Design Thinking for Service Enhancement: a case of Teaching Assistant as a Service

Cheng-Zhe Guo

Department of Management Information Systems, National Chengchi University

Rua-Huan Tsaih

Department of Management Information Systems, National Chengchi University

Chieh-Hua Lu

Department of Science and Engineering, Fu Jen Catholic University

Cheng-Zhe Guo

E-mail: jason1120112011201120@gmail.com

Address: No.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 116011, Taiwan

Abstract

This study explores the application of design thinking principles for service enhancement in the context of a Teaching Assistant as a Service (TAaaS). The TAaaS is equipped with MLOps capabilities, enabling students to develop and deploy their own new learning algorithms, codes, and AI models by utilizing their own and others' learning modules while enrolled in the New Learning Algorithms course. By integrating design thinking principles, which emphasize empathy, experimentation, and prototyping, this study aims to enhance the user experience and satisfaction in using the TAaaS system. The challenge lies in allowing students to create their own "new learning algorithm" through trial and error, independently from the multiple pipelines, such as model pipeline, deployment pipeline, and prediction service. Through the iterative and user-centric nature of design thinking, this study demonstrates the potential benefits of incorporating design thinking principles into the service design process, ultimately leading to a more successful AI solution tailored to the users' needs and expectations.

Keywords: AIaaS; MLOps; Design thinking; TAaaS; learning algorithm

1. INTRODUCTION

In recent years, artificial intelligence (AI) has undoubtedly become a prominent technology, with AI as a service (AIaaS) emerging as a popular approach to make AI technology more accessible to businesses and individuals (Lins et al., 2021). One subset of AIaaS, Teaching Assistant as a Service (TAaaS), leverages machine learning (ML) operations, or MLOps, such as continuous integration and continuous deployment to facilitate the development and deployment of new learning algorithms, codes, and AI models. Utilizing a TAaaS system for students enrolled in the course on New Learning Algorithms (Tsaih, 2022) offered by the Department of Management Information Systems at National Chengchi University, Taipei, Taiwan, the focus is on implementing design thinking principles to enhance user experience and satisfaction.

AIaaS aims to make AI technology available and affordable to businesses of all sizes and stages of technological development, guiding users through the development, deployment, and inference of analytical models based on data (Elshawi et al., 2018). By eliminating the need for users to master complex algorithms or technical processes, AIaaS allows users to focus on tasks such as training and customizing AI models without worrying about system installation, maintenance, or management (Boag et al., 2018).

MLOps, borrowing tools and procedures from the DevOps movement, simplifies the development and deployment of ML models (Kreuzberger et al., 2022). Automation of ML model training and testing, integration of ML processes with version control systems, and consistent container deployment are key aspects of MLOps.

Norman (2017) indicated that for AI systems to be effective in the field of education, they must fundamentally be human-centered, addressing the genuine, underlying needs of their intended users. These systems should be structured around the capabilities and requirements of both learners and educators, ensuring that the developed solutions are practical and yield significant benefits. This human-centered design approach aligns well with design thinking, a problem-solving methodology that involves empathy, experimentation, and prototyping. It is an important trend in AI research, as highlighted by Stembert and Harbers (2019), and Riedl (2019), emphasizing the creation of AI solutions that are easy to learn, use, and customize for specific user segments.

In this study, we employ an iterative approach to investigate the influence of research methods on AI services, utilizing TAaaS as our case in this experiment. This TAaaS system covers AI Software Services and AI Developer Services, assisting students in creating new learning modules and integrating existing ones into their learning algorithms. The intention behind this iterative approach is to discuss the impact of design thinking on service design and user experience. This user-centric approach, with its stages of empathizing, defining, ideating, prototyping, and testing, has been fundamental in ensuring that the system is tailored to address user needs, thereby enhancing their overall learning experience. Our exploration centers on three main dimensions:

1. We investigate the feedback collected from two cycles of user interaction and design thinking process (DTP_1 and DTP_2).
2. We compare the user interface and user experience (UI/UX) of two iterations of our service (TAaaS_1 and TAaaS_2).
3. We explore the insights derived from the counter-comparison of the two versions differences.

In conclusion, this study underscores the significance of integrating design thinking principles into the development of AI services like TAaaS. It offers insights into the benefits of a user-centered design approach in creating an intuitive and effective educational tool, thereby contributing to the wider AIaaS and MLOps fields.

2. LITERATURE REVIEW

2.1. Artificial intelligence

The advancement of AI is often discussed in terms of how it contributes to human progress. The recent advances in ML have led to expectations of enhanced efficiency as well as new and improved services for customers and society. These advancements are supported by enormous volumes of accessible data, as well as quickly expanding computer capabilities and public tools and libraries (Stahl et al., 2021). AI is a broad field that deals with the development of intelligent systems that can exhibit human-like behavior and intelligence. AI systems can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation (Lu, 2019). ML is a subfield of AI that focuses on the development of algorithms and models that can learn from data and improve their performance over time (Nagarhalli et al., 2021). ML algorithms use statistical techniques to enable computers to learn from data without being explicitly programmed (Dhillon et al., 2022). Deep learning (DL) is a type of ML that involves the use of neural networks, which are algorithms inspired by the structure and function of the human brain. DL is particularly useful for tasks that involve large and complex datasets, such as image and speech recognition (Zhang and Lu, 2021).

To train DL models, we use learning algorithms, which adjust the model's parameters based on the data it is presented with to make the model more accurate and effective at performing a task (LeCun et al., 2015). These algorithms are used to optimize the parameters of a ML model by minimizing the error between the predicted output and the true output. In general, learning algorithms iteratively adjust the model's parameters based on the input data and the corresponding output labels. The goal is to find the set of parameters that minimizes the error between the predicted output and the true output, so that the model can make accurate predictions on new, unseen data (Shrestha and Mahmood, 2019).

In conclusion, AI offers promising prospects for making positive contributions to the accomplishments and inventiveness of organizations and the progress and development of the community (Rai et al., 2019). The fast development of AI has led to significant changes in people's lives.

2.2. Artificial intelligence as a service (AIaaS)

AIaaS, which stands for "artificial intelligence as a service" or "cognitive services" is a type of cloud-based service that gives businesses and people access to AI tools and apps over the internet (Barlas et al., 2021). As AIaaS is a cloud service, it inherits the advantages and qualities that have established cloud services as a vital information infrastructure for our daily lives.

Cloud services provide ubiquitous, convenient, on-demand network access to a shared pool of customizable computing resources (e.g., networks, servers, storage, applications, and services). These resources may be immediately delivered and released with minimal effort required from the administrative side or the service provider. In addition, Mell et al. (2011) defined a model of cloud computing that is made up of a total of five essential characteristics, three service models, and four deployment models, which are detailed as follows:

- Essential characteristics includes on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.
- Service models contains software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).
- Deployment models covers private cloud, community cloud, public cloud, and hybrid cloud.

AIaaS can include a wide range of services, such as machine learning algorithms, natural language processing, image recognition, and more. AIaaS providers typically offer these services on a subscription or pay-per-use basis, making it more accessible and affordable for businesses and individuals to use AI technology. The concept of AIaaS was conceived because of the intersection of cloud computing and artificial intelligence. AIaaS may be defined as cloud-based platforms that provide on-demand services to people and businesses for developing, training, deploying, and managing AI models (Lins et al., 2021).

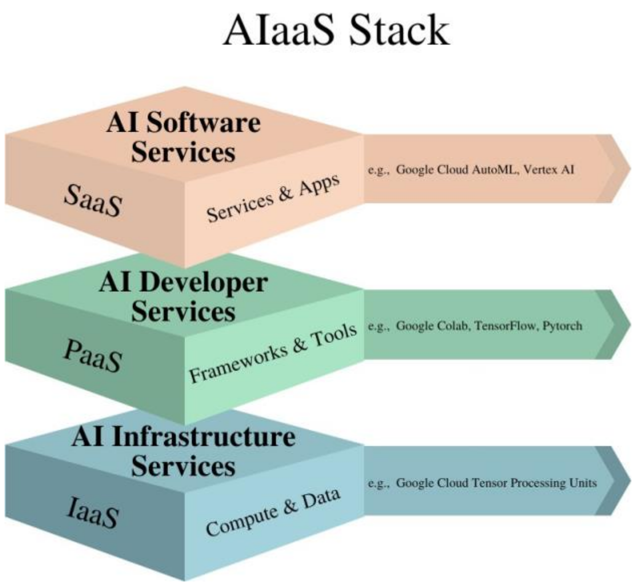


Figure 1 AIaaS stack revised from (Lins et al., 2021).

2.3. Machine learning operations (MLOps)

MLOps is a concept for incorporating ML into the DevOps lifecycle, which is a software development lifecycle. DevOps integrates software development (Dev) and IT operations (Ops) (Ebert et al., 2016). DevOps aims to reduce the development life cycle and deliver applications more often while assuring stability and reliability. DevOps automates software development and IT operations to ensure continuous delivery of high-quality products. It enables firms rapidly deliver apps and services to better serve customers and compete in the market. Continuous integration, delivery, and deployment are DevOps approaches that do this. Automation of build, test, and release procedures and infrastructure management are part of these techniques. DevOps also emphasizes collaboration, communication, and integration between development and operations teams (Leite et al., 2019).

MLOps involves using tools and processes to improve collaboration and communication among data scientists, software engineers, and operations professionals, and to automate the deployment and management of machine learning models (Kreuzberger et al., 2022). This includes managing the infrastructure and runtime environments for ML models, keeping track of how well they work and how accurate they are, and putting models into “production stage”.

Figure 2 depicts a MLOps level 2 architecture proposed by Google Cloud Architecture Center (2020). Some key functions are as follows.

- Source control: This refers to the use of version control systems, such as Git, to monitor and manage changes to the source code and other artifacts associated with an ML project. This ensures that various code versions can be traced and that any modifications made to the codebase can be readily monitored and rolled back if required.

- Test and build services: These are tools and services that are used to automate the testing and building of AI models. This includes tasks such as unit testing, integration testing, and continuous integration (CI).

- Deployment services: These are tools and services that are used to automate the deployment of AI models to “production stage”. This may include tasks such as packaging and releasing code, managing infrastructure, and performing canary releases.

- Model registry: This is a central repository that stores and manages machine learning models and their associated metadata. It allows teams to track the lifecycle of a model, including its training data, performance metrics, and any updates or changes made to it over time.

- Feature store: A feature store is a central repository for storing and managing the features that are used to train machine learning models. It allows teams to track the lifecycle of features, including their origin, transformation, and usage, and to make them easily accessible to data scientists and engineers.

- ML metadata store: This is a repository for storing and managing metadata related to machine learning projects. It may include information such as model training data, hyperparameters, and performance metrics.

