J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

KISTI-ML 플랫폼: 과학기술 데이터를 위한 커뮤니티 기반 AI 모델 개발 도구☆

KISTI-ML Platform: A Community-based Rapid AI Model Development Tool for Scientific Data

이 정 철[1] 안 선 일[1*]
Jeongcheol Lee Sunil Ahn

요 약

최근 서비스로서의 머신러닝(MLaaS) 개념은 데이터 자체를 제외하고 네트워크 서버, 스토리지 또는 데이터 과학자 없이도 생산적인 서비스 모델을 구축할 수 있다는 점에서 기계학습을 다루는 대부분의 산업 분야와 연구 그룹들의 많은 관심을 받고 있다. 그러나 과학 분야에서는 양질의 빅데이터를 확보하는 가정 자체가 커다란 도전이 된다. 즉, 연구자 간 연구 결과물의 공유가 쉽지 않을 뿐 아니라 과학기술 데이터의 비정형성 문제를 해결해야하는 문제가 선행된다. 본 논문에서 제안된 KISTI-ML 플랫폼은 과학기술 데이터를 위한 AI 모델 고속 개발 도구로서, 머신러닝에 익숙하지 않은 연구자들을 위해 웹 기반 GUI 인터페이스를 제공하고 연구자는 자신의 데이터를 이용하여 머신러닝 코드를 손쉽게 생성하고 구동할 수 있다. 또한 승인된 커뮤니티 멤버들을 중심으로 데이터셋 및 특징 추출에 사용되는 데이터전처리, 학습 네트워크 설계 등이 포함되는 프로그래밍 코드를 공유할 수 있는 환경을 제공한다.

☞ 주제어 : 기계학습, 빅데이터, MLaaS, 플랫폼, 과학기술 데이터

ABSTRACT

Machine learning as a service, the so-called MLaaS, has recently attracted much attention in almost all industries and research groups. The main reason for this is that you do not need network servers, storage, or even data scientists, except for the data itself, to build a productive service model. However, machine learning is often very difficult for most developers, especially in traditional science due to the lack of well-structured big data for scientific data. For experiment or application researchers, the results of an experiment are rarely shared with other researchers, so creating big data in specific research areas is also a big challenge. In this paper, we introduce the KISTI-ML platform, a community-based rapid AI model development for scientific data. It is a place where machine learning beginners use their own data to automatically generate code by providing a user-friendly online development environment. Users can share datasets and their Jupyter interactive notebooks among authorized community members, including know-how such as data preprocessing to extract features, hidden network design, and other engineering techniques.

☞ keyword : Machine Learning, Big Data, MLaaS, Platform, Scientific Data

1. Introduction

Artificial intelligence (AI) and machine learning have received much attention since the Alpha-Go shock. It is closely linked to the advent of big data and its innovative growth in almost all areas referred to as the Fourth Industrial Revolution (4IR). Several global IT companies and research groups offer AI and machine learning technologies such as Tensorflow [1], Scikit-learn [2] and MXNet [3] as framework services. It can relieve the user of the need to develop a lot of staff and time-consuming machine learning techniques. That is, the user can easily develop a model and a service by creating only development environments according to his wishes. This aspect of technology development contributes significantly to the spread of machine learning and services.

Recently, machine learning as a service [4-8], the

1 Center for Computational Science Platform, Korea Institute of Science and Technology Information (KISTI), Daejeon, 34141, Korea.
* Corresponding author (siahn@kisti.re.kr)
[Received 20 June 2019, Reviewed 25 September 2019, Accepted 12 December 2019]
☆ A Preliminary version of this paper was presented at the 10th International Conference on Internet (ICONI 2018) and was selected as an outstanding paper.

so-called MLaaS, has attracted much attention due to the spread and development of cloud technology. The MLaaS goes beyond using free open ML frameworks so that users can access a web server to easily develop ML models without incurring any cost, such as: Since nothing else than user data is needed, it can be disseminated and applied in almost all industries. For example, with AutoML [4] from Google, you can get a precise model after uploading some image datasets, and Amazon's SageMaker [5] will give you an automated modeler and parameter optimization tools. Such automated ML services will be gradually expanded.

However, there are many more challenges to achieving innovative growth through machine learning techniques. Since the performance of ML results is highly dependent on the quality and amount of data, an effective data collection and management strategy is critical. One of the most successful services of the ML model is, for example, the object recognition and classification model. Food can be detected in photos and even calories can be calculated, and facial recognition identification has also become a reality, but anyone can create such a powerful model. It is likely to be created by companies or research institutes that have collected large quantities of high-quality data. In reality, breaking the walls of global IT conglomerates is not easy.

