another benefit of illustrating current state is the possibility of mutual learning process for novice users. Apart from the main navigation functionality, each element in the sidebar removes the vague separations of the states and replaces them with clear terminologies to call a certain partial task. As the users obtain better understanding on what they currently do, they build stronger engagement and focus to the main goal.

#### 4.3.1.3. Dataset pre-processors

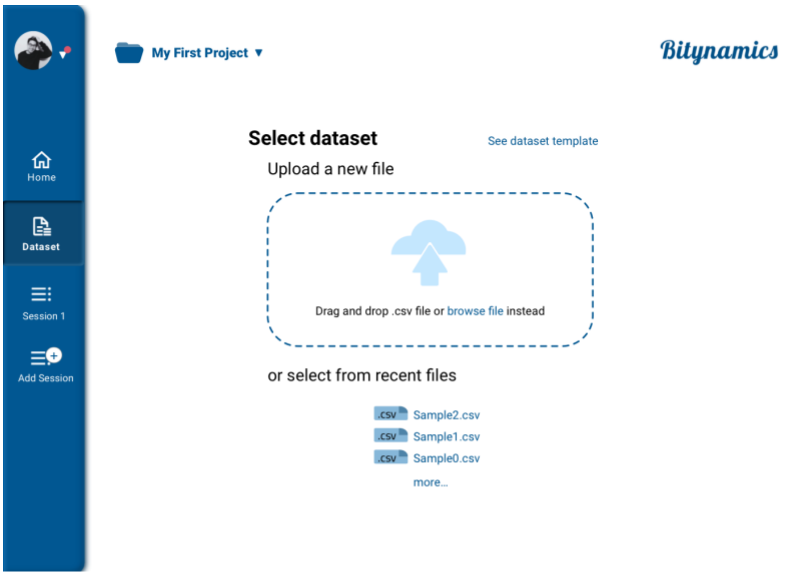



*Figure 9: The dataset selector enables users to upload new or reuse previous datasets.*

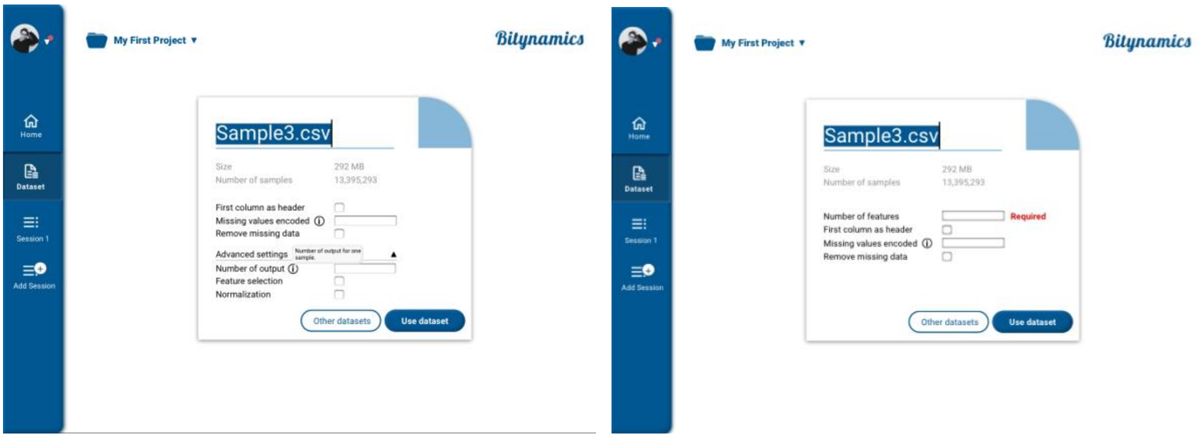
When users first time open the Dataset tab for a new project, the screen returns a dataset selector that consists of an upload area and a list of recent files. The upload area is designed as huge, obvious block with clear icon and instruction for guiding users' interaction with it. Otherwise, users are able to use recent datasets for the project by selecting from available options on the list just below the upload area. This straight-forward workflow requires minimum effort from novice users and avoids them from doing prone-to-error actions.

Once user manages to pick a dataset from either upload or select a file, the screen shows the Dataset pre-processor that ask users series of questions regarding how the file should be treated in order to fulfil training specifications. Expert participants of previous user testing argued that they expected chances to have certain degree of involvement to freely adjust the pre-processing itself. However, as this additional functionality might return a challenge for novice users who demands simplicity and automation, the pre-processor has to maintain clarity and include potential learning process for them. To bridge the gap between those expectations, we embodies several solutions in this state as seen in Table 6.

| No. | Solutions | Details |
|---|---|---|
| 1. | Using tooltip | Tooltip is a small pop-up box that appears when users hover on a certain element. While tooltips can actually contain various information regarding the element, in this pre-processor they particularly answer “what’s this?” question. The tooltip is used to help novices understand technical terminologies used in the pre-processor by describing it in a more generic language. If the content validity is high, a tooltip acts as a context-sensitive help that improves learnability of a system [25]. |
| 2. | Adding advanced settings | If a pre-processing parameter is optional in the sense that the dataset file can still be used for the further training process without particularly defining its value, it is located under the advanced settings. By default, the advanced settings is collapsed to preserve conciseness of the pre-processor. |
| 3. | Project-type based scenarios | To provide effective workflow, the pre-processor only includes relevant parameters for the selected output and input type. This eliminates potential irrelevant customization that might lead to prone-to-error actions. From users’ perspective, these multiple scenarios also consume less decision-making time by excluding consideration of not pertinent options. |

*Table 6: Some solutions incorporated by the Dataset pre-processor to bridge the expectation gaps between expert and novice users.*

In addition to this, some parameters show “Required” label next to their field if they are left empty without any meaningful context for empty values. In Figure 10, we can notice that available pre-processing options are different based on the dataset type. We based our project-type based scenarios by working closely with the backend team to decide which parameters are included or not as the results can be seen in the table below.




(a) The interface of pre-processor for tabular regression project.

(b) The interface of pre-processor for image classification project.

*Figure 10: Multiple scenarios are incorporated by showing different parameters in the Dataset pre-processor based on the selected project type.*

| No. | Parameters | Classification | | | Regression | |
|---|---|---|---|---|---|---|
| | | Tabular | Time-series | Image | Tabular | Time-series |
| 1. | First row as header | v | v | | v | v |
| 2. | Missing values encoded | v | v | v | v | v |
| 3. | Remove missing data | v | v | v | v | v |

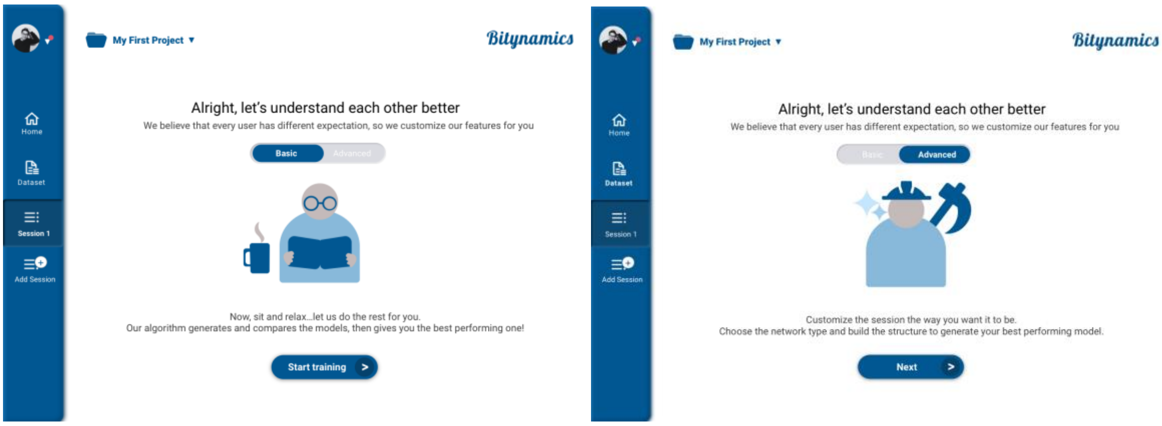
| | | | | | | |
|---|---|---|---|---|---|---|
| 4. | Number of features (per time period) | | v | | | v |
| 5. | Image dimension | | | v | | |
| | Advanced settings | | | | | |
| 6. | Number of multiple outputs | v | v | | v | v |
| 7. | Feature selection | v | | | v | |
| 8. | Normalization | | | | v | v |

*Table 7: List of different shown parameters in the Dataset pre-processor based on the selected project type.*

#### 4.3.1.4. Mode selectors

When opening a new Session page, users start by accessing the Parameter sub-page interface to choose a mode for building the model. A switch is located at the centre of main area to attract users' attention and select the mode in an interactive manner. Below the switch, a respective icon and a description text change in conjunction with the selected mode on the switch. These two self-explanatory elements serve to elaborate the definition and differences between each mode.