- ML pipeline orchestrator: This is a tool or service that is used to automate the orchestration of machine learning pipelines. It may be used to schedule and execute pipelines, monitor their progress, and track their results.

Overall, MLOps aims to improve the efficiency and reliability of machine learning projects by standardizing and automating processes, and by providing tools and services that support collaboration and traceability.

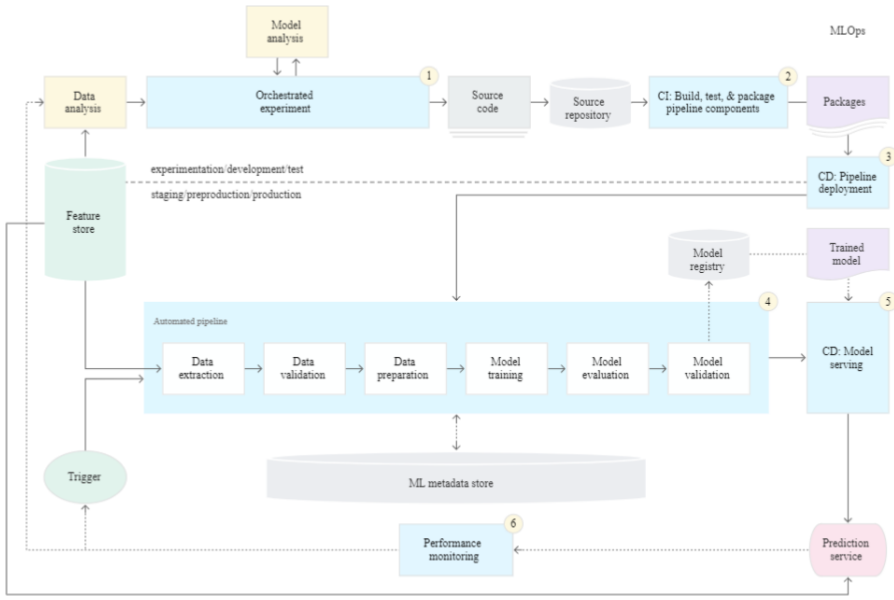


Figure 2 MLOps level 2: CI/CD pipeline automation (Google Cloud Architecture Center, 2020).

2.4. Human-Centered Design and Design Thinking

Human-centered design is a design framework which prioritizes user needs and experiences and has become a significant movement worldwide. According to human-centered design pioneer Donald Norman, modern design should focus on the needs of those who will interact with the product or service because every product or service involves people (Gasson, 2003, Norman, 2017, Xu, 2019).

Design thinking (DT) is a human-centered and user-centric interactive approach to innovation and problem solving (Brown et al., 2008). It is a methodology for the teams or organizations to design better products, services, or experiences and it has been applied in many fields such as healthcare, business, education, IT Industry and more (Hasso Plattner Institute of Design at Stanford University, 2023).

Nobel Prize laureate Herbert A. Simon (Brown and Martin, 2015) first advanced the concept of DT in 1969, defining design as “transformation of existing situations into preferred ones” and design thinking as “way of thinking.” The DT well-known guru Tim Brown, the co-chair of global design and innovation firm IDEO, describes DT as “a human-centered approach to innovation.” Design thinking has been developed by researchers of Stanford d. school and the IDEO (Kelley, 2001). The overall objective of DT is to generate innovative concepts based on a deep understanding of what users need and want to create a desirable, feasible, and viable solution. It is an iterative, non- linear process by which there are three spaces to keep in mind: inspiration, ideation, and implementation, for developing new ideas in an innovative and user-centered way (Brown and Katz, 2011).

Studies showed DT can make valuable contributions to software development and create products or services based on user needs. Lindberg et al. (2011) mentioned that DT can help engineers to define problems more precisely so that their expert knowledge may be applied suitably. Pereira and de F.S.M. Russo (2018) noted that the use of a DT approach helped checking both technical and non-technical factors and their study showed in some cases the quality of the software and the satisfaction of the users significantly increased.

There are five important phases of the DT cycle: 1. empathy toward users, 2. defining the problem, 3. ideation, 4. prototyping, and 5. testing; the phases would move through the cycle iteratively (Hasso Plattner Institute of Design at Stanford University, 2023). The users are always at the center of the different process phases. DT begins with understanding the challenges and empathizing with the end users. Then it gathers insights and identifies the problems, and at the third phase it generates ideas and solutions, then following by prototyping, testing, getting users' feedback and iterating.

An experimental study on Deutsche Bank's IT division showed that the DT approach helped integrate customers into the organization's innovation process and provided an efficient and effective way to launch user-centric service in a short time (Vetterli et al., 2016). IBM has been using DT to explore the problems and uncover the clients' spoken and unspoken needs and wants, and then validate and iterate before a product or service released to the market (Clark et al., 2008, Lucena et al., 2017).

Verganti et al. (2020) point out that AI reinforces the principles of DT, especially human-based activities often requiring significant investment of resources and time. Their study showed that

DT and AI can empower each other. DT helps empower a more effective, human-centered implementation of AI, and AI helps empower a more advanced practice of DT. They noted that, AI is intrinsically iterative and delivers through loops. The loops can leverage the most recent data and algorithms, and also offer a new opportunity of further learn.

3. RESEARCH METHODOLOGY

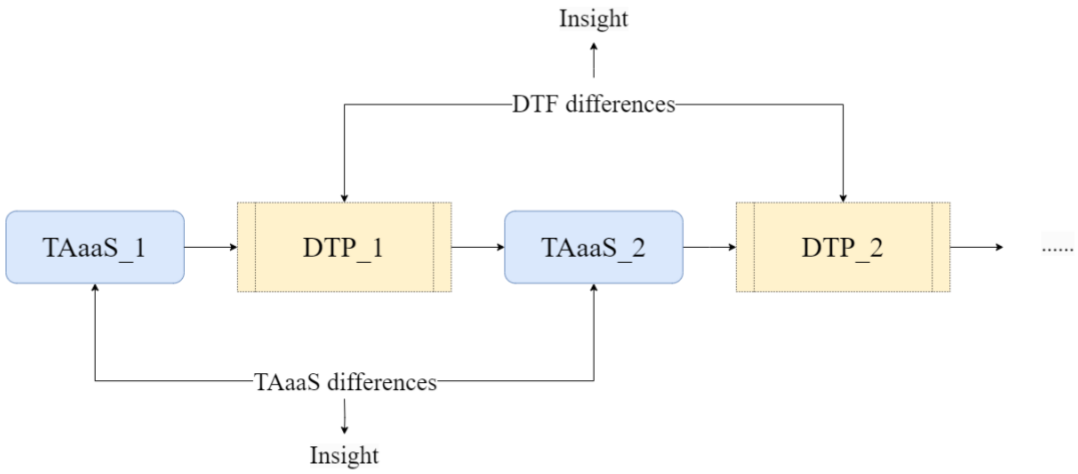


Figure 3 Research methodology diagram

This research aims to explore the influence of design thinking on service design and user experience, with a focus on an iterative process that incorporates user feedback at multiple stages. In the first design thinking process (DTP), participants interact with the initial version of the Teaching Assistant as a Service, referred to as TAaaS_1, which has been developed without explicit use of design thinking principles. TAaaS_1 integrates the functionalities of AI Software Service, AI Developer Service, and Machine Learning Operations (MLOps) to facilitate a more effective learning process for students engaging in AI studies. A detailed system architecture is discussed in Chapter 4.

Feedback derived from this interaction form the foundational structure of design thinking feedback (DTF), primarily focusing on user satisfaction, the usability of the system, and the overall participant experience. This approach is consistent with the principles of design thinking outlined by Brown et al. (2008), which emphasize a user-oriented, iterative process encompassing the understanding of user behaviors, empathizing, ideating, and testing prospective solutions. Through DTP_1, we define the challenges within TAaaS_1 based on feedback (DTF_1) from participants, thereby instigating the ideation process and leading to the creation of enhanced service prototypes. Following additional rounds of user feedback, these prototypes are refined, leading to the development of the redesigned service, TAaaS_2. Each participant experiences both the initial and redesigned services, which allows us to control for individual differences and potential confounding variables. Upon completion of the interaction with both versions of the service, participants' feedback is collected again to evaluate their experience with the redesigned service, TAaaS_2. The comparison of participant responses to the two versions of the service allows us to assess whether the application of design thinking principles through DTP_1 has led to significant improvements in service design and user experience. This cyclical process is broken down into three sections:

1. Differences in feedback gathered from the two cycles of user interaction and design thinking process (DTP_1 and DTP_2).
2. Differences in UI/UX between the two iterations of our service (TAaaS_1 and TAaaS_2).
3. Insight derived from the counter-comparison of the two versions differences.

In conclusion, this integrated research methodology allows us to systematically investigate the impact of design thinking on service design and user experience. Utilizing the iterative design approach, and comparing participants' responses to the two service iterations, we aim to provide valuable insights into the potential benefits of incorporating design thinking principles into the service design process, keeping in view users' perspectives and feedback.

4. THE TAAAS AND ITS IMPLEMENTATION

In this research, our primary focus is exclusively on examining the UI/UX components of TAaaS. While TAaaS is inspired by and indeed incorporates elements of the Google's MLOps level 2 concept, specifically CI/CD pipeline automation (Google Cloud Architecture Center, 2020), and also integrates the AI Software Services and the AI Developer Services from the AIaaS stack (Lins et al., 2021), we will not be delving into these aspects in detail. Our intention is not to explore the intricacies of MLOps or the broader AI service offerings, but to concentrate solely on how the UI/UX elements of TAaaS can be optimized and improved. This means the investigation will be directed at understanding and enhancing the way users interact with and experience TAaaS, with no in-depth discussion on MLOps. Consequently, the architecture of the system under review is divided into two sections, aimed at providing AI Software Services and AI Developer Services, respectively, but our research will only be concerned with the UI/UX facets of this arrangement.

4.1. The AI Software Services

Lins et al. (2021) described AI software services as ready-to-use AI applications and building components, akin to the traditional SaaS (Software as a Service) cloud layer. For users seeking to utilize an existing model, the AI Software Services provide pipelines and modules. The development team is responsible for creating new pipelines and modules that allow users to complete the model pipeline, deployment pipeline, and prediction service within the MLOps framework. In this study, we initially developed a pipeline for homework 1 (Figure 4), assigned in the New Learning Algorithms course, to facilitate the experiments we conducted in the Chapter 5.