In contrast to the collection data that can be collected in everyday life, scientific data should be treated in a different way. The collection and processing of scientific data require a high level of knowledge and know-how for the scope beyond a simple survey. The exchange of computational or experimental data between researchers is also very difficult, as it is very rare, making the generation of large amounts of data in a particular area of research a major challenge. Also, previous solutions can not be directly transferred to other research areas, although the number of specific systems [18-20] based on AI / ML techniques is increasing and some of them are very close to each other. Due to the functional differences between target applications and datasets, ML models and their services have to be developed individually. That is, machine learning often feels harder for most developers, especially in traditional science. In this situation, the intuitive solution can be selected from the following three. First, a collaborative system of data scientists and field scientists will be established. This strategy is ideal, but it costs a lot of money and can take a long time to get a useful result. Second, data scientists are promoted as field scientists. This strategy is very difficult to achieve. Finally, we will promote field researchers as data analysts. As mentioned above, this is the most realistic option in the current situation as the development of machine learning technology moves towards better accessibility.

The online machine learning platforms can be broadly divided into two types: a cloud platform for specialized analysis and a competitive platform for the exchange of know-how. First, cloud platforms such as Amazon AWS, Google's Cloud AI, and Microsoft Azure ML Studio [7] provide GUI-based development environments, computing, services and distribution, and programming tools. Although these platforms provide multiple machine learning algorithms and efficient data analysis tools, it is difficult to learn the direct application examples of field science data because creating a well-structured high-quality machine learning dataset is also another big challenge in science. In most cases, scientific data is raw data that can not be directly applied to machine learning modeling. This data should be preprocessed, tagged, and extracted into features such as fingerprints or specific components. Next, Kaggle [8] and OpenML [9] are competitive platforms that can share both ML algorithm instructions and data pre- and post-processing methods. For a skilled scientist who can handle machine learning algorithms to some degree, this is the smartest way to mimic and learn machine learning cases in his discipline using such a platform. Namely, these platforms require expertise in data analysis and programming techniques for machine learning.

In this paper, we introduce a novel machine learning platform entitled KISTI-ML Platform. As a MLaaS platform, the user can use the KISTI-ML platform web portal without computational resources such as CPU / GPU, resource distribution, and ML library management. The KISTI-ML platform is not only a cloud platform but also a competitive platform. It provides a user-friendly GUI interface for developing machine learning models as well as an automatically generated program code that beginners want to learn about. It's very easy and efficient to create a simple AI model. Users can easily modify and add the generated code to improve their model if their data set has changed based

on their scientific knowledge rather than replacing all the programming code. In addition, users can upload their own data, and such raw data can be preprocessed automatically by pre-processing a docker image created by a community manager. Users can select a kernel, a task, an algorithm and its parameters, as well as various statistical and analytical data tools. After that, users can run automatically generated code in their own development environment on the web. That is, users can share datasets and the Jupyter interactive notebook, including data preprocessing, hidden network design, or some tricks among authorized community members.

The rest of this paper is organized as follows. The Section 2 explains the related work and the Section 3 shows details of the KISTI-ML platform including a system architecture. An example scenario how to use the KISTI-ML platform is shown in the Section 4. Finally, the Section 5 concludes this paper.

2. Related Work

In this section, we introduce several platforms and their functional differences to understand recent trends regarding machine learning platforms. Until now, the well-known machine learning platform can be divided into two categories: a cloud platform for expert analysis and a competition platform for know-how sharing. First, we explain cloud platforms in detail.

Amazon AWS is the largest provider of cloud-based services and provides a platform for machine learning related services such as Amazon ML and SageMaker [5]. Amazon ML provides automated solutions for data classification, forecasting, and clustering. The functional characteristics are summarized as follows. First, it is possible to load data (csv) from multiple sources, including Amazon RDS (relational database) and Amazon Redshift (data warehouse). Second, it provides automated machine learning data preprocessing. Third, it provides prediction models based on supervised learning such as binary classification, multiclass classification, and regression. Fourth, since the selection of the algorithm is also automatic, the user's role is simplified by data upload and selection. Recently Amazon provides the Amazon SageMaker platform for developing and distributing specialized machine learning models. The functional features of SageMaker are as follows. First, it provides automatic model tuning and an interactive development environment, Jupyter notebook instance, for free data distribution and analysis. Second, it provides frameworks such as Tensorflow, Apache MXNet, Pytorch, and Chainer, as well as other frameworks using Docker. Third, it provides a built-in algorithm optimized for large data sets such as Linear learner, XGBoost, Image classification, and Seq2seq. That is, the Amazon Sagemaker platform is in the process of absorbing the capabilities of their existing ML platform.

Microsoft Azure [7] has received much attention in recent cloud storage and business environments. Azure ML Studio provides a GUI-based, easy-to-use interface for developers to create, test, and deploy machine learning algorithms. The functional features are as follows. First, it requires the skill of MS Azure interface and workflow but has the advantage of visualizing and analyzing each step in the workflow. Second, it provides data retrieval, preprocessing, algorithm selection, and model performance verification. Third, Binary classification, multiclass classification, anomaly detection, and text analysis. Fourth, the resultant model can be reshared to the API through the REST interface. Starting in September 2017, they are providing a new service called MS Azure ML Service. Unlike the ML Studio, it is easy to be confused with existing names, but it provides a new platform for users to do their own model engineering without providing a built-in approach. Currently, it is not interoperable with existing services. It supports several libraries such as Tensorflow and scikit-learn as well as other frameworks by using the Docker.