The presentation of the selector counts heavily on two icons that each represents users' further action once they select a respective mode. From the previous testing, we realized that the differences should be communicated in a way that simply builds correct users' impression regarding how a certain mode directly impacts their next actions.




(a) Descriptions of Basic mode (b) Descriptions of Advanced mode

*Figure 11: Two different modes presented in the selector.*

While the paper prototype had briefly explained the difference between the modes to the testing participants, Table 8 summarised how the modes are presented in the interactive prototype to meet the design requirements gained from the all previous steps.

| No. | Tasks | Basic mode | Advanced mode |
|---|---|---|---|
| 1. | Customize parameters for training algorithm | No. | Yes. |
| 2. | Run and compare multiple training algorithms | Yes, mandatory to allow the automation process. | Yes, optional for tuning parameters. |
| 3. | Build impressions to users | 1. Simplicity<br>2. Self-automation<br>3. Error-preventive | 1. Flexibility<br>2. Transparency<br>3. Sense of control |
| 4. | Mainly target particular user group | Novice users | Expert users |

*Table 8: Users' tasks comparison between Basic and Advanced modes.*

#### 4.3.1.5. Algorithm customizer

Algorithm customizer is only available for Advanced mode training to spare the span of advanced users' control by providing possibility to manipulate the algorithm that generates the expected model. Similar to the methods applied in data pre-processor, we used tooltips, collapsible advanced settings and project-type based scenarios to maintain simplicity and error-tolerant actions.

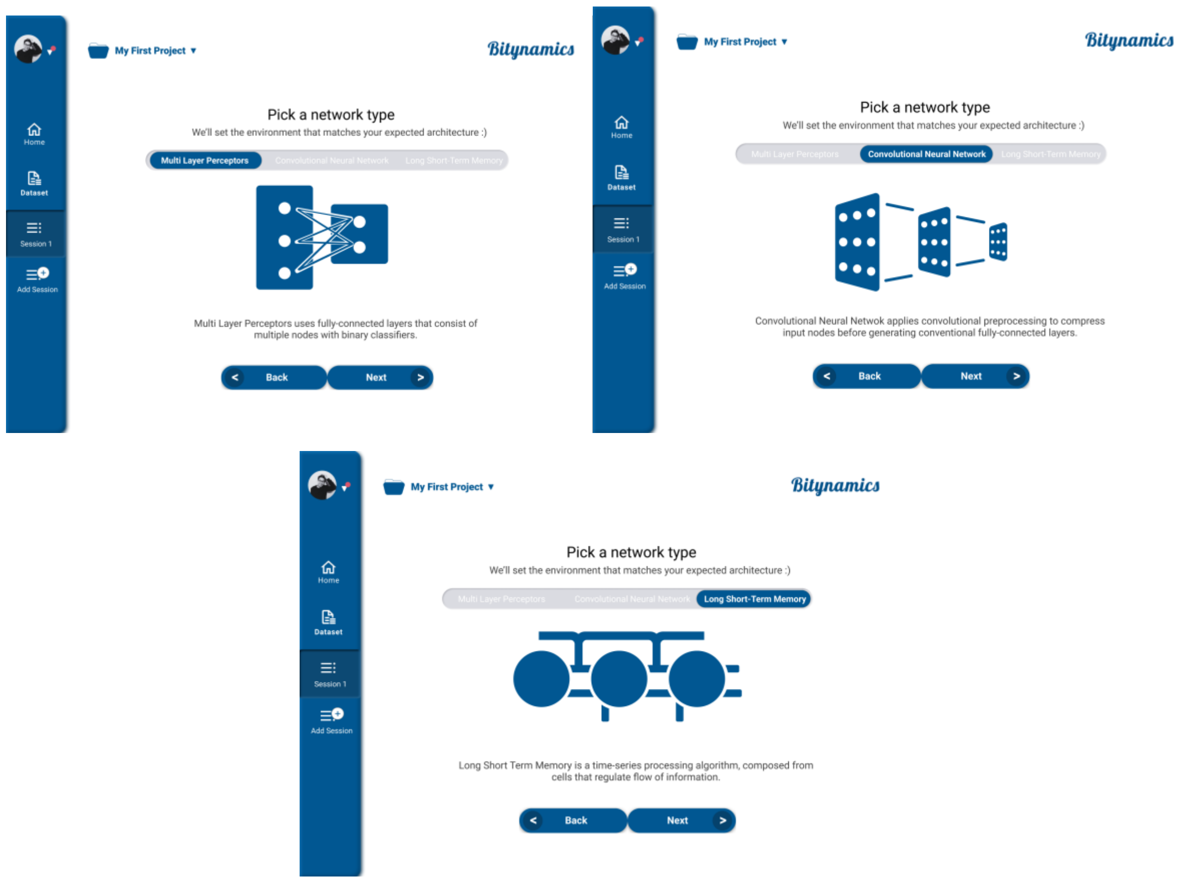



*Figure 12: The interfaces of Network type selector*

The functionality consists of three settings presented as sequences following the systematic workflow of a parameter tuning process as seen in Table 9.

| No. | Settings | Function |
|---|---|---|
| 1. | Network type selector | Generally, other customized parameters depend on the network type of the algorithm itself. This is the reason why users have to |

| | | |
|---|---|---|
| | | start by defining the network type before proceeding with further customization. Based on the Bitynamics' capabilities and requirements, there are three available types of network as shown in Figure 12: Multi-layer Perceptrons, Convolutional Neural Network and Long Short-term Memory.<br>The presentation of each option in the network type selector uses a corresponding layout to the mode selector. It uses a switch, interactive icons and description texts that change according to the active switch state to communicate the definition of each network type. The use of icons aim to quickly catch attention and refresh expert users' memory who are generally familiar with the terminologies. In case they have not figured a particular network type before, the textual description elaborates how it is practically utilized as an algorithm in the training process.<br>To eliminate irrelevant choices, the selector applies project-type based scenarios. This means that only executable network types that can train respective input dataset to achieve expected output are shown as options. |
| 2. | Network architecture settings | The second part of the customizer aims to set the structure of the network that runs as our training algorithm. After a network type is selected, the screen shows an interface as seen in Figure 13.<br>A network used in the algorithm is basically a set of layers that contain perceptrons or cells . The interface provides graphical representation of layers that construct the network as navigable rectangle-blocks located in the centre area of the screen. Inside each of the block, there are various attributes (parameters) to customize how a particular layer behaves during the training process. Only available attributes that can be executed for the selected network type are displayed inside the blocks. To switch between blocks, users can navigate using the left and right arrow buttons below the layer area or simply tap the desired block.<br>These GUI-based blocks follow the Gestalt's proximity principles |

| | | |
|---|---|---|
| | | by locating attributes that belong to the same layer close to each other [18]. As a result, users perceive them as a group of related elements. On the contrary, the relationship between different blocks also can be explained by Gestalt's continuation and similarity principles. Each block is clearly separated but serves the same function as a representation of an independent layer with continuous flow from one to next. As GUI is previously concluded in our design requirements, it tackles the complexity of command-line based platforms that are generally prone-to-error.<br>In addition to the layer blocks, the interface also has an Architecture panel on the right screen area. The panel provides a summary of the layers composing the network architecture. It also serves a navigation functionality by activating a layer if users tap on a certain layer name. |
| 3. | Session settings | The final part of customizer serves as a settings panel on how the session runs the network algorithm. There are four parameters controlled in this page: 1) Data Split determines the ratio of samples from the dataset that are used in Training and Testing process; 2) Epoch sets the number of repetition times for a learning algorithm to train the designated dataset while the model is improved for each time; 3) Optimizer applies different types of algorithm on top of the network type to increase the final models' performances; and 4) Batch Size defines the number of samples to work through before the algorithm updates the resulted model. |

*Table 9: The elements of algorithm customizer*

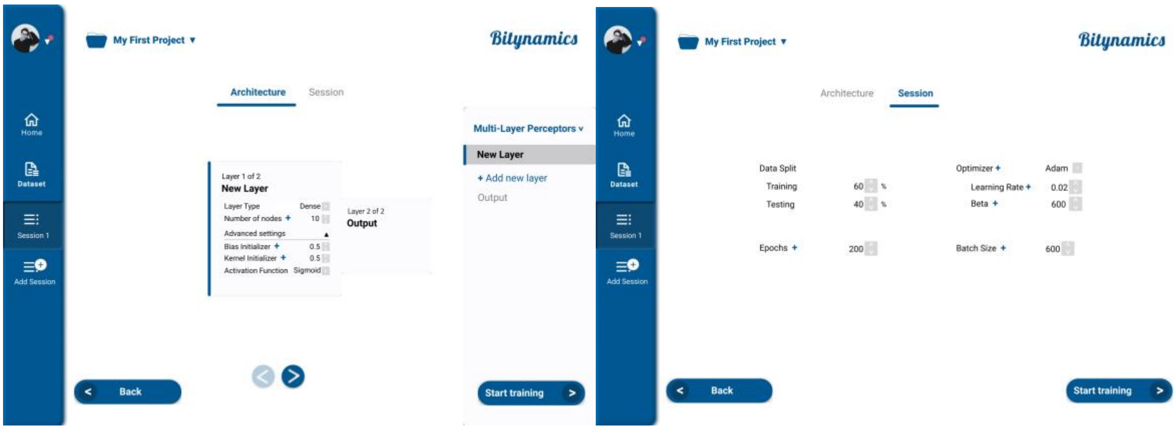



*Figure 13: The interfaces of Network architecture settings*

### 4.3.2. Testing and analysis

Before starting the evaluation, we identified participants' expertise level by knowing their previous experiences with ML as seen in Table 10. From this profile, three participants (Participant B, D and E) were assigned as expert users.

| No. | Participant | Built an ML model at least once | Used ML in professional context | Completed education in a ML-related field |
|---|---|---|---|---|
| 1. | Participant A | Yes | No | No |
| 2. | Participant B | Yes | No | Yes |
| 3. | Participant C | No | No | No |
| 4. | Participant D | Yes | Yes | Yes |
| 5. | Participant E | Yes | Yes | Yes |
| 6. | Participant F | Yes | No | Yes |

*Table 10: Obtained profiles of testing participants*

Subsequently, the result and analysis of the evaluation are presented and elaborated based on the types of feedback given by the participants.