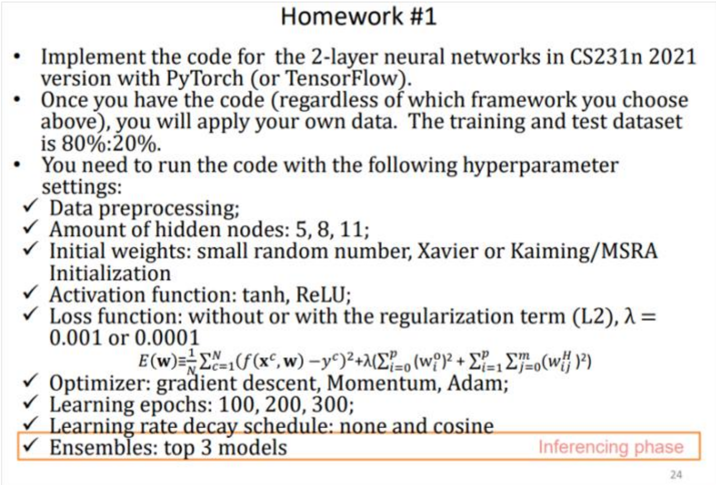


Figure 4 Homework 1 assigned in the New Learning Algorithms course

4.2. MLOps for The AI Software Services

The AI Software Services are responsible for supporting users in directly creating and deploying neural network models in the "production stage" by using the pre-defined pipelines developed in the "development stage" by the system engineer. The terms "development stage" and "production stage", which refer to the concept of MLOps, compose the diagram depicting the proposed implementation of MLOps of the AI Software Services (Figure 5). When users attempt to upload datasets, train new AI models, and implement those models in services, these steps are done in the AI Software Services. Under the proposed implementation of MLOps of the AI Software Services, the flow architecture in the "development stage" is nearly equivalent to the original architecture (Google Cloud Architecture Center, 2020). Instead, the flow in the "production stage" differs as follows:

1. Model pipelines:
    a. Data preparation: we exclude certain data-related components, such as data analysis and data extraction, because users in this use case should prepare the dataset before accessing the pipelines. When users finished the step of data preparation, the uploaded datasets would be stored as CSV files on the server and organized into folders depending on the data subjects and data usage. Data subjects include, but are not limited to, "solar power generation forecast" and "prediction of low birth weight in newborns." Data usage includes "training" and "testing" datasets.

b. Model training: this proposed model pipeline covers model training pipelines of homework 1, assigned in the New Learning Algorithms course.

c. Model validation: there is a difference between the definition proposed by Google Cloud Architecture Center (2020). The procedure for assessing the stability and generalizability of an AI model is the definition we agreed on. Typically, model validation is accomplished by splitting the data into training and validation sets and utilizing the validation set to measure the performance of the model. The purpose of model validation is to confirm that the model has not overfitted to the training data and can accurately predict new, unobserved data (Goodfellow et al. 2016).

d. Model evaluation: the model is evaluated on the training dataset and the validating dataset, which are split in the ratio of 8:2 from the original training dataset. The output of this step is a set of metrics and figures to assess the quality of the model. Furthermore, the model's metrics and figures will be stored on the server.

2. Deployment pipeline:

   a. Model deployment: the deployment state of models is stored on the MongoDB container; after a model has been deployed, the deployment status of that model will switch from "revoking" to "deploying." The GitLab CI/CD tool will then immediately and automatically start the deployment of the model through Docker on the server.

   b. CD: Image building: the step of building a Docker image based on the deployed model.

   c. CD: API building: the step of building a Docker container based on the built image. After the container is completely enabled, the information about the container, which is the model that has been deployed, such as the container ID and port, will be updated in the MongoDB container, and then the web-based UI will be updated synchronously.

3. Model API: after the CD pipeline is finished, the enabled container becomes an AI model API.

4. Prediction service:

a. Data preparation: this step in the prediction service is substantially similar to the step in the model pipeline, with the exception that the testing dataset is employed here.

b. Model prediction: the main service will request the Docker container and obtain the loss function value for the testing dataset.

c. Performance monitoring: the loss function value of the testing metric, as well as the data from previous training experiments, will be imported into the server from the metadata file corresponding to the model.

5. Manually trigger: the component of continuous training in the original architecture (Google Cloud Architecture Center, 2020) is manually triggered here because it is not an essential function in this use case. As a result, we designed the web-based UI for users to check the model's performance by themselves.

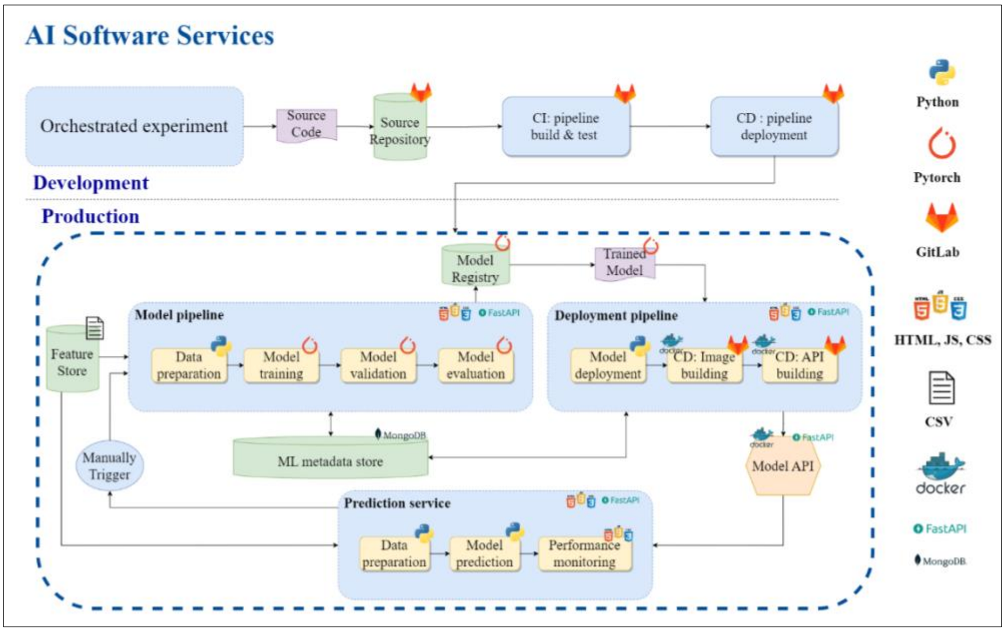


Figure 5 The proposed implementation of MLOps of the AI Software Services.

4.3. The AI Developer Services

Lins et al. (2021) characterized AI Developer Services as tools designed to assist developers in implementing code to unleash AI capabilities, analogous to the traditional PaaS (Platform as a Service) cloud layer. For users aiming to develop new algorithms, the AI Developer Services offers a range of tools and resources to create and deploy learning modules within the AI Software Services.

The development team plays a crucial role in designing and maintaining these tools, ensuring seamless integration with existing pipelines and modules. In this study, we leveraged the AI Developer Services to build custom algorithms tailored to the specific requirements of homework 1 in the New Learning Algorithms course, which we then utilized in the experiments detailed in Chapter 5.

4.4. MLOps for The AI Developer Services

The primary duty of the AI Developer Services is providing users with the resources and materials necessary to construct and deploy learning modules from the “development stage” to the “production stage.” This transition takes place from the “development stage” to the “production stage.” The terms “development stage” and “production stage”, which refer to the concept of MLOps, compose the diagram depicting the proposed implementation of MLOps of the AI Developer Services (Figure 6). Both the MLOps and DevOps theories constantly make reference to the two stages that we went over in the previous section. Users with the objective of using an existing model are closer to the “production stage” in the AI Software Services of this proposed TAaaS, whereas users with the objective of developing new algorithms are closer to the “development stage” in the AI Developer Services. Traditionally, the “development stage” of MLOps has been designated for data engineers and data scientists, who utilize it to design pipelines and conduct experiments. In this study, when users attempt to build new learning modules, which is referred to as the “development stage,” and once the submitted learning module has been validated and deployed to the “production stage,” are the AI Software Services able to apply the new learning modules. According to the MLOps that have been proposed for the AI Developer Services, the flow architecture in the “development stage” is partially analogous to the architecture that was originally designed. Alternatively, the flow is different in the “production stage,” as follows:

1. Orchestrated experiment: the orchestrated experiment in “development stage” often refers to the ability for researchers or engineers to explore the topic in which numerous components or pipelines are coordinated and controlled to generate a desired result. In this research,

however, users rather than a system engineer were designated for this task. Developer users are distinguished from the software users in that they only apply to previously defined pipelines in the "production stage". They can construct a customized pipeline component, which primarily relates to learning modules, and deploy it in the "production stage," making customized learning modules available to software users.

2. CI/CD: module build, test, and deployment: there exists a distinction between the AI Software Services and the AI Developer Services. In the first scenario, Docker will not be used to perform pipeline operations. Because in the latter scenario, we would conduct any testing necessary on container.
3. Module API: after the CI/CD pipeline is finished, the enabled container becomes a learning module API.
4. AI Software Services: the generated learning module API is now accessible to the model pipeline in the AI Software Services.

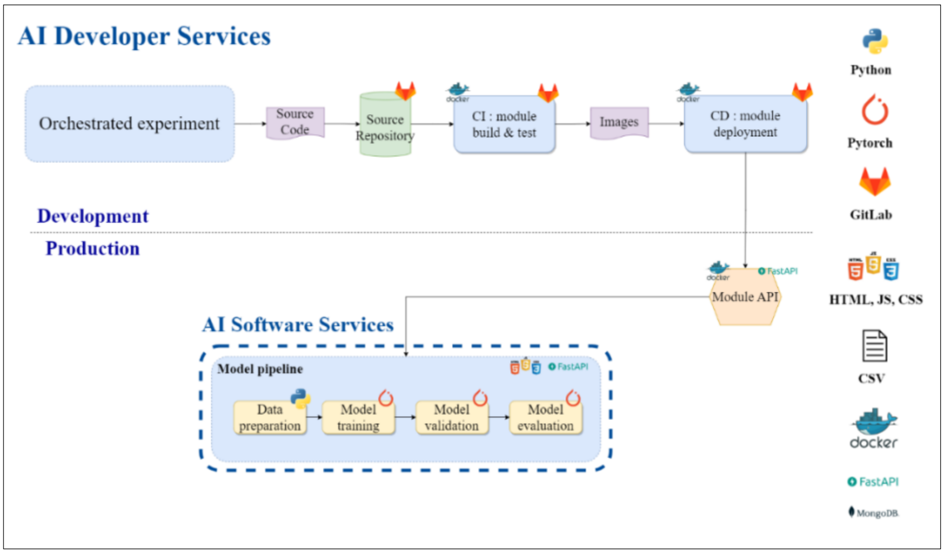


Figure 6 The proposed implementation of MLOps of the AI Developer Services.