Google has provided artificial intelligence services at two levels: a sophisticated machine learning engine for data scientists and a highly automated Google Prediction API, but since April 2018, Google has discontinued support for the Google Prediction API. The Google Prediction API used the user's data to grow the model, predict the classification and regression, and service the development model using the REST API interface. AutoML [4] has been beta testing since November 2018 through Alpha testing. People who do not have the expertise in data science using AutoML have the advantage of learning high-quality custom data models in an automated fashion. The first product was AutoML Vision,

which allows customized image recognition models. In the long term, services will be available for all other major AI areas. Recently, they unveiled the AutoML Tables for structured user datasets. The interesting aspect is that the AutoML Tables implementation regularly performs well in benchmark tests against Kaggle competitions, demonstrating state-of-the-art performance across the industry. The Google Cloud Machine Learning Engine is also a very flexible service for experienced data scientists, unlike the Prediction API. It is similar to Amazon SageMaker and MS Azure ML Service in that it is a specialized service for advanced users. Functionally, it supports machine learning and deep learning frameworks such as Tensorflow, scikit-learn, XGBoost, and Keras.

IBM's Watson Machine Learning provides Watson ML service [16], a machine learning service running on the IBM cloud, and Watson Studio service, a collaborative development environment that supports both GUI interface and code-level editable notebooks. Watson ML Service provides various model learning and distribution interfaces such as the Python client library, Command Line Interface, and REST API. Recently, they started to provide HPO optimization library, which is an automated tuning technique for high-quality model learning. It supports various machine learning frameworks such as Tensorflow, Spark MLib, PyTorch, IBM SPSS, and PMML. Watson Studio provides automated model learning services using a GUI interface. Built-in algorithms such as Logistic Regression, RF Classifier, and Gradient Boosted Tree Regressor provides an automated in-depth neural network learning workflow using a GUI-based Flow Editor. It also provides programming features such as Jupiter to provide flexible services for advanced users.

These cloud-based machine learning platforms tend to be two-track services that support highly automated model learning services and custom model engineering for advanced users at the same time. Because their final goal is to satisfy diverse demand effectively for market share by providing not only a simple service for automatic data analysis but also various development environment and API for specialized analysis service. Next, we explain the status of the competition platforms.

Kaggle [8] was acquired by Google in 2017 as a big data solutions competition platform company, they have built communities that create and compete models that deals with data and challenges provided by businesses, government agencies, organizations, laboratories, or individuals. That is, it prizes money and data in order to create a best-effort ML model by competitions. Kaggle also has a sharing system that allows feedback, debate, reassessment, and referrals, as well as writing personal analysis reports. Statistics show that the mutual sharing reference system in the community is one of the most effective methods for data science education and training of practitioners. Such opened data analysis techniques can be applied directly to the business, learning methods of applying the machine learning, and/or even can be used for forming a human network among experts groups.

OpenML [9] is an online tool for global scale scientific data analysis, structuring, and sharing, including large-scale problem resolution. It is also an open platform that allows data scientists to share, reconstruct, and discuss machine learning experiments results, data, and algorithms. Basic analysis tools, visualization tools, analysis methods, and analytic results are automatically registered to the OpenML platform. Large-scale collaborative science can solve the task of performing detailed analysis on specific data, and can dynamically allocate tasks such as idea formation, data collection, data mining, and reinterpretation of results. Copyright, public relations, productivity improvement, and educational effect can be obtained as a compensation system.

CloudCV [17] is an open source platform aimed at increasing the reusability of artificial intelligence research by allowing everyone to easily configure, compare and share the latest artificial intelligence algorithms. Since 2013, students and faculty have been working on projects at the Virginia Tech University (now Georgia Tech) and have been registered with Google Summer Code, an open source programming contest sponsored by Google to date, and are being developed on the basis of a large number of students and mentors. To date, CloudCV consists of three main platforms: First, Origami is a platform that provides the AIaaS (AI-as-a-Service) solution that allows users to directly configure their infrastructure and resolve dependencies to transform their deep learning models into online services. Second, Fabrik is an online collaboration platform that enables you to visualize and learn the deep learning model

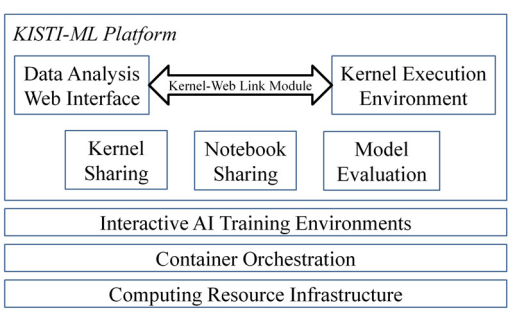
with a simple drag-and-drop approach. Users have been able to import deep-run networks used in popular deep-running frameworks such as Caffe or Keras to debug models through a web GUI interface. Finally, EvalAI is an artificial intelligence model evaluation server that can perform the evaluation of artificial intelligence competitions such as visual question answering and image captioning. CloudCV is designed to be easy for users to access and use, however, it is not easy to build a community because the platform should be preinstalled by community members.