#### 4.3.2.1. Visual design and workflow

All participants claimed the platform has an intuitive visual design that follows the general layout used in web-based dashboards. Participant A and D made a remarkable comment regarding the visual design that involved various "familiar icons with harmonized colour palette" that assisted them to perceive functionalities in switching tabs from sidebars, selecting project type, uploading dataset, deciding modes and choosing network type. Participant C added that the use of icons and description texts to represent different project types was helpful for her learning process as a novice user.

With no prior experiences in using any ML platforms, Participant C addressed the perceivable workflow that the platform has due to its minimalistic visual design. According to her, she earned minimal distraction and was able to remain focused on her task goal because the clean interface of each element properly highlights its functionality. Home page acts as what they expected to be "an entry point" by directly showing the properties of a recently-opened project and thus, this behaviour successfully builds an impression of a straight-forward and systematic workflow for all participants. Participant A, however, suggested that an improvement could be made to the Welcome page by presenting a list of all projects and allowing users to open a particular project from the list as what he usually found in the Welcome pages for project-based dashboards. The Session button that serves as a wrapper for Training and Testing states were easily captured by all participants as there was no hard time for them to figure where the Training could be done for Task 3. Both novice and expert users felt being addressed by the possibility to choose modes based on their expectation. All participant considered the mode selectors has been positioned correctly in the workflow as they had to decide before the training started.

When participants violated the default workflow by accessing Session tab before any datasets were set, the platform displayed an Error page in Figure 14.

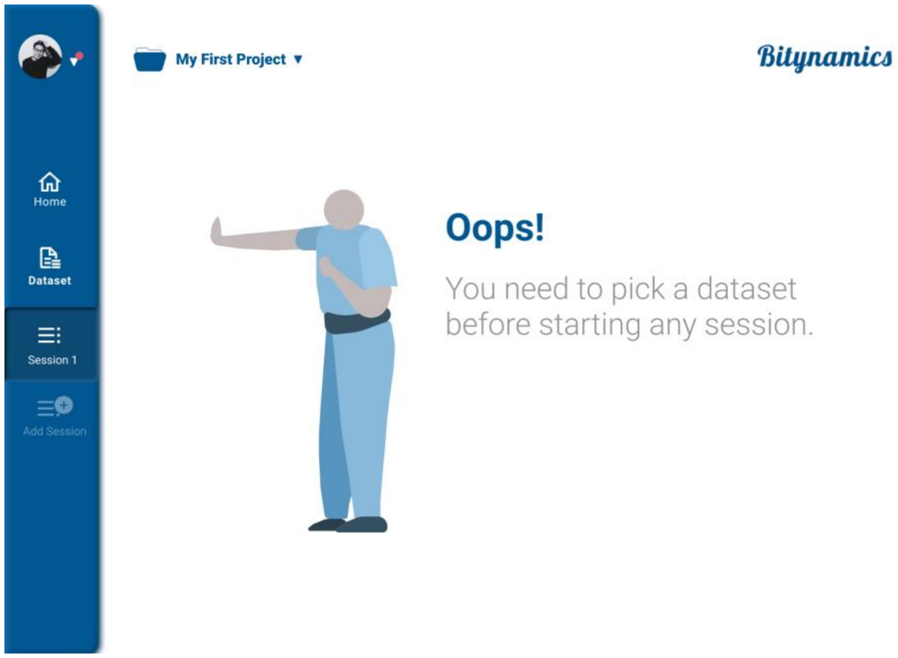



*Figure 14: The Error page appears if users skip a step and suggests the correct action.*

Inserting an Error page is a way to turn users' problem into a UX opportunity by providing enriching suggestions rather than just informing the error message. The error page is designed to avoid devastation that may lead to worse abandonment from users' perspective. Participant B, C, D, E and F experienced accessing this interface and understood what their next action was supposed to be from the description text. Participant D and E informed that the Error page design used in this platform was cognitively less frustrating since they did not become clueless on the correct workflow. Participant C even sensed a little humour from the icons that broke the tension with a surprising, yet engaging design.

Participant D and E recognized the layout has similarity with Google Cloud Platform since navigating into other projects can be done from the top area of every page and switching between states within a project uses a set of consistently-showing buttons. These familiarities helped them to deliver the task without any significant hassles. Both of them described the platform would be preferred than command-line based platforms if time was a valuable consideration due to its significantly-simplifying process. This argument was mainly based on the ability of our platform to include a sufficient level of customization.

The major improvement feedback from all participants was regarding lack of continuity that they sensed in the connection between tabs. They expected to end their actions in each page with "Save" and "Next" buttons to gain a feeling of closure while they were getting a hint to the subsequent steps at the same time.  Participant A and C mentioned that the buttons would have helped them to indicate their current actions as partial progress from the whole process and keep their focus to the goal better.

#### 4.3.2.2. Input techniques

The general remark from the expert participants was that our GUI-based platform successfully avoided them from entering irrelevant inputs as what they usually encountered in command line-based platforms. Apart from this error-preventive feature, Participant D and E also recognized a potential time reduction due to less debugging and troubleshooting.

Using the interface, there were three types of information that participants entered before the platform was able to train the model. The specific feedbacks given to the input technique for each information type can be seen in Table 11.

| No. | Information type | Feedbacks to respective input technique |
|---|---|---|
| 1. | Project type | The platform classifies the project types based on output (Classification or Regression) and input type (Tabular, Image or Time-Series). In Task 2 and 3, all participants did not experience any significant difficulty to declare the project type. They recognized and operated the project type selector easily by tapping on the switch and icons. Participant A mentioned the input technique generated good sense of learning as he could effortlessly checked on other project types and read the description texts below before deciding to proceed with one. |
| 2. | Dataset pre-processing | All participants expressed their familiarity to the form-based input that the pre-processor uses. In Task 2 and 3, Participant A, B, C and F used the tooltips and managed to understand the meaning of a certain parameter in the pre-processor.<br>In Task 2, one of the parameters entered for an image dataset had “Required” label that attracted all participants when they firstly arrived at the page. This was an expected behaviour as they could not left blank. |

| | | |
|---|---|---|
| | | Although all participants managed to deliver the tasks, Participant A suggested to avoid too "techy" terminologies in the tooltips. On the contrary, Participant D expected more elaboration for some parameters to avoid ambiguity since there were multiple approaches for them. This expectation gap can be bridged by defining two levels of definition inside some tooltips: 1) a short, concise explanatory sentence; and 2) collapsible information that contains the details on how the parameter determines the pre-processing.<br>In addition to this, Participant D suggested that the pre-processor can simulate and show preview on how certain parameters may affect the original dataset. |
| 3. | Algorithm customization | All three participants who delivered Task 3 as expert users considered the customizer pages had user-friendly input methods. The icons and texts in mode selector clearly distinguished the options by describing their respective next actions. Participant B mentioned if the description of an Advanced mode explicitly mentioned what could be customized in the following step, he could have raised his confidence level.<br>Participant D specifically addressed the architecture settings as "a visualization that triggers exploration". All participants easily operated the interface to add, edit or remove blocks that represent different layers in the network. |

*Table 11: Obtained feedbacks of each input technique*

#### 4.3.2.3. Navigation and interactivity

Despite lack of connectivity between states mentioned previously, all participants did not deal with any significant difficulty to navigate within the platform. The navigation is centralized using the sidebar and very intuitive. Participant C expressed her fond of sidebar interaction that enabled her to deliver Task 3 quickly. She figured out that switching to both previous and next states can easily done by tapping on different sidebar buttons.