5. EXPERIMENT

5.1. Research participants

The participants in the study were selected based on the quality of their submissions for homework 1 in the course on New Learning Algorithms offered by the Department of Management Information Systems at National Chengchi University, Taipei, Taiwan, as determined by a review process. The process emphasizes whether the students go beyond the basic requirements of completing the assignment and even engage in further comparisons and analyses among different models.

The study group consisted of 5 participants who conducted the experiments together. They are between the ages of 23 and 31, with an average age of 25.4 years. There are 2 males and 3 females in the group. All of the respondents are in a Master's program, with 4 in their first year and 1 in their second year. The most common age in the group is 24, which is also the median. The standard deviation of the ages is 2.97, indicating that the ages are relatively close to the mean.

5.2. Experiment design

5.2.1. DTP_1

We are selecting participants from students who have previously completed homework 1 in order to ensure that these participants satisfy the first step of the design thinking cycle: empathy. The participants are instructed to complete several tasks using TAaaS_1.

Table 1 Experiment tasks in DTP_1

- Task 1 - AI Software Service phase

1. Use your personal laptop to enter the TAaaS entry website: http://140.119.19.87/entry
2. Follow the webpage path to enter the hw1 page: AI Software Service -> Model pipeline -> Homework #1.
3. According to the instructions on the webpage, fill in the required hyperparameters for model training and train a model in sequence. Note that there is no need to upload a dataset under data preparation, as you can use the system' s default dataset (e.g., hospice, solar).
4. Check the model performance and training configuration.

- Task 2 - AI Developer Service phase

1. Use your personal laptop to enter the TAaaS entry website: http://140.119.19.87/entry
2. Enter the AI Developer Service UI.

3.	Download the example program for homework 1.
4.	Follow the instructions in the hw1.py program to complete the steps.
5.	Fill in the required fields.
6.	Compress the hw1.py program.
7.	Rename the compressed file as hw1-student ID.zip, e.g., hw1-110356021.zip.
8.	Return to the AI Developer Service UI and upload the compressed file.
9.	Enter the Ensemble page and search for the trained model using your student ID.
●	Task 3 - Ensemble phase
1.	First, train at least three models on AI Software Service/AI Developer Service using the same dataset configuration.
2.	Enter the Ensemble page: http://140.119.19.87/pipeline/model/hw1/ensemble
3.	Search for the trained model using your student ID.
4.	Use the sorting function in the table to view the training loss or validating loss of the models in different orders.
5.	Select the top three models with better training results and submit.
6.	Check the validating loss of the Ensemble model.

After completing the tasks, we engage in a discussion about the user experience of The TAaaS_1 by addressing the following questions.

Table 2 Discussion topics in DTP_1

1	Do you think the information provided on the UI can guide you to complete the required task? Why?
2	Do you think this UI provides sufficient guidance and assistance to help you complete the required task? Do you need additional help or guidance to complete the task?
3	What are the differences between completing the assignment on your own and completing it with the assistance of the system?
4	When completing assignments on your own, what system-assisted tools do you consider using? How do these tools help you complete tasks more effectively?
5	Do you think the overall use process is smooth? Did you encounter any difficulties or setbacks? Is there any part that confuses you or makes you want to stop using it?
6	Based on your user experience, are there any elements that can be added to the website to help you complete the task more smoothly?
7	Briefly describe your feelings after using the system today.
8	If you use the system first and then complete the assignment on your own, does it deepen your learning process?

5.2.2. DTP_2

The participants are instructed to complete several tasks using The TAaaS_2.

Table 3 Experiment tasks in DTP_2

● Task 1 - AI Software Service phase
1 Access the TAaaS portal using your personal laptop at http://140.119.19.87/entry.
2 Navigate to the hw1 page under AI Software Service -> Model pipeline -> Homework #1.
3 Follow the on-screen instructions to complete the model training process using the system's default datasets (e.g. hospice, solar).
● Task 2 - AI Developer Service phase
1 Access the TAaaS portal using your personal laptop at http://140.119.19.87/entry.
2 Navigate to the online editor page under AI Developer Service.
3 Follow the on-screen instructions to complete the model training process.
4 If the process is completed successfully, you will be automatically redirected to the Ensemble page.
● Task 3 – Ensemble phase
1 Using the same dataset, train three or more models using the first two services.
2 Access the Ensemble page at http://140.119.19.87/pipeline/model/hw1/ensemble.
3 Search for the trained models using your student ID.
4 Use the website's features (e.g. model training information, table sorting) to select the top three best-performing models and submit your choices.
5 Check the validating loss of the ensemble model.
● Wait patiently for the other participants to complete the experiment.
● You can take notes about your experience with the system in the provided document.

After completing the tasks, we engage in a discussion about the user experience of The TAaaS_2 by addressing the following questions.

Table 4 Discussion topics in DTP_2

1	Do you think the information provided on the UI is sufficient to guide you in completing the required tasks? Why?
2	Do you think this UI provides sufficient guidance and assistance to help you complete the required task? Do you need additional help or guidance to complete the task?
3	What are the differences between completing the assignment on your own and completing it with the assistance of the system?
4	Do you think the overall use process is smooth? Did you encounter any difficulties or setbacks? Is there any part that confuses you or makes you want to stop using it?
5	Based on your user experience, are there any elements that can be added to the website to help you complete the task more smoothly?
6	Did making improvements to the system based on the feedback received from classmates during the DTP_1 make it better aligned with the needs of the students in the class?

7	Briefly describe your feelings after using the system today.
8	If you use the system first and then complete the assignment on your own, does it deepen your learning process?

5.3. Feedback difference between two DT cycles

5.3.1 DTP_1

During the study, participants are asked to complete multiple tasks using The TAaaS_1, followed by a discussion of their user experience. In the design thinking cycle, step 2 entails defining the problem. The feedback collected from discussions reveals a range of user experience issues. These issues, impacting the learning process, represent the “problems” we aim to define in this step. Some students report that the process is smooth, while others encounter difficulties such as unclear operations and opaque mechanisms behind the Ensemble function. Regarding the impact of the system on the learning process, responses are mixed, with some students finding it helpful for understanding hyperparameters, but still needing to write the code themselves. Others feel that the system do not significantly aid their learning experience.

These feedback suggestions, including adding system elements such as remembering previous training hyperparameters, providing comment guidance, and allowing a code editor on the webpage, are the outcomes of the design thinking cycle step 3: ideation. This table presents user feedback as design thinking feedback (DTF) on expectations and suggested improvements for the service. It is created by the research team after gathering ideas from participants who use the service during the study.

Table 5 Feedback derived from the DTP_1

Index	Category	Feedback
DTF_1-1	Function	Provide field memory and layout retention functionality to enhance user experience.
DTF_1-2	UI	Provide a flowchart to help users clearly understand the entire process steps.
DTF_1-3	UI	Structure blocks, including data preparation, model training, and model performance, to differentiate different sections.
DTF_1-4	UI	Provide a pop-up window to view model training hyperparameters for easy comparison and analysis.

DTF_1-5	Function	Provide external links for forward and backward knowledge to help users better understand the operational principles of the model.
DTF_1-6	Function	Provide data description functionality to more clearly explain the features and attributes of the dataset.
DTF_1-7	Function	In addition to loss value, provide output value for a more comprehensive evaluation of model performance.
DTF_1-8	Function	Allow users to download model files and code for further application and improvement.
DTF_1-9	Function	Provide a usage comparison table for different packages, such as PyTorch and TensorFlow, to facilitate users in selecting the appropriate package.
DTF_1-10	Function	Support compatibility between multiple package usages to enhance system openness and flexibility.
DTF_1-11	Function	Ensemble mechanism is unclear.
DTF_1-12	Function	Provide more text descriptions (comments) to explain the use of hyperparameters, helping users better adjust model design and parameters.
DTF_1-13	Function	Provide compiler extension functionality to improve the efficiency and convenience of writing code for users.
DTF_1-14	Function	Provide compiler and debug functionality to improve the efficiency and convenience of writing code for users.

After making these following adjustments, the outcome is TAaaS_2, which incorporates the selected feedback and ideas from Table 5(including: DTF_1-1, DTF_1-2, DTF_1-3, DTF_1-4, DTF_1-8, DTF_1-12, and DTF_1-13) into the system's functionality, following step 4 of the design thinking cycle: prototype. We will complete the final stage of the entire design thinking cycle, which is testing, within DTP_1. This involves conducting a full DT process again in DTP_2.

5.3.2 DTP_2

During step 1 of DTP_2, participants have already met the conditions for double empathy: they have completed homework 1 from the class and the TAaaS_1 experiment. As for steps 2, 3, and 4, they follow the experimental procedure of DTP_1.

During the discussion, participants shared their experiences using the redesigned TAaaS_2 system to complete various tasks. Most of them found the process smooth and appreciated the step-by-step guidance provided. They also encountered some difficulties, such as the lack of warnings for missing hyperparameters in the Developer section and the unclear operations in the Ensemble function. Participants provided valuable feedback, suggesting the addition of features like visualizing loss comparisons, offering back buttons for easier navigation, and enhancing Ensemble function usability.

In terms of the system's impact on learning, students had mixed opinions. Some found it helpful for understanding hyperparameters and facilitating the completion of tasks, while others still preferred writing the code themselves for a greater sense of accomplishment. Overall, most participants agreed that the system, particularly after incorporating feedback from DTP_1, better addressed the needs of students taking the course.

This table presents user feedback on the expectations and suggested improvements for the service. It was created by the research team to gather feedbacks from participants who used the service during the study.

Table 6 Feedback derived from the DTP_2

Index	Category	Feedback
DTF_2-1	UI	Provide a button that allows users to return to the initial stage of the training process, making it easier to retrain the model.
DTF_2-2	UI	On the Entry page, include a section that leads directly to the Ensemble feature, corresponding to the three task topics.
DTF_2-3	UI	On the Ensemble page, prompt users to select 1 to N models before submitting, and add a section to directly access the Ensemble feature.
DTF_2-4	Function	Add loss charts to the model links so users can view the training results.
DTF_2-5	Function	In the Ensemble page, use a visualization interface to display the loss comparison charts of all or selected models.
DTF_2-6	UI	Choose a single presentation method for the Online Editor and the code upload/download feature to avoid confusion.
DTF_2-7	Function	Add foolproof features for hyperparameters in the Developer panel, such as warnings for empty value submissions.

5.3.3 Comparison between two DT cycles

Design thinking is a systematic, human-centered approach to problem-solving that is typically applied to improve product design, user experience, and user interface. In both DTP_1 and DTP_2, feedback was gathered from users who interacted with the TAaaS system. The feedback from these sessions were used to understand user pain points, usability issues, and areas of improvement for the system.