In summary, the cloud platform makes it easy to apply your data to the latest machine learning techniques and to analyze a variety of data, but it is difficult to learn how to apply on-site scientific data directly. On the other hand, competition platforms have the advantage of imitating and learning the machine learning cases in the applications desired by the users, but there is a limit to the expert knowledge on data analysis and programming techniques in order to understand and apply them. Therefore, we propose a novel platform that can easily generate structured data for AI analysis from raw data and build models easily to compete and/or share them with other users without expert knowledge of data analysis. Detailed functions of the proposed platform are described in detail in the next Section.

3. The KISTI-ML Platform

3.1 Architecture

Figure 1 shows an overview of the KISTI-ML platform. The KISTI-ML platform uses the EDISON



(Figure 1) An Overview of the KISTI-ML Platform

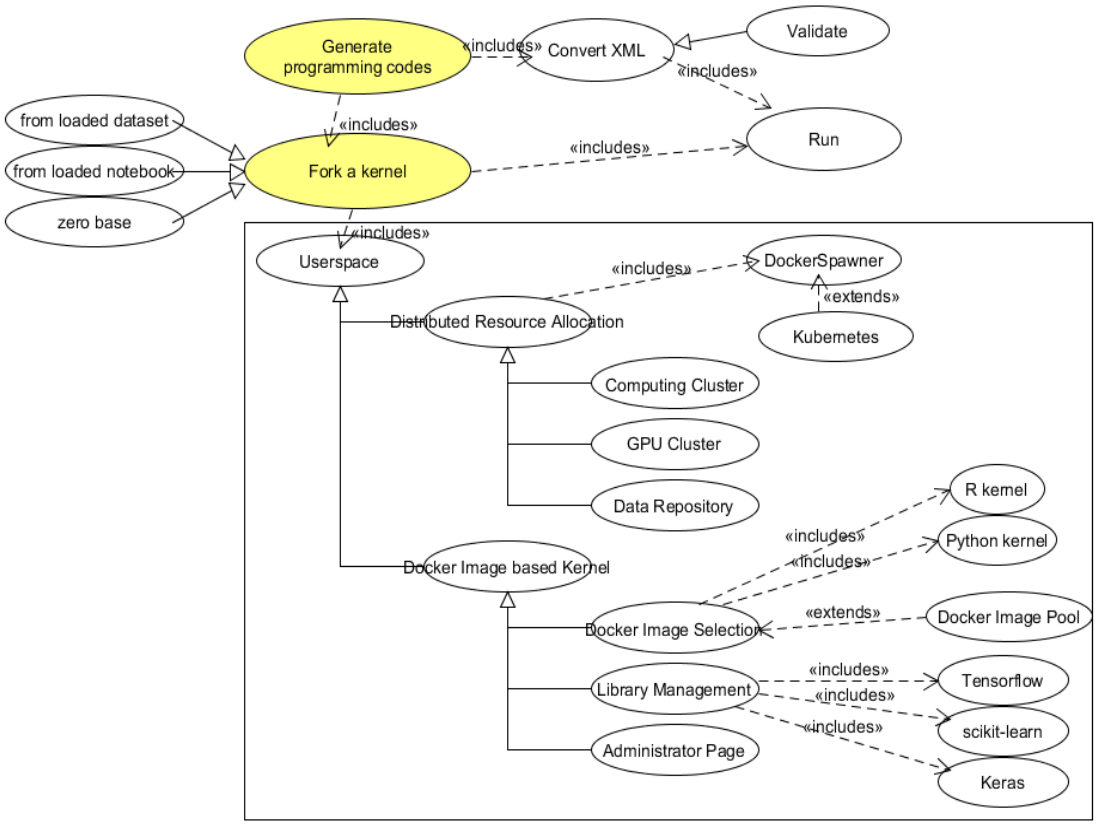
DATA platform technology to effectively analyze and refine data in the EDISON platform [10] as a data repository. Users can apply a "csv" type of refined metadata directly to a machine learning model and use community-developed data preprocessing or data pre-processing images that extract metadata from raw data using a scripting language. The data stored in the data repository can be searched, analyzed and managed on the Web. The system for running the machine learning framework was rebuilt using JupyterHub [11], one of the most popular open-source spam servers on the single-user Jupyter notebook server. So far, Python 3, R and other kernels are supported. The KISTI-ML platform provides a container-based kernel execution environment and supports the distributed resource allocation environment with DockerSwarm [12]. The JupyterHub managed account management provides a single sign-on service (SSO), like the current EDISON platform, and the KISTI-ML platform runs on this infrastructure.

The architecture of the KISTI-ML platform consists of the following five elements:

First, the web-based data analysis interface registers the type of refined metadata "csv" on the platform and converts it into the program code of the format desired by the user. It can manage data sharing level, license, DOI, etc. and automatically generates a summary page that allows easy data analysis when the user uploads data. The GUI interface for selecting the kernel, task, algorithm, parameters for generating the machine learning model is designed as an intuitive interface that takes into account users who are unfamiliar with machine learning.

Second, the kernel execution environment uses a container to set up a distributed resource allocation environment and can work with clusters and data repository. It is being extended to connect GPU clusters and supercomputing resources in the near future. It provides an administrator page for managing libraries and docker images as well as the development environment for each user.