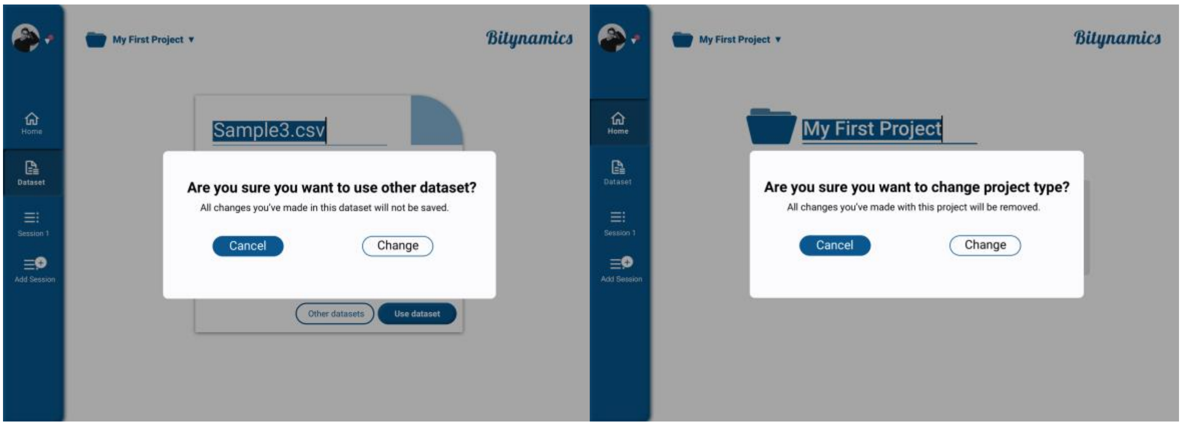



*Figure 15: The Reconfirmation page appears if users are about to execute risky actions that modify or cancel previous changes.*

Regarding the interactivity, all participants mentioned the advantage of having a confirmation boxes in the platform that simply reconfirmed their decisions if they were about to cancel or modify previously-made actions. All participants experienced the boxes in Task 3 when they had to change the project type and dataset or start the training process. The confirmation boxes are added to the platform to balance the reachability of all previous states by avoiding unintended manipulation to the progress they have made. The boxes are not used to confirm routine actions as overusing can result on users stop paying attention. Instead of "Yes" or "No" answers, the boxes provide response options that summarize what will happen as consequences.

Participant C believed that the boxes clearly informed her about the specific risk of executing such actions and prevented her from doing unnecessary errors. Participant D and E commented on the form of dialog boxes that popped up, interrupted and drew their attention before the platform proceeded. Participant B used the box to cancel the training process as he accidentally started the training before he finished the customizations. From this interaction, he came into conclusion that the confirmation boxes were important and raise the error-tolerance of the platform.

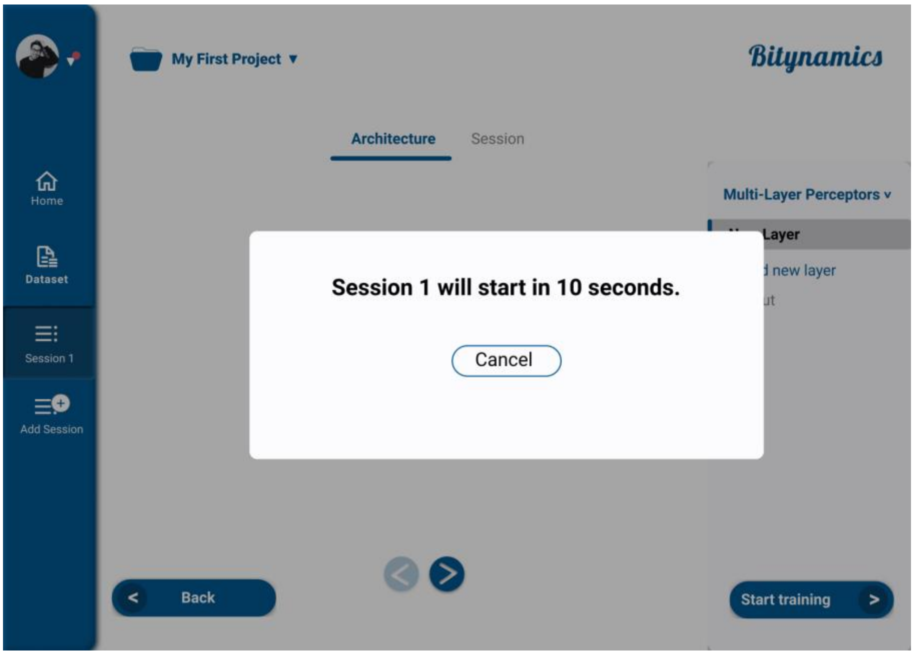



*Figure 16: A reconfirmation page asks users before the platform proceeds the ML training.*

# 5. Implications

To answer the research question, this chapter concluded the findings from all three methodologies that we conducted into a set of implications. These implications can act as general recommendations for designing an input interface for a machine learning platform.

## 5.1. Start with a clear definition of the specific ML goal

The high-level goal in ML generally, is to accurately describe the relationship between output and input values using a data-driven concept incorporated by a resulting model. When it comes to a specific ML project, the goal is narrowed down by including what kind of relationship wants to be explained or which type of values are used as output and input in the dataset.

User-centred approach emphasizes the importance of understanding the goal as an end-state that users want to reach from operating a certain product. Unless it is relevant to users' actual goal, a product that runs some functionalities with intuitive interactions does not guarantee any usefulness. In order to be defined clearly, a goal must refer as close as possible to the real world's expected result [19]. Following these premises, the goal should have an incentive effect that informs the user what can be accomplished by the platform as an end-state.

Our platform captures users' goal in the early stage when users start a new project by asking them to choose among five explicitly-provided choices of project type. Each of the project types represents a combinations of an output and an input type. This means that users should only focus to define 'what' rather than 'how' to achieve in the beginning.

Subsequently, determined goals also provide sufficient information to decide further steps based on the options that are toward users' satisfaction. In this platform, early determination of goals by selecting a specific project type enables a collaboration between platforms and users to determine the correct strategies and constraints for further steps. Additionally, as humans are generally imprecise and inconsistent, a clear goal helps them to organize their ideas by providing a journey orientation. Thus, in the further steps of the workflows, users can use the determined goal as a medium to keep them highly-focused while the platform supports the process with relevant states.

For expert users, the importance of goal determination is related to the fact that the training process is open-ended. The complete state is not binary and generally measured based on the accuracy. Instead of aiming for a perfect accuracy level that is barely achieved in reality, the platform serves the project type as a captured initial intent that users want to achieve with their datasets, whether they want to classify or regress the samples in their tabular, image or time-series information.

For non-experts, defining a goal is a chance to translate their initial expectation into more ML-related objective as a starting point. One of the main challenges in designing a universal platform that also works for novice users is their lacking of ML-related knowledges. Non-experts' unfamiliarity with the internal behaviours of ML constructs a mental model to aid their perception from learning-by-doing experiences. Thus, it is ethical to promote learning opportunities in our platform by including correct information in a user-friendly manner. While there are various methods to deliver the information, our platform uses icons and description texts that translate the goal without any "too techy" terminologies. The possibility to interactively switch between different choices of goals has also been proven to generate good sense of learnability and eliminate the boundaries of low expertise level.

Additionally, as revealed in the interview session, users like the sense of control. The opportunity to tell the platform regarding which goal to achieve gives user power to effortlessly manipulate the journey. As real world end-state is usually predetermined before users start to use the platform, even novice users should not find any difficulties to pass the goal information. Although it is just the beginning, the platform builds an impression that the users can perceive the changes according to their own expectations, control the process and obtain the results they seek.

## 5.2. Present states of ML with a straight-forward flow that promotes learning opportunity

A straight-forward flow is defined as serving a systematic sequence of tasks that only includes essentials to deliver users' goal. The importance of a straight-forward flow has been revealed from the initial interview when participants expressed their fond of simple, clear GUI used by AutoML. This turns the platform into content-centric that avoids an obstruction, a dark UX pattern that complicates the process than how it is supposed to be. While obstructions makes users hard to maintain focus, a straight-forward flow should be pursued as it arranges the process in a logical manner and reduces the noises. The flow enables less users' cognitive resources to conveniently perceive the required information and execute tasks [20].

The final design of this platform follows the logical systematic workflow for input process in a general machine-learning process.

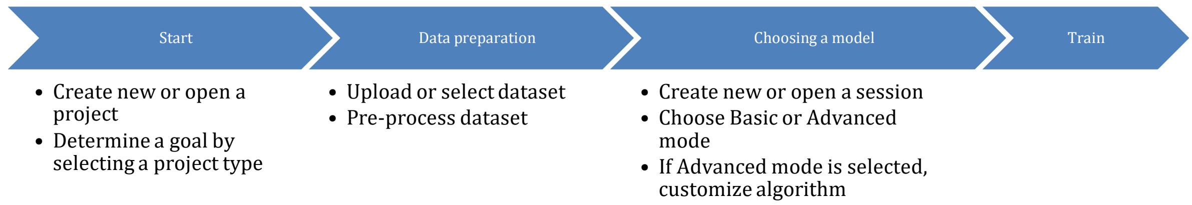



*Figure 17: Workflow used in the final design of input interface*

Furthermore, a straight-forward flow alone does not suffice. In a conventional ML platform, users need to do prior research and learn about the best practices before choosing a model to train. This problem forces an interruption of the whole ML process that leads to less enjoyable user experience. As a solution, this platform includes elements that support integrated learning. If the machine learning platform is able to act as one-stop service to both execute and learn the ML process, the interruptions are minimized and the platform flows smoothly to allow users maintain focus on their main tasks.