1. DTP_1: During the first session, participants reported a mixed experience using The TAaaS_1. Issues included unclear operations and the lack of a transparent mechanism behind the Ensemble function. However, participants also provided valuable feedback that helped identify areas for system improvement. These suggestions were largely centered on enhancing the software's user experience and functionality, with suggestions such as adding system elements for remembering previous training hyperparameters, providing a flowchart for the process, and allowing for a code editor on the webpage. The developer-related feedback focused on improving code-writing efficiency and convenience.
2. DTP_2: After incorporating feedback from the first session, the second round of design thinking was conducted on the redesigned TAaaS_2 system. Participants found this iteration smoother, with step-by-step guidance provided. However, there were still issues identified, such as the lack of warnings for missing hyperparameters and unclear operations in the Ensemble function. The feedback from this round of testing was geared toward additional feature suggestions, such as visualizing loss comparisons and offering back buttons for easier navigation. User suggestions also focused on better streamlining and user-proofing the developer section of the service.

Overall, it appears that the iterative design thinking process helped improve the system from TAaaS_1 to TAaaS_2, with feedback from users being incorporated into design changes that increased overall user satisfaction and addressed user needs more effectively. However, both rounds also highlighted that further improvements could still be made, particularly in the areas of system clarity, user guidance, and feature augmentation.

5.4. UI/UX difference between two services

The following table outlines the features and reasons for improvement that I have chosen based on user feedback gathered by the research team during the study.

Table 7 Selected feedback from DTP_1 for Improvement

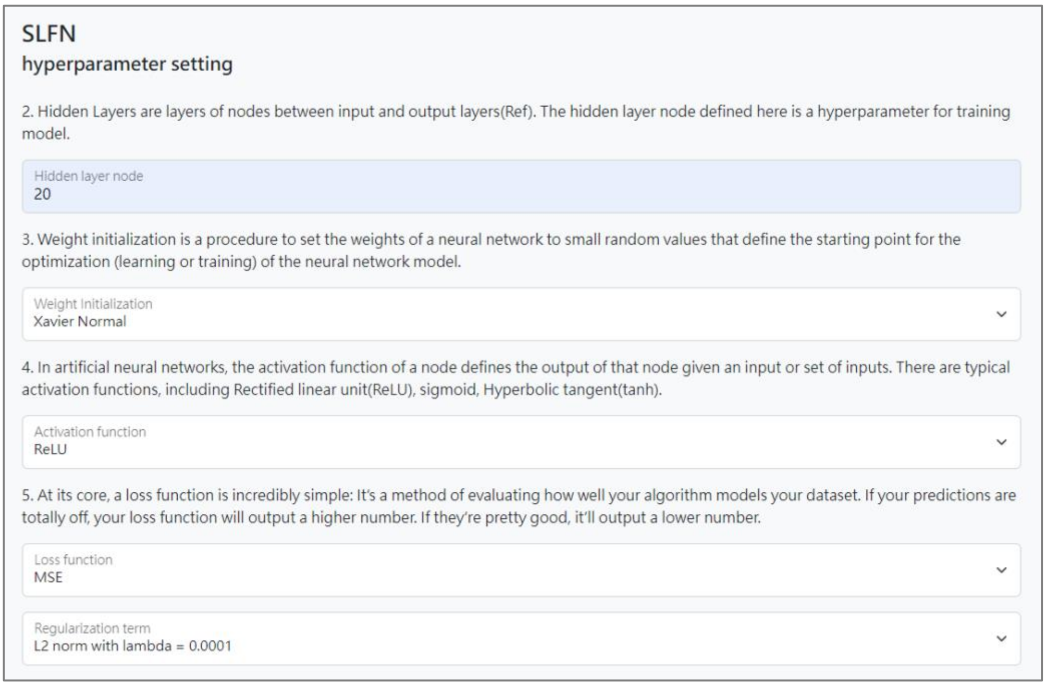
Index	Improve	Reason
DTF_1-1	Y	Enabling fields to remember previous input values and the webpage to remember the previous layout position can enhance user experience and efficiency.
DTF_1-2	Y	Providing a flowchart can improve user understanding and mastery of the entire process.
DTF_1-3	Y	Segmenting blocks for data preparation, model training, and model performance can help users better understand and operate them.
DTF_1-4	Y	Providing a pop-up window to view model training hyperparameters by clicking on a model in the model list can make it more convenient for users to compare and analyze models.
DTF_1-5	N	The feature requiring additional external links may increase development time and cost as it requires additional research and organization, potentially adding unnecessary confusion and learning burden for users. Therefore, it may be temporarily shelved.
DTF_1-6	N	Providing more detailed data descriptions may require more time and resources for organization and editing, and may undergo significant changes due to differences in data format and type, increasing development cost and time. Therefore, it may be temporarily shelved.
DTF_1-7	N	Providing more training information may require more detailed monitoring and output of the model, increasing development time and cost and potentially affecting the model's efficiency and accuracy. Therefore, it may be temporarily shelved.
DTF_1-8	Y	There is already a feature to download model files, but providing an additional feature to download code may require more complex processing and potentially create additional issues due to differences in code type and format. Therefore, it may be temporarily shelved.
DTF_1-9	N	Organizing and comparing different packages for a usage comparison table may require more time and resources for organization and editing and may undergo significant changes due to differences in

		package version and functionality. Therefore, it may be temporarily shelved.
DTF_1-10	N	Integrating and adjusting different packages for compatibility may require more time and resources for organization and editing and may undergo significant changes due to differences in package version and functionality. Additionally, since this feature may only apply to specific users, not all users require this functionality, and temporarily not developing it can reduce development cost and time.
DTF_1-11	N	Since the Ensemble mechanism is unclear, developers may need to conduct more research and testing to determine the appropriate implementation method, increasing development time and cost. Additionally, since this feature may only apply to specific users, not all users require this functionality, and temporarily not developing it can reduce development cost and time.
DTF_1-12	Y	This feature can provide more detailed explanations to help users better understand the purpose of hyperparameters and how to adjust them.
DTF_1-13	Y	Providing compiler extension functionality can improve user efficiency and convenience in writing code.
DTF_1-14	N	The feature requiring additional debugging functionality may require more complex processing of the code, increasing development time and cost. Additionally, users may already be using other debugging tools while writing code, so it may be temporarily shelved.

Here is a list of feedback I obtained from DTP_1 and selected as improvement requirements for the system. I have also provided a system screenshot for reference.

- DTF_1-1

Initially, in TAaaS_1, the web form fields had fixed default values. After the adjustment based on the feedback from DTF_1-1, they were modified to retain the user's previous input data.


SLFN

hyperparameter setting

2. Hidden Layers are layers of nodes between input and output layers(Ref). The hidden layer node defined here is a hyperparameter for training model.

Hidden layer node
20

3. Weight initialization is a procedure to set the weights of a neural network to small random values that define the starting point for the optimization (learning or training) of the neural network model.

Weight Initialization
Xavier Normal

4. In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs. There are typical activation functions, including Rectified linear unit(ReLU), sigmoid, Hyperbolic tangent(tanh).

Activation function
ReLU

5. At its core, a loss function is incredibly simple: It's a method of evaluating how well your algorithm models your dataset. If your predictions are totally off, your loss function will output a higher number. If they're pretty good, it'll output a lower number.

Loss function
MSE

Regularization term
L2 norm with lambda = 0.0001

Figure 7 DTF_1-1 on TAaaS_2

- DTF_1-2 and DTF_1-3

Based on the feedback from DTF_1-2 and DTF_1-3, compared to TAaaS_1, in TAaaS_2, we provide an implementation phase flowchart and segment the stages, allowing users to progress step by step through tasks such as data preparation, model training, and model performance.