Third, the Web Kernel Link module selects the learning data and imports it into the execution environment of the single kernel or selects the published notebook to import into the execution environment of the single kernel. In this way, users can become familiar with the methods and effects of efficient machine learning in their respective fields. You can



(Figure 2) Usecase Diagram: A Userspace Notebook Management

also discuss comments with experts in the community.

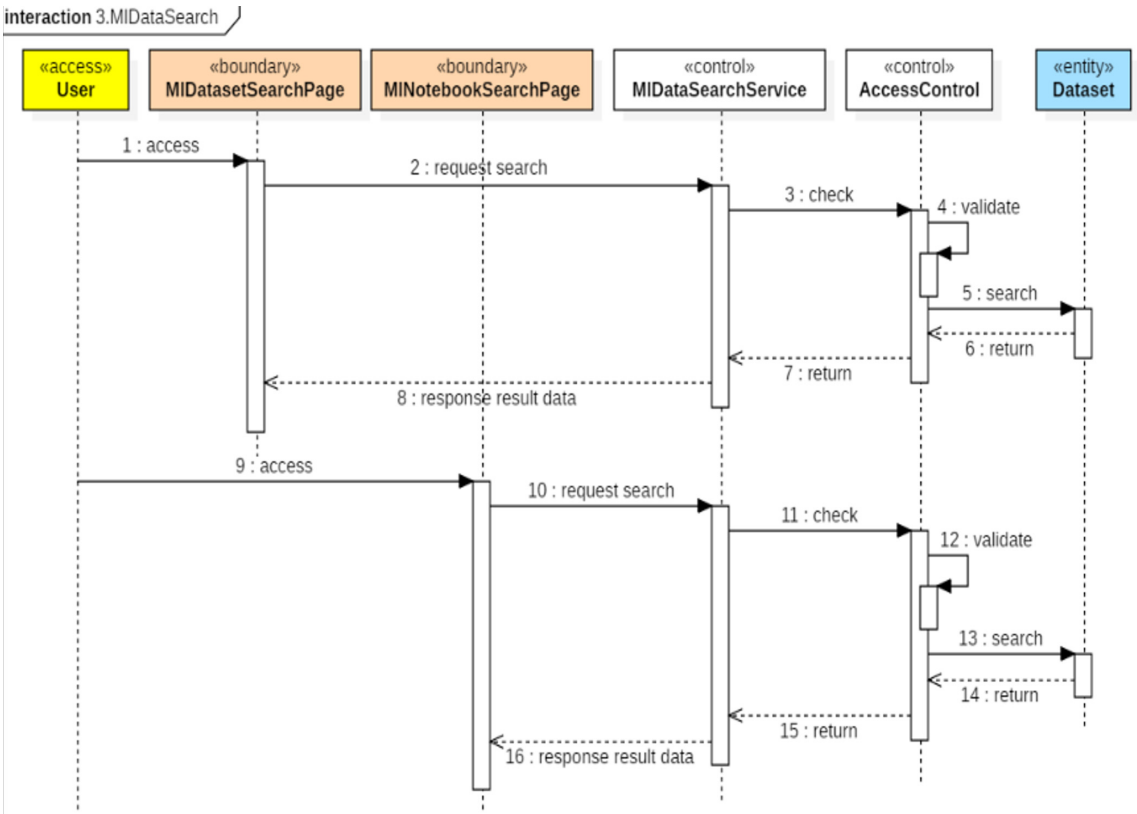
Fourth, using the data and notebook sharing interface, it is possible to check the selected parameter value, docker image used, and kernel information through the GUI interface, and supports various searches such as title, tag, facet, and task. After concluding the discussion on the reward system, the KISTI-ML platform can also be used as a competitive platform such as Kaggle.

Finally, it provides simple tools to evaluate the performance of the development model. Users can use automatically generated code to calculate known performance metrics according to the task. Graphs such as a history score and a correlation matrix can also be drawn only by clicking on the checkbox. Additional features such as the export of graphic files will be added in the near future.

3.2 ML Dataset Upload

This is a comprehensive requirements design for uploading machine learning data. It includes data upload format, user data upload, verification, metadata extraction, and separate storage processing. Machine learning data upload includes all the procedures from when the user accesses the submission format to submit the data, to when the metadata is entered and submitted, and when the submission is finally completed. Data uploads consist of an ML Data Submission that uploads a single dataset to the collection.

As mentioned earlier, the KISTI-ML platform can analyze the structured datasets. Therefore, the user uploads the structured dataset directly or selects a dataset that has completed the curation process in order to perfume the machine learning analysis and modeling. However, most of



(Figure 3) Sequence Diagram for a Data Search

the scientific data are likely to exist in a raw format, so there are additional requirements to support the customized dataset preprocessing on the web. Due to the lack of this paper, we do not handle the docker-based automatic curation system in detail.