The platform enables users to have a learning-by-doing experience by providing non-disruptive guidance, definitions and other details alongside the interface. This way, users with various levels of expertise are able to make well-informed decisions while making progress in the workflow without prior research to different sources. In the input interface of our platform, there are two incorporated approaches, such as explicit instructions in each state and collapsible tooltips to explain specific ML-related terminologies.

Although there are various ways to preserve a straight-forward flow, it is important to put users in a circumstance that increases their concentration to the core tasks rather than complementary functionalities or ornaments. In the design evaluation testing, novice participants were revealed to easily perceive the workflow as our platform uses minimalistic visual design. Every element serves a purpose and with only a single focal point that highlights the functionality well. Minimalism was chosen to avoid distractions while emphasizing the simplicity of our platform at the same time. The design aims to tackle a stigma regarding ML as a complicated and overwhelming field to master [21].

Another necessary point is to show the connection flows between states. Users need to understand that the current state is a partial step from the whole ML process to achieve the main goal. It is also revealed from the design evaluation that users should be able to sense a progress towards the final intent in order to reach satisfaction. This requirement becomes more demanding when the process is complex and long. Thus, a continuous update of the progressing steps must be available to avoid frustration. Breaking down a long journey of ML process into sequential steps is an effective solution to prevent overwhelming tasks but users then need to be equipped with a progress tracker. Simply, the progress tracker gives users a clear idea of how far they have walked compared to the whole journey that they must accomplish [22].

## 5.3. Enable two-way transitions between all states

In this platform, two-way transitions were initially introduced to build an interface where users can reiterate the decisions made in the previous state. As opposed to proceeding to the next steps, users should also be able to go backward for many different reasons, such as bumping into unexpected paths, tuning parameters that lead to improvements or even just undoing unintended clicks. Based on the interview results, two-way transition is a suitable navigation approach towards a sense of control and possibility to explore that some participants proposed. The early remarkable findings for navigation system include the importance of clearly labelled return actions to avoid dead-end and increase error-tolerance.

Subsequently, the transitions for this platform are then boosted into reachability of all states using the sidebar menu. The improvement was made as an ML process is generally open-ended with continuous refinements and users should be convenient to modify all previous states. In UX context, an element is considered as reachable if it can be effortlessly utilized by all users irrespective of their current circumstances, abilities or contexts. The sidebar, in the platform, has an absolute position on the left area of all states. This consistency ensures users' ease to seamlessly navigate from and to every state.

Furthermore, the sidebar mainly has two functionalities: 1) as a state indicator that gives users clear illustration of the current state while allows reachable navigation to other states at the same time; and 2) to introduce the hierarchy between projects and sessions within the platform. This hierarchical structure aims to present the states in a more user-friendly order, make them easier to understand and allow reusability of states that are attached in the project-level. After choosing and pre-processing a dataset to work with, users can create multiple parallel sessions that still belongs to the same project with different training algorithms for each of them. As our platform is cloud-based, it removes the low limitation of capacity encountered by common local storages and effectively arrange different sessions to reuse pre-processed dataset in the same project.

As it was also decided from the early findings that our platform uses a GUI-based interaction, the transitions can even be presented in an intuitive and interactive manner. All participants in the design evaluation testing mentioned that they had no problem understanding how to navigate between states. Users can easily switch to both previous and next states by tapping on different sidebar buttons. The interaction takes another major advantage of using GUI by having the ability to provide users with immediate, visual feedback about the effect of each action.

From the design evaluation, the participants were revealed to seek for security alongside the interactivity. The platform fulfils this premise by adding a confirmation box to avoid unexpected changes and inform the risk every time an action is about to modify previously-made changes or be cancelled. The confirmation boxes are a safety net that balances the reachability of all previous states by avoiding unintended manipulation. It is beneficial to note that such confirmation boxes should not be used to confirm routine actions as overusing can result on users stop paying attention. Instead of “Yes” or “No” answers, the boxes provide response options that summarize what will happen as consequences. The confirmation boxes, according to our participants, is also a good addition to raise the error-tolerance of the platform.

## 5.4. Accommodate different users’ goals with multiple scenarios

In UX domain, a scenario is an expected story of users’ goal accomplishment via a product. The goal-based scenarios use what users want to achieve as the factor that determines them to take certain paths [23]. If an ML platform decides to support multiple scenarios, this means that there are more than one story that accommodates users with wide variety of goals. The story should be adjusted to suit the users with a sufficient adaptability level so that they can utilize the platform towards their respective end state. If users’ effort and attention to operate the platform are considered as resources with limited supply, it is reasonable to argue that all spent resources must be in pursuit of the final goal [5]. Thus, users with different goals should only be exposed to the relevant steps in their scenario.

In addition to spending less resources, multiple scenarios helps users to avoid unexpected errors by eliminating irrelevant options of decision. Actions that are non-executable or do not support users’ goal should not become choices, especially when there have been overwhelming number of them. While many ML platforms pick different adaptive elements to pursue multiple scenarios, the adaptivity should not be overused and too-smart. If the adaptivity decides too much for users, they might suffer from frustration as they lack of control to do what they actually intend. The case when an adaptivity becomes too smart is, for instance, when some choices are removed only because they are too risky without considering some reasonable circumstances when users might pick them. Thus, it is also important to make the adaptivity unobtrusive to users' intent. The platform must also communicate that it has an adaptive system and becomes transparent regarding the reasons why certain choices are blocked.

In the platform, multiple scenarios are used in some states. First, the dataset pre-processor has different scenario for each of the project types by only including relevant parameters for the selected project type. As this eliminates potential irrelevant users' input that leads to an error, it also promotes an efficient workflow and consumes less decision-making time by excluding consideration of not pertinent options. The platform also benefits from its GUI-based interaction as it becomes possible to make an intuitive visual representation of the scenario that users can easily perceive and operate. Using the common interface of a form, the dataset pre-processor only displays labels and fields that users can fill according to the project type. The fields for irrelevant parameters remain hidden and users have no access to see or modify them. Second, if the platform aims to target users from wide variety of expertise level, it might be necessary to add adaptivity based on the level itself. From the interview, we revealed the fact that there are different expectations between novice and expert users accommodated by the platform in the final design with two available training modes: Basic and Advanced mode. To achieve their respective goals, novices can use the Basic mode, a simple setting that allows them to obtain a well-performing model without adjusting any parameters as they expects automation and simplicity. On the contrary, the Advanced mode accommodates advanced users' urge to build a flexible and transparent model by allowing them to customize the algorithm before the training starts. Although the selection of modes is not directly related to the goal itself, multiple scenarios in this context helps the platform to ensure its usability while provide an enjoyable experience that meets different users' expectation at the same time.

## 5.5. Provide expert users with more control to customize the models

From the initial interview, the expert participants was revealed to have a strong preference towards a platform that provides flexibility to train the model. Using the command line-based platforms currently available in the market, they were willing to gain full-control of customization at the cost of complexity and lack of assistances. Based on this finding, we come into conclusion that expert users must have a broader span of control than novices. This

platform fills the gap by providing an intuitive GUI-based platform that still maintain flexibility in defining the output models.
Generally, the sense of control is indicated by how our actions have the power to impact and change a given circumstance. When expert users with sufficient ML skills get exposed to a high level of control, they gain confidences from their ability to exploit their given power by executing rational actions. Users are less likely to frustrate as they are authorized to make their expected changes. Subsequently, users will also build stronger engagement to the platform considering its reliability to fulfil their desire.
This platform builds a deeper sense of control for expert users by providing an algorithm customizer. The functionality consists of a network type selector, network architecture settings and session settings that are presented as a sequence of three different panels following the systematic workflow of parameter tuning. To reduce potential errors, the customizer also applies project-type based scenarios which mean that only executable actions are displayed as users' choices.
Although expert users generally have prior encounters with other ML platforms, it is still important to maintain the clarity of contents and workflows. This platform avoids complexity by utilizing a familiar web-based form layout, icons, description texts and GUI-based visualization. The use of icons aim to quickly catch attention and refresh expert users' memory who are familiar with the terminologies. In case the information is new, the textual description elaborates how it is practically utilized in the customization process. Additionally, the navigable GUI-based visualization forms a clear idea about the content of users' algorithm network while also adds interactivity that boosts users' sense of control.
Furthermore, fulfilling users' need for flexibility in an ML platform is strongly related with their tendency to explore the algorithm before deciding the best performing one. Expert users anticipate a series of trial-and-errors to iterate and improve their final output model. The platform make exploration possible with multiple sessions under a project. After expert users generate a model from a training session, they can create new session and retrain the project dataset with different customized algorithm for multiple times before picking the one that meets their expectation best.
The sense of control, consecutively, helps to build transparency in the platform although it does not explain the entire process. As suggested by some expert participants during the initial interview, a good platform should be open regarding how they train to generate the resulted model. When users can choose their "own path" to reach the goal, they believe that the final result totally counts on their decision although there can be some parts of the process that

remain unexplained. The platform benefits from this illusion to allow users perceive sense of transparency without having to provide an end-to-end elaboration of the training process.