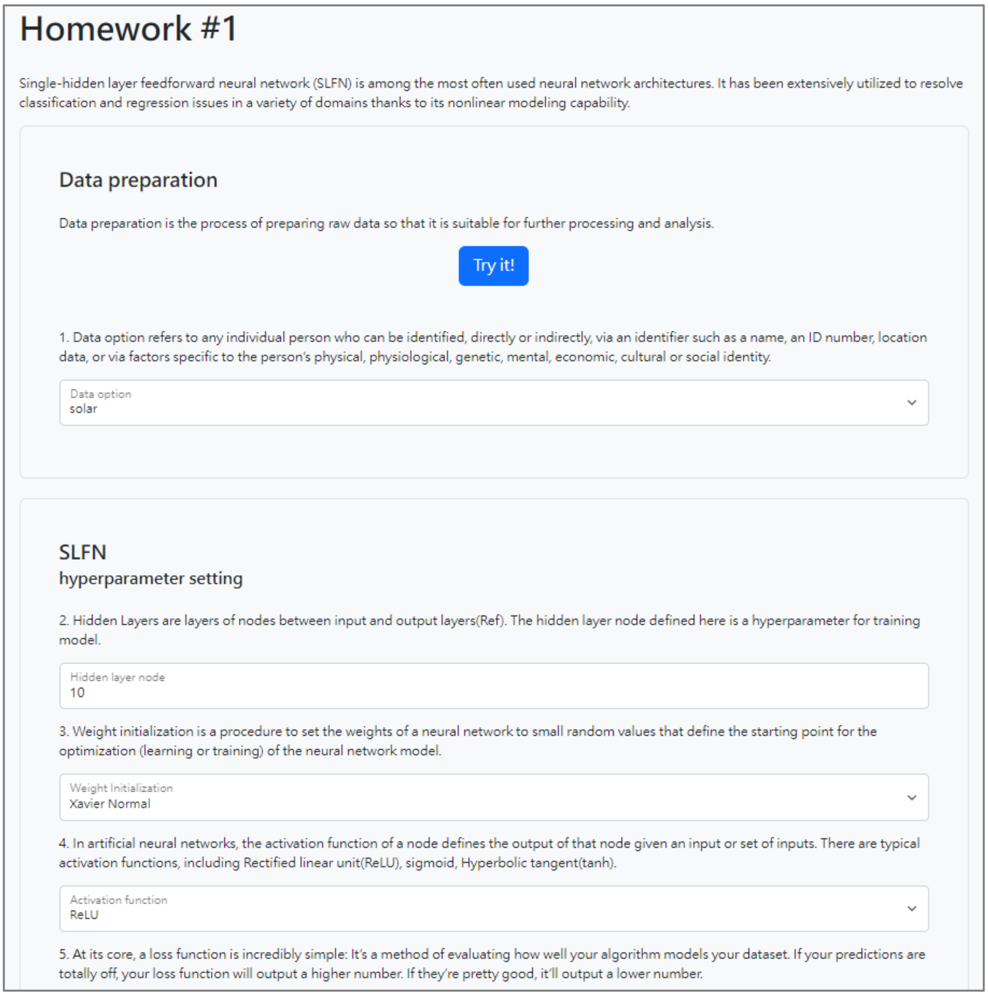


Figure 8 DTF_1-2 and DTF_1-3 on TAaaS_1

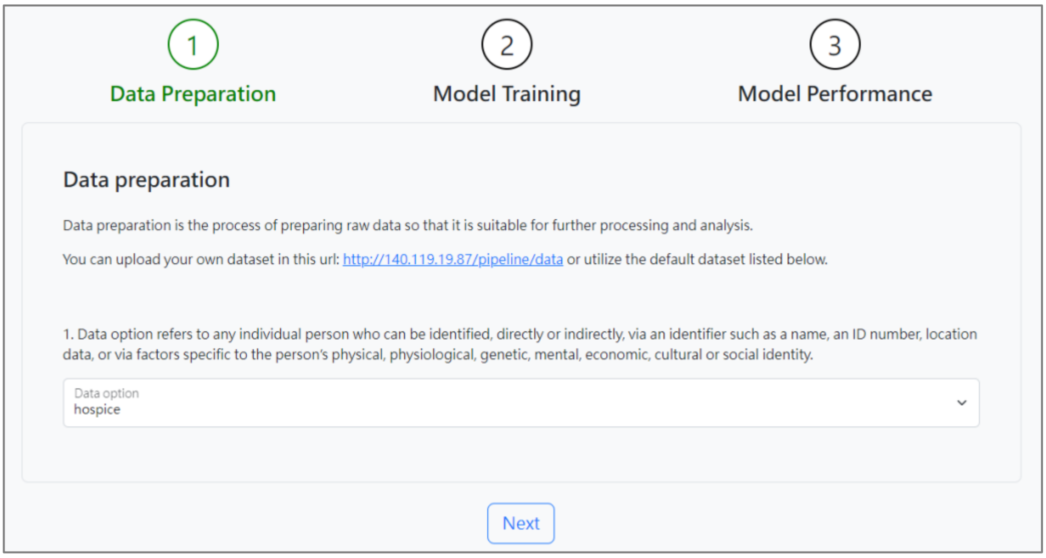


Figure 9 DTF_1-2 and DTF_1-3 on TAaaS_2

- DTF_1-4

Based on the feedback from DTF_1-4, we provide users with the functionality to query the record of model training hyperparameters on the Ensemble page.

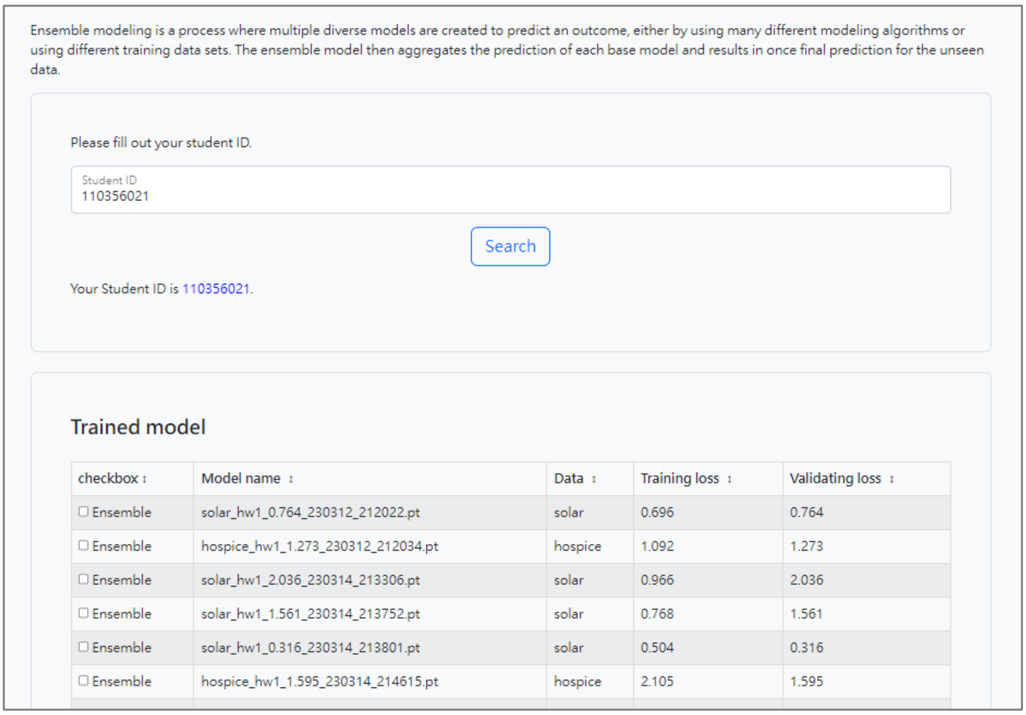


Figure 10 DTF_1-4 on TAaaS_1

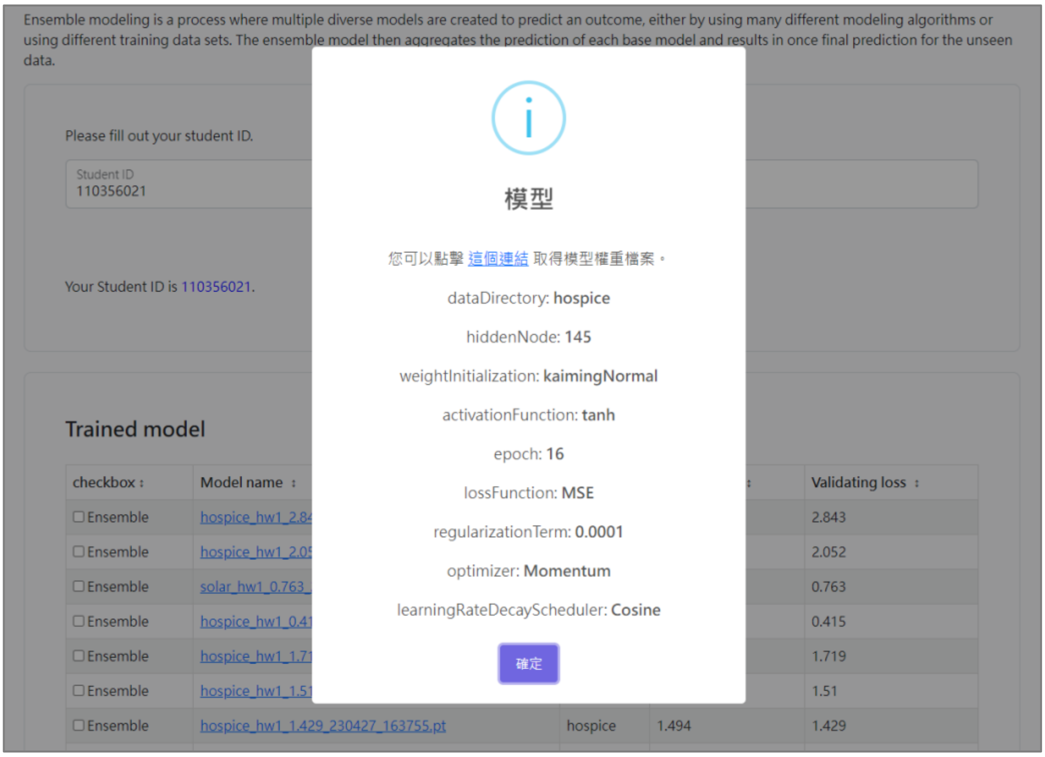


Figure 11 DTF_1-4 on TAaaS_2

- DTF_1-8

Based on the feedback from DTF_1-8, while providing users with the functionality to query the record of model training hyperparameters, we also enable them to download the model's weight storage file (PyTorch file).

Figure 12 DTF_1-8 on TAaaS_2

- DTF_1-12 and DTF_1-13

Originally, the design allowed users to submit code by downloading and uploading files, which would generate new models or modules. After the feedback from DTF_1-12 and DTF_1-13, the required information and method for code submission were designed directly on the web page with a web compiler provided. However, there was no debugging functionality offered.