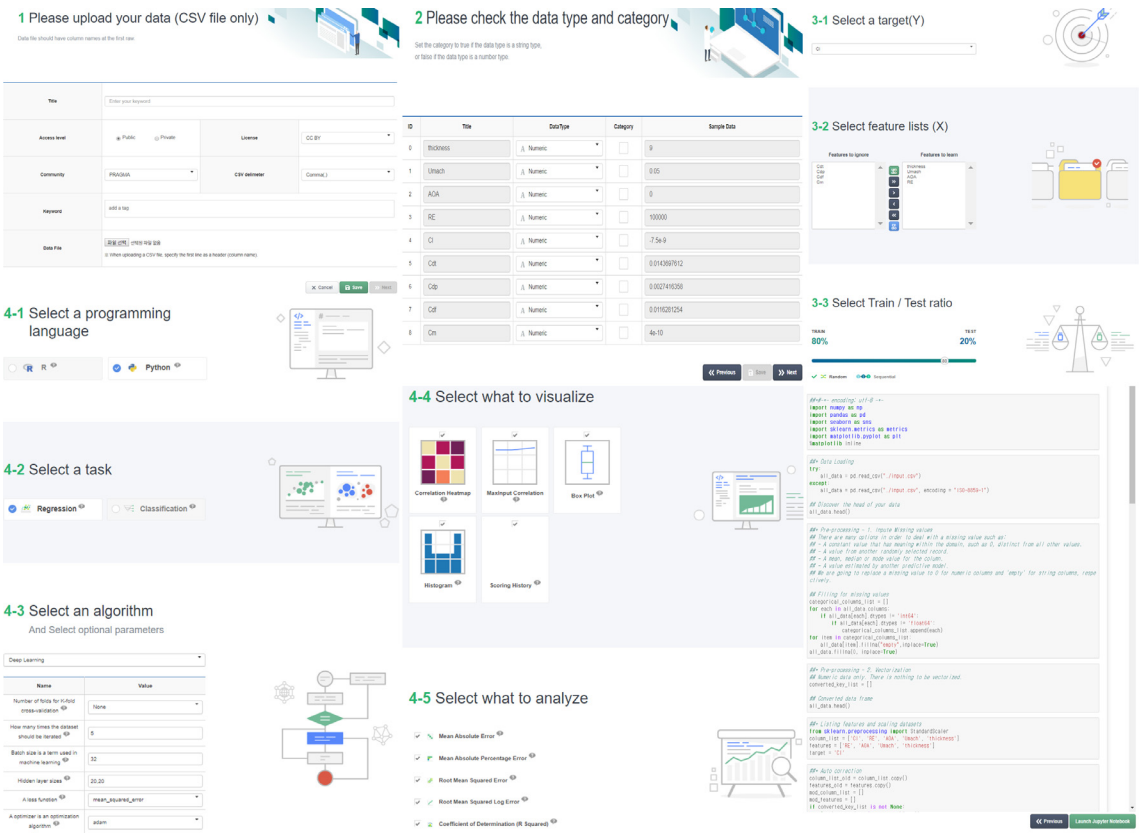
3.3 Management of Userspace and Notebooks

Generated programming code is stored as a well-known notebook format (.ipynb) in order to easily share itself with other researchers. To implement the personalized kernel execution environment and manage the published notebooks, we have developed an environment including view, edit and share functions which can manage these notebooks in a comprehensive manner. The Figure 2 shows the use cases of the management functions in the KISTI-ML platform.

My Notebook Page to viewing, editing and sharing the notebook, executing the code editing window, and completing the notebook sharing. Notebook management of user repository consists of the notebook list view, the notebook edit window, the notebook code editing, the notebook sharing.

If you access My Notebook Page in the user repository, you can search Title, Share, and Create Data information in the form of a table. To run the notebook compilation window, you can click the text corresponding to the title of the desired notebook. The executed notebook can perform comprehensive notebook management such as editing, execution, and debugging.

After the editing is completed, the sharing function can expose other users to search in the search menu. To share, you can share by clicking the Share button in the Share field of the notebook you want to share on My Notebook Page,



(Figure 4) An Example of Machine Learning Modeling by Platform Web Interface

which means that the code is registered as a public data set in your personal kernel environment store. If the state of this button is Shared, it means it is already shared. In the Sharing state, the process of preprocessing in the backend extracts the notebook's metadata and registers it in Dataset It proceeds. When all the process is completed successfully, the text of the button changes to Shared and the sharing is completed. Note that the shared notebook cannot be deleted by the user except a community administrator.

3.4 Search for ML Datasets and Notebooks

The user has the authority on the published dataset or notebooks through the preprocessing and the indexing procedure using the search engine Lucene after uploading the machine learning dataset. It means that search results can be founded by various search methods such as keyword, facet, and tag cloud to facilitate access to provided data through an access control module.

The flow of data retrieval is as follows as shown in the Figure 3. When a user attempts to search a data search page by a keyword, a facet, or a tag cloud method, it is determined whether the user has a search right through the access control logic. If the user has permission, the query conditions are converted into BooleanQuery and then retrieved through the search engine. In addition, the same request information is searched for a facet list grouped by using a dynamic query. The query results are stored and returned in a Map form with page information, facet table of contents, and additional output elements.