As a final point, control can manifest in various shapes and implementation methods of the platform as long as it maintains users' safety. Depending on the safety standard, the platform should not be able to provide users with threatening options. Bigger span of control exposes users to more risks that can prevent them from reaching the goals. Some fatal unexpected modifications can result on worse impact than bad performing models if the algorithm becomes not compatible to train the dataset. The frustration then can damage the user experience and reliability of the platform itself. Thus, it is very fundamental to pay attention at which options of customization should be available considering the right balance between sense of control and their potential risks.

# 6. Conclusion

## 6.1. Conclusion

The desired objective of this degree project is to generate a user-centred design of input interface for an ML platform. The research question defined to be answered is *How should the input interface of a machine platform be designed to deliver on common users' expectations and tackles their problems according to interviews, prototyping and design evaluation method?*

From the interview results combined with literature study and benchmarking, we concluded initial design requirements that summarizes users' expectations and problem: 1) provide a cloud-based data storage; 2) declare explicit instructions; 3) use GUI; 4) include transparency; 5) promote straight-forward flow; 6) provide interactivity; 7) enable flexibility to customize the model; 8) avoid prone-to-error actions; 9) recommend actions for improving models; and 10) illustrate current state of the process.

Subsequently, paper and interactive prototypes were used to iteratively check, test and improve the interface design. During the evaluation, participants used the prototypes to perform a set of input tasks in an ML process: 1) create new project; 2) upload dataset; and 3) choosing models. As a final deliverable that answers to the research question, the findings from all three methodologies were concluded into a set of general implications that guide the design process of input interface for an ML platform.

First, *start with clear definition of the specific ML goal.* Early establishment of the right goal in the platform helps user to organize their ideas by providing a journey orientation. Subsequently, determined goals also provide the platform with sufficient information to decide further steps based on the options that are toward users' satisfaction. The high-level goal in ML generally, is to accurately describe the relationship between output and input values using a data-driven concept incorporated by a resulting model. Following the previous premise, the goal should have an incentive effect that informs the user what the platform can accomplish as an end-state. This means that users should only focus to define 'what' rather than 'how' to achieve in the beginning.

Second, *present states of ML with a straight-forward flow that promotes learning opportunity.* A straight-forward flow reduces noises and requires less cognitive resources to process as only prioritized information is perceived. When it comes to designing an ML platform, a straight-forward flow alone does not suffice to avoid interruptions since conventional ML process forces users to do prior research and learn about the best practices. Indeed, excessive interruption inevitably leads to bad user experience and thus, a machine learning platform can tackle with including elements that support integrated learning experience. If the machine learning platform is able to act as one-stop service to both execute and learn the ML process, the interruptions are minimized and the platform flows smoothly to allow users maintain focus on their main tasks.

Third, *enable two-way transitions between all states.* As opposed to proceeding to the next steps, users should also be able to go backward for many different reasons, such as reiterating the decisions made in the previous state, bumping into unexpected paths or even just undoing unintended clicks. Two-way transition is a suitable navigation approach towards a sense of control and possibility to explore that some participants proposed. It also helps to avoid dead-end and instead increase error-tolerance. Specifically for this platform, the transitions are then boosted into reachability of all states as an ML process is generally open-ended with continuous refinements and users should be convenient to modify all previous states.

Fourth, *accommodate different users' goals with multiple scenarios.* The scenario should be adjusted to suit the users with a certain adaptivity level so that they can utilize the platform towards their respective end state. If users' effort and attention to operate the platform are considered as resources with limited supply, it is reasonable to argue that all spent resources must be in pursuit of the final goal. Thus, users with different goals should only be exposed to the relevant steps in their scenario.

Fifth, *provide expert users with more control to customize the model.* When expert users with sufficient ML skills get exposed to a high level of control, they gain confidences from their ability to exploit their given power by executing rational actions. Users are less likely to frustrate as they are authorized to make their expected changes. Subsequently, users will also build stronger engagement to the platform considering its reliability to fulfil their desire. This platform builds a deeper sense of control for expert users by providing an algorithm customizer. The functionality consists of a network type selector, network architecture settings and session settings that are presented as a sequence of three different panels following the systematic workflow of parameter tuning.

## 6.2. Sustainability

The ML platforms provide centralized cloud-based computing resources that can be shared by multiple users. This possibility of sharing inventories reduces the average consumption of resource in each individual ML process and less overall required space. From the environmental perspective, it produces fewer pollutants, emission and carbon footprints generated for the manufacturing process of computing resources. In a long term, the use of ML platform is also expected to change the consumer behaviours from ownership to demand-fulfilment in other related assets, such as physical offices or electricity equipment. If more behaviours have shifted towards this direction, an end-to-end sustainable ML communities can be achieved to support Sustainable Development Goal 11[6].

While the involvement of ML in deciding reliable data-driven actions has been forecasted to become an enabler of all SDGs by 2030, the rise of ML platforms expands the affordability of ML itself to broader target users [27]. Moreover, SDG 4[7] states that education is a key to escape poverty and should be affordable to all potential learners. In the design implications section, this research concluded that including a learning opportunity in an ML platform is essential. Apart from delivering its main function in ML process, the interface exposes relevant knowledges to users in a simple and practical way.

## 6.3. Method Critique

As I am a novice in ML myself, the main challenge of this research was to ensure that the final design has accommodated the expert users' problems and expectations. Thus, prior to the researches, several literature studies were conducted to understand relevant terminologies and general workflow of ML processes.

In the early stage, the interview method significantly relied on the representativeness of expert participants and the validity of their opinions. To ensure the design process would include suitable considerations, it was important to capture the right problems and expectations. Furthermore, the paper prototypes were iteratively built and tested to the CEO of Bitynamics as he is an expert user who has an interest to improve the platform.

## 6.4. Future work

This degree project has mainly contributed to the general input interface design principles of an ML platform. Each of the final design recommendations has been revealed to may have multiple approaches. The most possible next step is to define more specific requirements that

[6] https://www.un.org/sustainabledevelopment/cities/
[7] https://www.un.org/sustainabledevelopment/education/

improve the design guidelines' elements respectively. While the methodology can remain the same, different prototype that focus on particular parts of the platform should be developed and tested to enhance the recommendations.

Subsequently, in the design evaluation, it was revealed that a participant expected the visualization of the dataset before proceeding to the pre-processor. As visualization is another broad topic with many design considerations, it is currently discussed in the design process of output interface by another degree project. However, a future research work regarding the design of an input interface might include a partial involvement of visualization that is relevant to dataset alone.

Furthermore, in this ML platform, the researches towards both input and output interface were conducted separately. In the initial brainstorming part, this degree project was conducted closely with another project to design the output interface. However, as the respective requirements became more overwhelming in each parts, we then realized that an integrator should be in charge in early and final part of both projects. Thus, there should be another research that takes responsibility as an integrator and delivers relevant guidelines to this topic. This matters as if the integration is done poorly, there might be conflict of interests between both interfaces that suffers the users. Some input can become useless while some output can have no sufficient data to process.

Finally, as the input interface resulted from this research is supposed to be integrated with the back-end and output interface of the platform from two other degree projects, there should be a further evaluation of the integration result itself. Before implementing the conclusions of each research, a user experience evaluation that involves the design of an integrated platform should be conducted. This is necessary since users are dynamic and might behave differently with a platform that provides a whole journey of ML process.

## References


[1] Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10), 78-87.

[2] Hurwitz, J., & Kirsch, D. (2018). Machine learning for dummies. *IBM Limited Edition*, *75*.

[3] G, Y. (2017, September 7). The 7 Steps of Machine Learning (in Google Cloud Platform). Retrieved March 1, 2020, from https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e.

[4] Baylor, D., Breck, E., Cheng, H. T., Fiedel, N., Foo, C. Y., Haque, Z., ... & Koo, C. Y. (2017, August). Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1387-1395).

[5] Dudley, J. J., & Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *8*(2), 1-37.

[6] Ribeiro, M., Grolinger, K., & Capretz, M. A. (2015, December). Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 896-902). IEEE.

[7] Pal, S. K., & Mitra, S. (1992). Multilayer perceptron, fuzzy sets, classification.