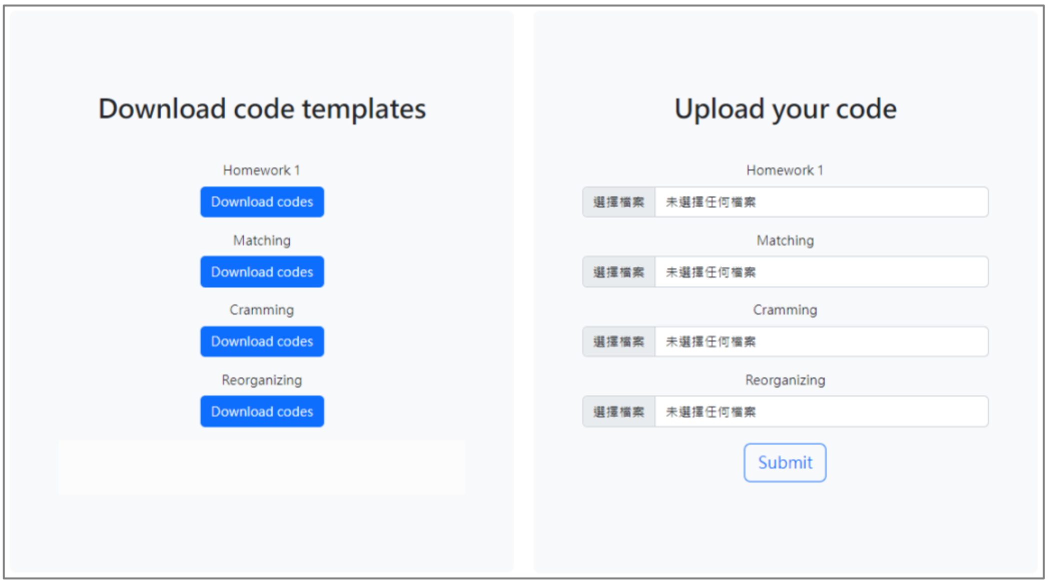


Figure 13 DTF_1-12 and DTF_1-13 on TAaaS_1

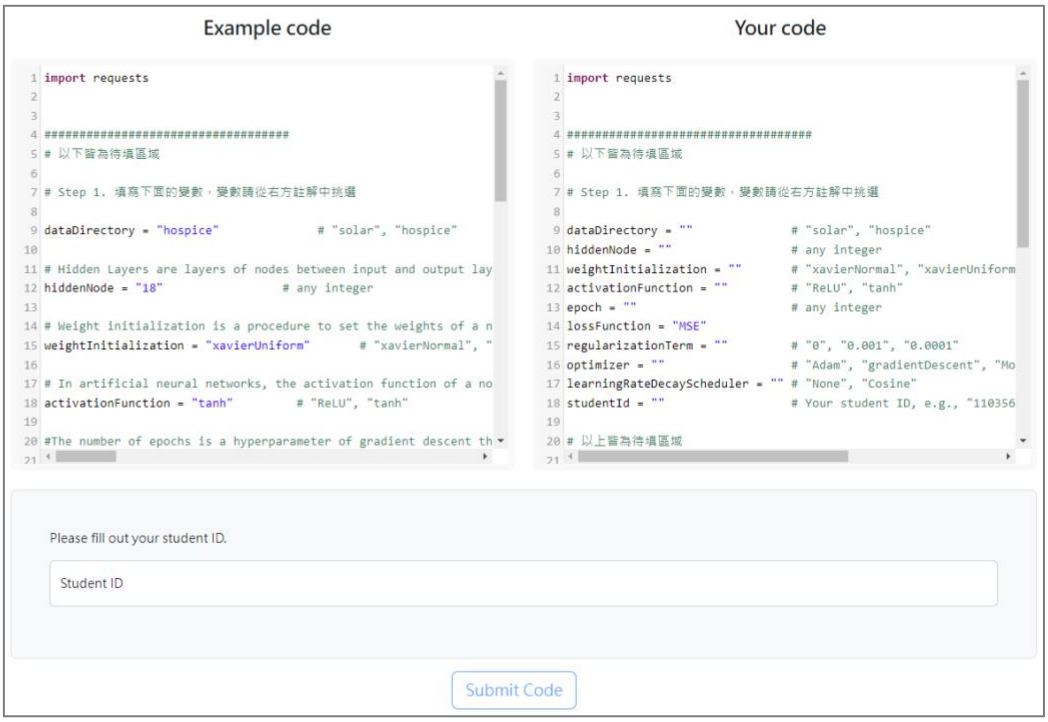


Figure 14 DTF_1-12 and DTF_1-13 on TAaaS_2

5.5. Insight derived from the counter-comparison of the two versions differences

During the initial DTP_1, we conducted the first cycle on TAaaS_1, collecting crucial feedback on user experience and interface issues. Our users point out areas of difficulty, express their needs, and suggest improvements. For example, one piece of feedback (DTF_1–1) suggests the system should retain the user's previous inputs. Responding to this, we modify the web form fields in TAaaS_2 to remember the user's previous data. This modification aims to enhance the user experience by reducing redundant effort and making the process smoother for repeated use. Feedback DTF_1-4 suggests that users need to view model training hyperparameters easily. Therefore, in the TAaaS_2 system, we added a feature that allowed users to query the record of model training hyperparameters on the Ensemble page. This addition makes the comparison and analysis of different model parameters more straightforward for our users. We also receive feedback regarding the mechanism of code submission (DTF_1-12 and DTF_1-13). Originally, users submitted code by downloading and uploading files. Responding to user preferences for a more direct method, we designed the required information and code submission process to be completed directly on the web page. To facilitate this, we incorporate a web compiler into TAaaS_2. However, we also note the lack of debugging functionality in the current setup, which we will consider for future iterations. Another valuable feedback (DTF_1–8) is the need to download the model's weight storage file. In response, we added this functionality in TAaaS_2, allowing users to download the PyTorch file corresponding to the model's weights.

After implementing these design modifications based on the first round of feedback, we run a second cycle: DTP_2 with TAaaS_2. This process allows us to evaluate the effectiveness of our changes and gather more feedback for further improvements. From our experiment, we can know that the TAaaS_2 system, designed based on user feedback from DTF_1, had an impact on the nature of the feedback received during the second DTP_2.

1. Enhanced user satisfaction: Modifications such as retaining user's previous inputs, allowing easy access to hyperparameters, and the provision of a web compiler enhanced the overall user

experience in TAaaS_2. As a result, the feedback in DTF_2 was more focused on improving specific features and streamlining the user interface rather than addressing broader usability issues. This shift demonstrates that the system's usability improved significantly, leading to a smoother and more intuitive user experience.

2. Focused feedback: In the DTF_2, users had fewer fundamental issues with understanding the system or performing tasks. Instead, they suggested enhancements for specific functionalities. For instance, the feedback suggested that the system should provide a button to return to the initial stage of the training process (DTF_2-1), include a section that leads directly to the Ensemble feature (DTF_2-2 and DTF_2-3), and add loss charts to the model links so users can view the training results (DTF_2-4). This shift in feedback represents a more mature stage of system development, where users are more focused on fine-tunings rather than core functionalities.
3. Developer service improvement: Feedback from DTF_2 also suggested that while the overall user experience had improved, there was still room for improvement in the developer section of the service, specifically in making it more user-proof (DTF_2-7). This indicates that while the overall user interface and experience had improved, some areas, like the developer service, needed further attention and improvement.
4. Additional feature suggestions: With the core usability issues largely addressed in TAaaS_2, users had the space to suggest additional features like visualizing loss comparisons and offering back buttons for easier navigation. These suggestions point towards a desire for more sophisticated features that can further enhance the system's utility.

Throughout this study, we utilize an iterative design thinking process to significantly improve the TAaaS system. Starting with TAaaS_1, we obtain vital user feedback (DTF_1) that highlights key areas of improvement for the system's user interface and experience. This feedback drives the design and functionality of TAaaS_2, leading to an enhanced, more user-friendly system that is more closely aligned with user needs.

6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

In conclusion, the implementation and continued fine-tuning of the TAaaS in accordance with the principles of design thinking, has effectively addressed the needs of its users, especially the students in an advanced programming course. This user-centric approach, with its stages of empathizing, defining, ideating, prototyping, and testing, has been fundamental in ensuring that the system is tailored to address user needs, thereby enhancing their overall learning experience. The valuable feedback we gather from users play a vital role in driving enhancements to TAaaS, resulting in a more intuitive and user-friendly system. The features of TAaaS, including the online editor and the ability to remember and review previous training hyperparameters, are greatly appreciated by the users. These features not only streamline the model training process but also provide students with an opportunity to reflect on their strategies and make informed adjustments for future iterations. Moreover, TAaaS has proven to be a practical and convenient tool in training models and understanding hyperparameters. The system's clear instructions and user-friendly interface not only expedite homework completion but also promote a deeper understanding of course material.

In summary, the implementation of design thinking principles in the development of the TAaaS system has shown significant benefits in enhancing the users' learning journey. The iterative, user-centric design process has led to a highly effective, intuitive educational tool, emphasizing the importance of a user-centered design approach in the creation of AI systems.

6.2. Limitation and Future work

The research framework and the TAaaS system have both demonstrated significant potential in their respective domains, despite a few notable limitations. These limitations, coupled with promising directions for future work, serve as a roadmap for the continued fine-tuning and application of both the research framework and the TAaaS system.

- Limitations:
  1. Representativeness, control of variables, and temporal lag influence: In the research framework and the TAaaS system, potential issues with sample size and diversity were identified, which could impact the accuracy and validity of the results. Additionally, controlling variables such as changes in user situations or external environmental shifts posed challenges. A significant consideration is the temporal lag in the system improvement process and feedback collection, which could affect the precision of the outcomes.
  2. System scalability and ease-of-use: In the case of the TAaaS system, scalability is not extensively tested. As the user base and computational demands increase, this might become an issue. Furthermore, the learning curve and initial learning process with the system are not adequately considered.

- Future Work:
  1. Diversity of Feedback and Users: An important direction for future research is to expand the range of user feedback and diversify the user groups involved. This could entail collecting a variety of user feedback for the research framework, as well as engaging a more diverse range of users, including novice programmers or professionals in different fields, in the case of the TAaaS system.
  2. Use of Advanced Technologies: Leveraging advanced techniques like Natural Language Processing (NLP) or Generative Pretrained Transformer (GPT) could

significantly enhance the scientific and efficient analysis of collected data within the research framework. Similarly, for the TAaaS system, incorporating debugging functionality and adaptive learning features that employ AI can yield a more personalized user experience.

3. System Improvements: For the TAaaS system, future research should consider improving scalability and the user interface, especially the Ensemble function, to increase system efficiency and user-friendliness.

7. REFERENCES

1. Barlas, P., Kyriakou, K., Guest, O., Kleanthous, S., and Otterbacher, J. (2021). To" see" is to stereotype: Image tagging algorithms, gender recognition, and the accuracy-fairness trade-off. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–31.
2. Boag, S., Dube, P., El Maghraoui, K., Herta, B., Hummer, W., Jayaram, K., Khalaf, R., Muthusamy, V., Kalantar, M., and Verma, A. (2018). Dependability in a multi-tenant multi-framework deep learning as-a-service platform. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 43–46. IEEE.
3. Brown, T. et al. (2008). Design thinking. *Harvard business review*, 86(6):84.
4. Brown, T. and Katz, B. (2011). Change by design. *Journal of product innovation management*, 28(3):381–383.
5. Brown, T. and Martin, R. (2015). Design for action. *Harvard Business Review*, 93(9):57–64.
6. Chen, L., Chen, P., and Lin, Z. (2020a). Artificial intelligence in education: A review.
7. Ieee Access, 8:75264–75278.
8. Chen, X., Xie, H., and Hwang, G.-J. (2020b). A multi-perspective study on artificial
9. intelligence in education: Grants, conferences, journals, software tools, institutions, and researchers. Computers and Education: Artificial Intelligence, 1:100005.
10. Clark, K., Smith, R., et al. (2008). Unleashing the power of design thinking. *Design Management Review*, 19(3):8–15.
11. Dhillon, S. K., Ganggayah, M. D., Sinnadurai, S., Lio, P., and Taib, N. A. (2022). Theory and practice of integrating machine learning and conventional statistics in medical data analysis. *Diagnostics*, 12(10):2526.
12. Ebert, C., Gallardo, G., Hernantes, J., and Serrano, N. (2016). Devops. *Ieee Software*, 33(3):94–100.
13. Elshawi, R., Sakr, S., Talia, D., and Trunfio, P. (2018). Big data systems meet machine learning challenges: towards big data science as a service. *Big data research*, 14:1–11.

14. Gasson, S. (2003). Human-centered vs. user-centered approaches to information system design. *Journal of Information Technology Theory and Application (JITTA)*, 5(2):5.
15. Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
16. Google Cloud Architecture Center (2020). Mlops: Continuous delivery and automation pipelines in machine learning.
17. Hasso Plattner Institute of Design at Stanford University (2023). Tools for taking action.
18. Hwang, G.-J., Xie, H., Wah, B. W., and Gašević, D. (2020). Vision, challenges, roles and research issues of artificial intelligence in education.
19. KELLEY, T. A. (2001). *The art of innovation: Lessons in creativity from IDEO, America's leading design firm*, volume 10. Broadway Business.
20. Kreuzberger, D., Kühl, N., and Hirschl, S. (2022). Machine learning operations (mlops): Overview, definition, and architecture. *arXiv preprint arXiv:2205.02302.*
21. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
22. Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of devops concepts and challenges. *ACM Computing Surveys (CSUR)*, 52(6):1–35.
23. Lindberg, T., Meinel, C., and Wagner, R. (2011). Design thinking: A fruitful concept for it development? In *Design thinking*, pages 3–18. Springer.
24. Lins, S., Pandl, K. D., Teigeler, H., Thiebes, S., Bayer, C., and Sunyaev, A. (2021). Artificial intelligence as a service. *Business & Information Systems Engineering*, 63(4):441–456.
25. Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics*, 6(1):1–29.
26. Lucena, P., Braz, A., Chicoria, A., and Tizzei, L. (2017). Ibm design thinking software development framework. In *Brazilian workshop on agile methods*, pages 98–109. Springer.
27. Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.