The function of ML Data Search was developed to

process using index function of the Lucene search engine. We implemented the paging function using Search Container function of the Liferay framework, and tried to improve the retrieval speed by limiting the number of documents called in conjunction with paging information. In the case of the facet search, the default grouping function of Lucene is ambiguous, so dynamic query is used to generate the list by directly querying the information in the database.

3.5 Automatic Programming Code Generation

The development of the web-based user interface mentioned above can be regarded as an external functional element of the platform. In contrast, a module that converts user input through a GUI interface to a machine learning programming code that can be operated in a user-specific development environment can be determined as an internal functional element of the platform. The automatic programming code generation module can be divided into a process of generating a JSON file from user input and a process of converting the generated JSON file into programming code.

The JSON file consists of the following parameters. The Kernel is a string type that determines the language you want. The Algorithm is the string type, which means the algorithm chosen by the user. The hparams argument contains the parameter options used in the algorithm and the values of those options. The user can call functions that can evaluate the performance of the model through the list contained in the perf_eval. The information specifying the input column and the output column that you want to predict from the dataset is key / value in the form of {"index": "column_name"} in the whole_columns_index_and_name, input_columns_input_and_name, and/or output_columns_ index_and_name. It is stored as a list of values. To automate the vectorization process of the categorical data used in the classification model of machine learning, the data type information of each column is stored as a list in the parameter datatype_of_columns. For separation from the input dataset to the train / test dataset for model learning and verification, the ratios and screening methods are stored in the testing_frame_rate and testing_frame_extract_method parameters, respectively. Finally, ml_file_path and ml_file_ name information including the path and file name of the input dataset are recorded in the JSON file. The following is an example of a generated JSON file.

Thus, the JSON file containing the information of the selected dataset as well as the other options such as a language, a task, an algorithm, its parameters, additional analysis tools, methods for performance evaluation are converted into R or Python languages for machine learning. The basic conversion structure is divided into three parts: prev, body, and post. They are correlated with data preprocessing, algorithm body, and additional analysis and evaluation parts, respectively.

First, the data preprocessing part first specifies the path of the input user data set and loads and parses the path. When there is an empty field in each column thereafter, it is replaced by another value, or the code for vectorizing the category data field, which is not a scalar value, is applied according to the item of the input data set. The function to vectorize the categorical data field is basically provided in R, so it can be easily implemented. In Python, however, the implementation of a function that vectorizes some of the columns extracted from the loaded dataset and replaces it with the input data frame of the machine learning Is required. After the vectorization process, the input dataset is divided into a training train dataset and a test dataset for verification according to a user-specified ratio.

Second, the algorithm body part consists of the split dataset created in the above process and the part that generates the code according to the selected algorithm and the options of the corresponding algorithm. The R code generation supports Multiple Linear Regression, Support Vector Machine, Random Forest, Boosted Tree, Local Regression and Deep Learning algorithms. The Python code generation supports Multiple Linear Regression, Support Vector Machine, Random Forest, Adaboost and Deep Learning algorithm. The code generated automatically from the user GUI input can be edited by the user directly in the kernel execution environment, but users who are not familiar with machine learning can easily create their own models by selecting the basic options that are mainly used.

Finally, the additional analysis and evaluation parts are configured to be selectable variables according to the selected

task and algorithm. For example, in the case of a regression, a model of Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Root Mean Squared Log Error (RMSLE), and Coefficient of Determination (R Squared) can be used for performance evaluation. In other hands, for a classification, Categorization Accuracy and Mean F1 Scores can be used to compare the performances among machine learning models. The evaluation function can be used. In the case of Classification, it is designed to evaluate the performance of a model made using model evaluation functions such as Accuracy (Categorization Accuracy) and MeanF1 (Mean of F1 Scores). Specific relationships can be found in the following table.

4. Case Study: Making a Machine Learning Model

This chapter shows an example scenario for developing a machine learning model using the KISTI-ML platform. The Figure 4 shows web interfaces for making a machine learning model. First, the user uploads a csv file specifying the title, access level, and license level. In this case, we use airfoil datasets, which are generated by CFD simulations and extracted features as a csv format file. Next is the management page for the datatype and the categorical information of the uploaded dataset. You can see some part of the data on the page. After that, select the columns to predict and select the feature lists to be used for learning. Then select the language you want to write. In this example, we selected the Python language. Finally, we select a task, algorithm, hparams, and performance evaluation tools to generate a JSON table for generating a programming code.

The code can be automatically imported into the individual development environment via the generate button. After uploading the dataset and importing the code into your private development environment, you can share the results with other researchers in the community at any time via the share button, consequently, other researchers can search and reapply them.

5. Conclusion

In this paper, we propose a novel machine learning web platform, called KISTI-ML Platform, running on KISTI computing resources. By using community-based data sharing including machine learning datasets, collections, and interactive notebooks, the KISTI-ML platform becomes a place for machine learning beginners to learn field experts' know-how. These users can learn basic programming techniques of machine learning by using user-friendly designed GUI interfaces to automatically generate programming code with users' own data. In the near future, the KISTI-ML platform will be expended to utilize the GPU cluster and the computing resources of the 5th KISTI supercomputer Nurion. A preliminary version of this paper [13] has been introduced with the simple idea and its blueprint. Fortunately, our platform functions is in a beta service at EDISON-AI platform [14] as well as EDISON-PRAGMA [15] which is a specialized platform running under the resources of the Pacific Rim Application and Grid Middleware Assembly community.