[8] Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

[9] Saha, S. (2018, December 17). A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. Retrieved March 1, 2020, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Image classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[12] Perkel, J. M. (2018, October 30). Why Jupyter is data scientists' computational notebook of choice. Retrieved April 15, 2020, from https://www.nature.com/articles/d41586-018-07196-1

[13] Michael Shilman, Desney S. Tan, and Patrice Simard. 2006. CueTIP: A Mixed-initiative Interface for Correcting Handwriting Errors. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06).* ACM, New York, NY, USA, 323–332. https://doi.org/10.1145/1166253.1166304.

[14] Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. 2015a. iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction. Technical Report. MIT Computer Science and Artificial Intelligence Laboratory.

[15] Sarkar, A. (2015, July). Confidence, command, complexity: metamodels for structured interaction with machine intelligence. In *PPIG* (p. 3).

[16] Yan, R. (2011). Icon design study in computer interface. Procedia Engineering, 15, 3134-3138.

[17] Beaird, J. (2007). The principles of beautiful web design. SitePoint Pty. Limited.

[18] Han, S., Humphreys, G. W., & Chen, L. (1999). Uniform connectedness and classical Gestalt principles of perceptual grouping. *Perception & psychophysics*, 61(4), 661-674.

[19] User Goals. (n.d.). Retrieved from https://usability.yale.edu/understanding-your-user/user-goals

[20] Gray, C. M., Kou, Y., Battles, B., Hoggatt, J., & Toombs, A. L. (2018, April). The dark (patterns) side of UX design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-14).

[21] Mullins, C. (2015, July). Responsive, mobile app, mobile first: untangling the UX design web in practical experience. In *Proceedings of the 33rd Annual International Conference on the Design of Communication* (pp. 1-6).

[22] Babich, N. (2020, May 14). Progress Trackers in UX Design. Retrieved from https://uxplanet.org/progress-trackers-in-ux-design-4319cef1c600

[23] Benyon, D. (2019). *Designing User Experience: a guide to HCI, UX and interaction design*. Pearson UK.

[24] Browne, J. T. (2019, May). Wizard of Oz Prototyping for Machine Learning Experiences. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-6).

[25] Isaksen, H., Iversen, M., Kaasbøll, J., & Kanjo, C. (2017, July). Methods for Evaluation of Tooltips. In *International Conference on Human-Computer Interaction* (pp. 297-312). Springer, Cham.

[26] Leech, B. L. (2002). Asking questions: Techniques for semistructured interviews. *PS: Political Science & Politics*, *35*(4), 665-668.

[27] Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., ... & Nerini, F. F. (2020). The role of artificial intelligence in achieving the Sustainable Development Goals. *Nature Communications*, *11*(1), 1-10.

[28] Sauer, J., Seibel, K., & Rüttinger, B. (2010). The influence of user expertise and prototype fidelity in usability tests. *Applied ergonomics*, *41*(1), 130-140.

# Appendix

This section contains the notes taken during the user testing for design evaluation method that involved six participants.

| | |
|---|---|
| **USER TESTING** | |
| **Participants** | A |
| **Time** | Fri, 17/4 10.00 |
| **Expertise** | Novice |

**General feedback/First Impression**

1. He figured out the console easily
2. He understood the fields to select project name and project type
3. He thought the project name was a project navigation
4. He managed to switch project from the top menu

**Task 1 - Create New Machine Learning Project**

1. At first, he tried to click on the field to change project name
2. He thought about it because the field was highlighted. Might have helped if the title was not highlighted.
3. He tried to change project type but still couldn't manage to find what he wanted. He was expecting Create or Add New Project.
4. He figured out Add New Project inside the top menu.
5. He was confused what to do next because there was no Next button to continue.
6. Home is an entry point so he was not expecting that he had to switch to other window to Switch Project or Add New Project.

**Task 2 - Switch to Image Classification and Upload a Dataset**

1. He didn't check the project dataset template.
2. He focused on the field with Required text. Filling it was the first thing he did.
3. He was confused what the Dimension meant. He suggested it was (number of) pixels.
4. He hovered on the Information icon next to missing values encoded and read the description well.
5. He thought he did not understand that for a while but then he read the explanation twice. He then understood the definition of missing values encoded but he was not sure 'Encoded' is the correct word.
6. He was confused what the normalization could do the dataset.
7. He suggested to use 'Replace datasets' instead of 'Other dataset'
8. He managed to deliver the task.
9. Again, he was confused what to do next from the last step.

**Task 3 - Change the project into a table classification and start generating the model.**

1. He clicked on Home > Change dataset > Yes > Table Classification
2. He said the confirmation box was already good since it clearly explained the risk of doing such an action.

3. He clicked on Session and got the disclaimer to reupload the dataset.
4. He was confused since he had to reupload a dataset. But, he went to the Dataset tab again to do the action.
5. He didn't figure out that different project type required different dataset.
5. Again, he didn't check on the dataset template. (Ask why a participant did or did not do that)
6. He went through all steps in Dataset tab very fast. (Perhaps because he thought it was just a bug to repeat this step! Go ask this on your next participant.)
7. He figured the next step was supposed to go to Session tab again because he checked there previously.
8. He read the explanation of Basic function easily but he was confused since nothing happened after he clicked on Start Training

**Overall**
1. He liked that visual appearances had been included as design consideration.
2. He felt there were missing links between stages. Such instructions or 'Next' button could help user to understand the Next Stage.
3. He would like to see the Task Overview or 'Active Stage' of the project.
4. He suggested to follow the general flow since Home usually shows Overview and All Projects instead of just becoming 'Home' for one project.
5. He liked the sense of mutual learning that the explanation of each terminology had. He suggested to avoid too techy words.
6. He proposed to use 'Next' button because it could help users to focus on the entire goal.
7. For the thesis purpose, the definition of User Group would be very important.

---

| | |
|---|---|
| **USER TESTING** | |
| **Participants** | B |
| **Time** | Sat 18/4 09.00 |
| **Expertise** | Expert |

**General feedback/First Impression**
1. He quickly figured how to switch between projects from the top bar.
2. He figured out how to change project type although at first he thought it was for setting the dataset
3. He understood the explanation of each project type because he is familiar with the terminologies.
4. He realized the function of sidebar to switch between tabs (he went to the Dataset and Session tabs)
5. He found the 'Cannot Access Session' and followed what the page suggested (to set the dataset in advance) by going to Dataset tabs.

**Task 1 - Create New Machine Learning Project**
1. He figured out easily how to create new project and directly checked on the top bar.

2. He easily found the Add New Project button.
3. He was not expecting to click on the radio button next to the new project name. He thought it should have been the selected project.
4.  He then figured out that the project should be selected.
5. He suggested to automatically select (switch the active radio button) to the new project after user clicked on 'Add New Project'

**Task 2 - Switch to Image Classification and Upload a Dataset**
1. He expected to have a 'Save' button from the 'Home' dataset to make sure that his selection of project type was saved.
1. He easily figured out that he should switch to Dataset tabs.
2. He didn’t check on the dataset template because he didn't recognize it as a button.
3. He was expecting if the dataset he uploaded was unreadable because it didn't follow the template, he would get an error message.
4. He then could figure out that such things like a template existed from the error message.
5. He started by setting the required field.
6. He was familiar with all terminologies during the steps.

**Task 3 - Change the project into a table classification and start a training with MLP**
1. He switched to Home tabs and clicked on 'Change Dataset'
2. He liked that there was a confirmation to the action.
3. He then clicked the Dataset tab because he understood that he required different dataset (and all his previous changes were removed)
4. He was confused for some time because he was expecting to click Add Session instead of Session 1.
5. He understood the different between Basic and Advanced options after reading the definition.
6. But he was confused because he could not see anything related to MLP.
7. He read that Advanced options would allow him to customize the network type, but it didn't mention what the available types were.
8. At the first attempt, he tried to do the basic function because he couldn't find 'MLP' anywhere on the screens.
9. He easily figured out that it was not what he was supposed to do and he clicked Cancel on the Countdown window.
10. He tried the advanced option and found MLP.
11. He understood the function of 'Add New Layer', 'X' to delete a layer, and 'Left/Right Arrow' to switch between layers.
12. He didn't understand what MLP was but he enjoyed the functionality that enabled him to build the layer structure.

**Overall**
1. There was no significant difficulty for him to understand how the platform worked.
2. He suggested to have a 'Next' button in the end of each stage because sometimes he got confused on what to do next.

3. The 'Console' button from the Bitynamics homepage was a proper and logical entrance that he easily recognized.
4. When he first arrived in the Session 1 tab, he were only looking for 'MLP' as a keyword.
5. He only read the explanation of different options (basic vs advanced) because he didn't manage to find MLP.

---

**USER TESTING**

| | |
|---|---|
| **Participants** | C |
| **Time** | Sat 18/4 09.00 |
| **Expertise** | Novice |

**General feedback/First Impression**

1. She figured that the home page was supposed to be the control panel for projects.
2. She figured out how to switch and make a new project from the top bar.
3. She understood that she could change the type of machine learning by clicking on the picture.
4. She liked that there were explanations to each project type.
5. As she was not familiar with any machine learning type, she didn't fully understand the function but she knew which dataset type to put (from the picture).
6. She figured out that Dataset tab was supposed to upload and set the dataset.
7. The drag and drop field was super clear for her and she liked that she could get the statistics of the datasets.