28. Nagarhalli, T. P., Vaze, V., and Rana, N. (2021). Impact of machine learning in natural language processing: A review. In *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, pages 1529–1534. IEEE.
29. Norman, D. (2017). Design, business models, and human-technology teamwork: As automation and artificial intelligence technologies develop, we need to think less about human-machine interfaces and more about human-machine teamwork. *Research Technology Management*, 60(1):26–30.
30. Pereira, J. C. and de F.S.M. Russo, R. (2018). Design thinking integrated in agile software development: A systematic literature review. *Procedia Computer Science*, 138:775–782. CENTERIS 2018 - International Conference on ENTERprise Information Systems/ ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 -International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018.
31. Rai, A., Constantinides, P., and Sarker, S. (2019). Next generation digital platforms:: Toward human-ai hybrids. Mis Quarterly, 43(1):iii–ix.
32. Riedl, M. O. (2019). Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33–36.
33. Shrestha, A. and Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE access*, 7:53040–53065.
34. Stahl, B. C., Andreou, A., Brey, P., Hatzakis, T., Kirichenko, A., Macnish, K., Shaelou, S. L., Patel, A., Ryan, M., and Wright, D. (2021). Artificial intelligence for human flourishing–beyond principles for machine learning. *Journal of Business Research*, 124:374–388.
35. Stembert, N. and Harbers, M. (2019). Accounting for the human when designing with ai: challenges identified. *CHI'19-Extended Abstracts, Glasgow, Scotland Uk—May 04-09, 2019*.
36. Tsaih, R.-H. (2022). The course materials of new learning algorithm

37. Verganti, R., Vendraminelli, L., and Iansiti, M. (2020). Innovation and design in the age of artificial intelligence. *Journal of Product Innovation Management*, 37(3):212–227.

38. Vetterli, C., Uebernickel, F., Brenner, W., Petrie, C., and Stermann, D. (2016). How deutsche bank’s it division used design thinking to achieve customer proximity. 15:37–53.

39. Xu, W. (2019). Toward human-centered ai: a perspective from human-computer interaction. *interactions*, 26(4):42–46.

40. Xu, W., Dainoff, M. J., Ge, L., and Gao, Z. (2021). From human-computer interaction to human-ai interaction: new challenges and opportunities for enabling human-centered ai. *arXiv preprint arXiv:2105.05424*, 5.

41. Zhang, C. and Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23:100224.

8. APPENDIX

Table 8 Experiment tasks in DTP_1 (Chinese)

- 任務一 - AI Software Service phase
  - 使用個人筆電進入 TAaaS 入口網站，連結: http://140.119.19.87/entry
  - 依照網頁路徑進入 Hw1 的頁面，AI Software Service -> Model pipeline -> Homework #1。
  - 根據頁面上的文字說明，依序填入模型訓練所需的超參數，並訓練出一個模型。注意不必進入 Data preparation 上傳資料集，可以使用系統預設的資料集 (e.g., hospice, solar)。
  - 查看 Model Performance, Training configuration。
- 任務二 - AI Developer Service phase
  - 請用個人筆電進入 TAaaS 入口網站，連結: http://140.119.19.87/entry
  - 進入 AI Developer Service 頁面
  - 下載 homework 1 的範例程式
  - 依照 hw1.py 程式內指示完成步驟
  - 填入所需的欄位
  - 將 hw1.py 壓縮
  - 將壓縮檔命名為 hw1-學號.zip，e.g., hw1-110356021.zip
  - 回到 AI Developer Service 頁面，上傳壓縮檔
  - 進入 Ensemble 頁面，用自己的學號查詢訓練好的模型
- 任務三 - Ensemble phase
  - 請先在 AI Software Service/AI Developer Service 訓練至少 3 個模型，並且在資料集設定使用相同資料集。
  - 進到 Ensemble 頁面，連結: http://140.119.19.87/pipeline/model/hw1/ensemble
  - 用自己的學號查詢訓練好的模型
  - 透過表格的 sorting 功能，查看 Training loss 或 Validating loss 在不同排序下的模型。
  - 選擇前三個訓練較佳的模型，並點擊按鈕送出
  - 查看 Ensemble model 的 validating loss

Table 9 Discussion topics in DTP_1 (Chinese)

1	你認為這個服務頁面提供的資訊是否能引導您完成任務要求的項目嗎? 為甚麼?
2	您認為此服務頁面是否在引導您完成任務要求的項目方面提供了足夠的指導和協助

	？您是否需要額外的幫助或指導才能完成任務？
3	自己寫作業，與透過系統協助完成任務的差異?
4	在自己完成作業時，您會考慮使用哪些系統協助工具？這些工具是如何幫助您更有效地完成任務的？
5	您認為整體使用流程是否順利? 有沒有遇到任何困難或挫折? 有沒有什麼地方使你感到困惑、讓你想要中止使用?
6	根據您的使用經驗，有沒有什麼元素是可以增加在網站內，以幫助您更順利完成任務的?
7	簡短的說一下今天使用完此系統的感受
8	若先使用系統後，再自己完成作業，是否加深學習歷程?

Table 10 Experiment tasks in DTP_2 (Chinese)

- 任務一 - AI Software Service phase
  - 請用個人筆電進入 TAaaS 入口網站，連結: http://140.119.19.87/entry
  - 請依照網頁路徑進入 Hw1 的頁面，AI Software Service -> Model pipeline -> Homework #1
  - 請依照頁面上的文字說明，依步驟完成模型訓練的操作。實驗請使用系統預設的資料集 (e.g., hospice, solar) 即可
- 任務二 - AI Developer Service phase
  - 請用個人筆電進入 TAaaS 入口網站，連結: http://140.119.19.87/entry
  - 請進入 AI Developer Service -> Online Editor 頁面
  - 請依照頁面上的文字說明，依步驟完成模型訓練的操作
  - 若操作順利完成，將會自動導向至 Ensemble 頁面
- 任務三 - Ensemble phase
  - 請使用相同資料集，在前兩個服務中，訓練出三個以上的模型
  - 進到 Ensemble 頁面，連結: http://140.119.19.87/pipeline/model/hw1/ensemble
  - 用自己的學號查詢訓練好的模型
  - 透過網頁上的功能 (例如: 模型訓練資訊、表格排序)，協助您選擇前三個訓練較佳的模型，並點擊按鈕送出
  - 查看 Ensemble model 的 validating loss
  - 實驗結束，請耐心稍等其他受訪者
  - 您可以先將系統使用體驗筆記在提供的文件上

Table 11 Discussion topics in DTP_2 (Chinese)

1	你認為這個服務頁面提供的資訊是否能引導您完成任務要求的項目嗎? 為甚麼?
2	您認為此服務頁面是否在引導您完成任務要求的項目方面提供了足夠的指導和協助？您是否需要額外的幫助或指導才能完成任務？
3	自己寫作業，與透過系統協助完成任務的差異?
4	您認為整體使用流程是否順利? 有沒有遇到任何困難或挫折? 有沒有什麼地方使你感到困惑、讓你想要中止使用?
5	根據您的使用經驗，有沒有什麼元素是可以增加在網站內，以幫助您更順利完成任務的?
6	TAaaS_2 是否有 TAaaS_1 比具備更好的使用體驗?
7	簡短的說一下今天使用完此系統的感受。
8	透過前測中，收取同學們的回饋之後，再改善這個系統，是否更貼近修課同學的需求?

Table 12 Feedback derived from the DTP_1 (Chinese)

Index	Category	Feedback
DTF_1-1	Function	提供欄位記憶和版面停留位置功能，以提高使用者體驗。
DTF_1-2	UI	提供流程圖，以幫助使用者清晰理解整個流程步驟。
DTF_1-3	UI	結構化區塊，包括資料準備、模型訓練和模型績效，以區分不同部分。
DTF_1-4	UI	查看模型訓練超參數的彈跳視窗功能，以方便使用者查看和比較模型。
DTF_1-5	Function	提供 Forward 和 Backward 知識的外部連結，以幫助使用者深入理解模型運作原理。
DTF_1-6	Function	提供資料描述的功能，以更清楚地說明資料集的特徵和屬性。

DTF_1-7	Function	除了損失函數值，也提供模型輸出值，以提供更全面的模型績效評估。
DTF_1-8	Function	可下載模型檔案和程式碼，以便使用者進一步應用和改進模型。
DTF_1-9	Function	提供不同套件之間的使用對照表，例如: Pytorch、TensorFlow，以方便使用者選擇適合的套件。
DTF_1-10	Function	支援多種套件的使用相容性，以增強系統的開放性和靈活性。
DTF_1-11	Function	Ensemble 的背後機制不明確
DTF_1-12	Function	提供更多的文字敘述(註解)，說明超參數用途，以幫助使用者更好地調整模型設計和參數。
DTF_1-13	Function	提供編譯器的擴充功能，以提高使用者編寫程式碼的效率和便捷性。
DTF_1-14	Function	提供編譯器的和 Debug 功能，以提高使用者編寫程式碼的效率和便捷性。

Table 13 Feedback derived from the DTP_2 (Chinese)

Index	Category	Description
DTF_2-1	UI	有一個可以回到訓練步驟初始階段的按鈕，以方便重新訓練模型。
DTF_2-2	UI	在 Entry 頁面提供一個直接進入 Ensemble 的區塊，對應到三項任務的題目。
DTF_2-3	UI	在 Ensemble 頁面提示使用者點選 1~N 個 model 後提交，並增加直接進入 Ensemble 的區塊。
DTF_2-4	Function	在模型連結中加入損失函數圖，以便使用者查看訓練過程的結果。

DTF_2-5	Function	在 Ensemble 頁面中，透過視覺化的介面展示所有或指定模型的損失函數比較圖。
DTF_2-6	UI	在 Online Editor 和上傳下載程式碼的功能中選擇一個呈現方式，避免混淆。
DTF_2-7	Function	在 Developer 版面中加入超參數的防呆功能，例如空值提交時的警示。