Also, in line with the recent trend of increasing interest in the field of artificial intelligence, even for beginners, comfortable and convenient platforms that can help to make their own models by themselves will take a big part of the whole market place. Global IT companies such as Google, IBM, and Amazon have come up with an artificial intelligence-related automation system to meet this demand. However, if we design a flexible system especially for scientific datasets that can be directly applied and edited as in the proposed method, not only can we obtain a model that performs the desired level of prediction on the user side, but also benefit from education. It provides additional benefits such as saving trial and error, providing useful metadata lists in the same community, processing outliers, selecting appropriate algorithms, and forming big data through sharing related datasets and scientific insights. Consequently, we believe that these will lead to solving several large-scale scientific problems.

참고문헌(Reference)

[1] M. Abadi, A. Agarwal et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," Software available from tensorflow.org. 2016.

[2] Pedregosa et al., "Scikit-learn: machine learning in python," The Journal of Machine Learning Research 12, pp. 2825-2830, 2011.

[3] Tianqi Chen et al., "MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems," arXiv e-prints abs/1512.01274, 2015.

[4] Google Cloud AutoML, available from https://cloud.google.com/automl/

[5] Amazon SageMaker, available from https://aws.amazon.com/ko/sagemaker/

[6] Google Cloud AI, available from https://cloud.google.com/

[7] J. Barnes, "Azure machine learning," Microsoft Azure Esstntials. 1st ed, Microsoft.

[8] Kaggle: your home for data science, available from https://kaggle.com/

[9] J. Vanschoren et al., "OpenML: networked science in machine learning," SIGKDD Explorations 15(2), pp. 49-60, 2013.

[10] EDISON: EDucation-research-industry Integration through Simulation On the Net, available from https://www.edison.re.kr

[11] T. Kluyver et al., "Jupyter notebooks: a publishing format for reproducible computational workflows," Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87-90, 2016.

[12] Swarm: a docker-native clustering system, available from https://github.com/docker/swarm/

[13] J. Lee et al., "KISTI-ML Platform: A Practical Guide to Machine Learning through the Automatic Code Generation with Your Own Datasets," KISS the 10th International Conference on Internet (ICONI), Cambodia, Dec. 2018.

[14] EDISON-AI: a specialized Artificial Intelligence and Data Platform, available from https://www.edison.re.kr/web/ai

[15] EDISON-PRAGMA: a specialized EDISON platform at Pacific Rim Application and Grid Middleware Assembly, available from https://www.edison.re.kr/web/pragma

[16] IBM WATSON, avail from https://www.ibm.com/watson

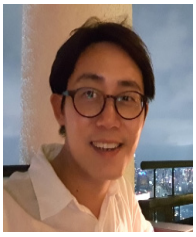
[17] H. Agrawal, C. S. Mathialagan et al., "CloudCV: Large Scale Distributed Computer Vision as a Cloud Service," In Mobile Cloud Visual Media Computing, Springer, pp. 265 – 290, 2015. https://doi.org/10.1007/978-3-319-24702-1_11

[18] H. Lee, "Development of Supervised Machine Learning based Catalog Entry Classification and Recommendation System," Journal of Internet Computing and Services, vol. 20, no. 1, pp. 57-65, 2019. https://doi.org/10.7472/jksii.2019.20.1.57

[19] H. Park, "Personal Credit Evaluation System through Telephone Voice Analysis: By Support Vector Machine," Journal of Internet Computing and Services, vol. 19, no. 6, pp. 63-72, 2018. https://doi.org/10.7472/jksii.2018.19.6.63

[20] G. Shin, D. Kim, S. Hong and M. Han, "The Identification Framework for source code author using Authorship Analysis and CNN," Journal of Internet Computing and Services, vol. 19, no. 5, pp. 33-41, 2018. https://doi.org/10.7472/jksii.2018.19.5.33

◐ 저 자 소 개 ◑



이 정 철(Jeongcheol Lee)
2008년 충남대학교 컴퓨터공학과(공학사) (8.5pt)
2014년 충남대학교 대학원 컴퓨터공학과(공학박사)
2015년~2017년 미국 UCLA 대학원 컴퓨터과학과 박사후연구원
2017년~현재 KISTI 국가슈퍼컴퓨팅본부 선임기술원
관심분야 : 인공지능, 계산과학, 사물인터넷, 무선네트워크
E-mail : jclee@kisti.re.kr

안 선 일(Sunil Ahn)
1999년 서울대학교 전산학과(이학석사)
2010년 서울대학교 대학원 컴퓨터공학과(공학박사)
2004년~2005년 IITA 텔레매틱스 PM실
2005년~현재 KISTI 국가슈퍼컴퓨팅본부 책임연구원
관심분야 : 정보시스템, 분산컴퓨팅, 인공지능
E-mail : siahn@kisti.re.kr