**Task 1 - Create New Machine Learning Project**

1. She initially tried to click on the project name.
2. Because the project name was highlighted all the time, she thought that the project was new.
3. She was expecting to have a 'Save' button because she was going to switch to the Dataset tab for the next step.
4. She didn't manage to complete the task (because she thought she already made a new project).
5. She felt she only delivered the task partly because she could rename the project but she did not manage to save the changes.

**Task 2 - Switch to Image Classification and Upload a Dataset**

1. Again, she felt switching tabs resulted on losing all her work in the previous tabs.
2. She recognized the image classification from the icon.
3. She was confused because after she uploaded, she saw an option to 'Use other dataset'
4. She expected to have a 'Next' button instead of switching the tabs manually

**Task 3 - Change the project into a table classification and start a training with MLP**

1. She figured out directly that Project Type options were available in Home tabs.
2. She thought Change Type button was very obvious and she liked that there was confirmation box.

3. She still thought she had to keep reminding herself about switching tabs to execute different steps.
4. She was not sure if her change was saved because she had to switch tabs.
5. She argued that switching tabs resulted on losing continuity between steps.
6. She was confused why she had to reupload the dataset.
7. She didn't recognize the Dataset template as a button.
8. She proposed to add confirmation or warning (to follow the dataset template) when a dataset was uploaded.

**Overall**
1. She felt that the menu interaction generated 'partial' steps that lacked of 'holistic' sense.
2. She thought the interaction was simple, very intuitive and easy to understand.
3. She suggested to use progress interaction instead of menu interaction.
4. She thought having icon was very helpful to help her understand the project type and the session option.
5. She easily recognized the upload area (also because of the clear icon).
6. She didn't read the explanation about the session options, but she only skimmed through the title ('Basic' vs 'Advanced').
7. She liked that the design implemented an attractive 'Start training' button.

---

**USER TESTING**

| | |
|---|---|
| **Participants** | D |
| **Time** | Sun 19/4 11.30 |
| **Expertise** | Expert |

**General feedback/First Impression**
1. He read the definition of project type.
2. He understood how to set classification or regression.
3. He was not sure if 'Table' is the correct terminology. He suggested to use 'Tabular' instead.
4. He checked on Upload dataset tab.

**Task 1 - Create New Machine Learning Project**
1. He tried to rename the file because he thought it was a clickable button to manage projects.
2. He thought it was a button because it had highlighted text.
3. He easily figured out that the project panel was at the top bar.
4. He thought it was user-friendly because it looked just like Google Cloud Platform.
5. The highlighted text triggered him to rename the project.
6. He considered highlighted text as 'Edit'.
7. He suggested to distinguish icons (or other representations) for new and recurring project.

8. In 'switch project' dialog box, he suggested that project name should also be clickable (other than the radio button).

**Task 2 - Switch to Image Classification and Upload a Dataset**

1. He easily understood the field to change project type.
2. He chose Image Classification by recognizing the icon first, then reading the description.
3. He was confused what to fill in the 'Dimension' field. He was considering between the .csv table size or image dimension.
4. He was also wondering how the normalization would be done (per row or per column).
5. He understood that normalization would be done for all samples in one feature but he was wondering if the platform knew that the features were put as columns or rows.
6. He liked the explanation of 'Missing Values Encoded' from the 'Information' icon hovered.
7. He used to do 'Missing data removal' in Python with two available options: remove samples with any feature(s) missing or samples with all features missing.
8. He expected if he chose 'Remove missing data', the platform would remove all samples with any feature(s) missing.
9. If there was an option to 'Remove missing data' with samples that only all features are missing, the platform should ask further question about how to treat the remaining missing cells.
10. The best practice to treat remaining missing cells is by replacing the values with mean. Another option are to replace with 0.
11. He said that there are different types of 'Normalization ' so having the explanation about which normalization the platform did could also be useful.

**Task 3 - Change the project into a table classification and start a training with MLP**

1. He managed to change the project type by switching to Home tab.
2. He realized that he had to upload different dataset because he switched to different project type.
3. He easily navigated to Advanced option and selected MLP.
4. He read all available explanations for Basic vs Advanced options and different network types.
5. He expected to switch network type from the down arrow next to 'MLP' in 'Architecture Bar'.
6. He expected to change the Activation Function for Dense Layer and Output.
7. He thought he couldn't manage the data split ratio between Training and Testing because he didn't see in the Architecture tab.
8. He then managed to find the required parameter when he switched to Session tab.
9. He was confused if he had multiple output columns, how he should tell the platform.
10. He suggested the platform could show list of features. If the dataset had header, it could show the feature name. Otherwise, it only showed Column B, Column C, etc.
11. He suggested that 'One hot encoding' could matter to allow users select which features used nominal or ratio scales.

**Overall**

1. He suggested to have a notification or warning if the upload dataset already followed the template.
2. He liked that the platform had already simplified the tasks to train models.
3. He thought that all essentials were already included.
4. He thought the platform avoided users to make a novice user to make an error.
5. He was confused with the feature to add multiple values for one parameter.
6. He considered having multiple values as a hyper-parameter optimization. Grid search could be a more common terminology.
7. When he did steps in Session 1 tab, he was little confused to distinguish Back button and navigate to the previous layer.
8. He suggested to add the destination in each of the 'Back' buttons.

---

| | |
|---|---|
| **USER TESTING** | |
| **Participants** | E |
| **Time** | Sun 19/4 13.00 |
| **Expertise** | Expert |

**General feedback/First Impression**

1. He understood that Home tab worked as the project control panel.
2. He understood that he could pick two types of problem: Classification and Regression.
3. He was wondering if he could perform different techiques other than Classification and Regression.

**Task 1 - Create New Machine Learning Project**

1. He tried to click on the project name because it was highlighted.
2. He thought he was in the circumstance when he was about to rename the project so he started by clicking on random place.
3. He was initially confused because he expected to see a 'Generate new project' button
4. He managed to find the project and it was not hard for him although he didn't realize it at the first place.

**Task 2 - Switch to Image Classification and Upload a Dataset**

1. He figured out how to change the project type from Home tab.
2. He clicked on the dataset download template but he didn't check on the file.
3. He thought the template would follow general convention and the uploaded file already fulfilled the specifications.
4. He managed to upload dataset.
5. He was expecting to get an error message if it didn't follow the template. But, since nothing happened, he thought it was fine.
6. He understood all terminologies for Image Classification dataset.
7. He expected to upload raw image files (not .csv) as dataset.
8. He used to code with Python that could read directly from image (or convert image to binary values in .csv files).

**Task 3 - Change the project into a table classification and start a training with MLP**

1. He liked that the 'Change type' button was straight-forward, obvious, yet safe because it returned a confirmation dialog box.
2. He figured out that he had to upload different dataset for different project type.
3. He thought the platform was fulfilling enough to set important parameters.
4. The platform also had nice interaction to customize and switch between layers (for advanced options).
5. He thought that the 'Activation Function' should be as important as 'Number of Nodes'. He could only access the options under 'Advanced Settings' in each layer.
6. He initially didn't recognize the Session tab (next to Architecture) when he had to set Data Split and Epochs.

**Overall**

1. He liked that the platform was very intuitive.
2. He suggested user should be able to upload dataset in their raw format.

---

**USER TESTING**

| | |
|---|---|
| **Participants** | F |
| **Time** | Mon 20/4 16.30 |
| **Expertise** | Novice |

**General feedback/First Impression**

1. He understood the project name
2. He realized that it's the project type
3. He understood the definition
4. He clicked on session figured out the error message and checked on the dataset

**Task 1 - Create New Machine Learning Project**

1. Misleading because the highlighted
2. Make a new project from top bar
3. He expected to see the select project as home or first landing page
4. He didn't expect the platform to open the last project or any project on my own
5. He expected to select on his own.
6. It was easy

**Task 2 - Switch to Image Classification and Upload a Dataset**

1. Classification is already selected
2. Choose the image classification
3. He read the explanation
4. He went to the dataset
5. He was confused what dimension was
6. He was not sure that he finished the task.

7. He didn't feel any completion after uploading the dataset
8. 'You can only change dataset' should only appear when you want to change dataset.

**Task 3 - Change the project into a table classification and start generating the model.**
1. He figured out easily to switch from home
2. He upload again the dataset
3. The warning  should also include what will be deleted.
4. He understood every definition in the dataset
5. He was wondering if the remove missing data will remove the whole row
6. Missing values encoded should be option instead of blank fields

**Overall**
1. He thought it didn't make sense to have Session 1 when Add Session were not clickable
2. If you can add number to help users to follow the order.
3. Concrete functionality
4. Smooth transition by giving continuity between stages