International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

Design and Implementation of a Data Preprocessing Automatic Assessment Module in Jupyter Notebook

HakNeung Go [a], Hyunwoo Moon [b], Youngjun Lee [b], Seong-Won Kim [c,*]

[a] *Songjeong Jungang Elementary School, Gwangsan-Gu, Gwangju, Republic of Korea*
[b] *Department of Computer Education, Korea National University of Education, Cheongju, Republic of Korea*
[c] *Department of Computer Education, Busan National University of Education, Yeonje-gu, Busan, Republic of Korea*
*Corresponding author: *swkim@bnue.ac.kr*

***Abstract*—In data analysis, the preprocessing step is crucial, directly impacting the accuracy and reliability of results. While data preprocessing using a programming language offers capacity, speed, and reproducibility advantages, the complexity of learning programming languages and the scarcity of supportive educational tools pose significant challenges. This study introduces the Data Preprocessing Automatic Assessment (DPAA) module, designed to facilitate learning data preprocessing through programming. The DPAA module, developed using Python and Pandas, utilizes a model answer-based assessment method. It features a self-assessment mechanism that simultaneously displays the outputs of the student's code alongside the model answer, highlighting discrepancies for visual emphasis. Additionally, it includes an automatic evaluation method that compares and evaluates results after transforming them into an array. Furthermore, feedback is provided when the student's answer is incorrect. An example was constructed to validate the DPAA module based on a tutorial from the official Pandas website. The DPAA module, along with examples, was reviewed by informatics teachers at a high school for gifted students and was confirmed for its effectiveness. The DPAA module is expected to support the learning of data preprocessing syntax using Pandas, thereby aiding in the broader application of Python in data analysis. This innovative tool promises to enhance educational outcomes by making the learning process more interactive and supportive, ultimately fostering a deeper understanding of data preprocessing techniques.**

***Keywords*— Data preprocessing; Pandas; Python; program automatic assessment system, data analysis; online judge; Jupiter notebook.**

Manuscript received 11 Nov. 2023; revised 24 Mar. 2024; accepted 12 Sep. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

Data analysis is becoming increasingly important in modern society, serving as an essential tool to support and optimize decision-making processes across various fields. In particular, data analysis is critical in solving real-world complex problems, enhancing efficiency, and discovering new insights [1]–[3]. Data preprocessing is crucial in producing the accuracy and reliability of analysis results and creating an artificial intelligence (AI) model. The data preprocessing includes data normalization, missing value handling, outlier removal, encoding, feature extraction and selection, dimensionality reduction [4], [5].

Data analysis and preprocessing have been emphasized in education and reflected in the informatics curriculum, and data science and AI basics have been newly established [6]. In the informatics curriculum, data analysis is guided by using software or programming languages [6], and data preprocessing is presented as representing data in a form that is easy to analyze. Data science specializes in learning data preprocessing, such as outliers, missing values, and normalization in data preparation and analysis [6].

Using programming languages for data preprocessing offers advantages in handling large volumes of data, flexibility, speed, and reproducibility [7]–[9]. However, mastering these skills requires practice. One method to facilitate learning programming is using a Program Automatic Assessment (PAA) system that provides real-time evaluation and feedback on the code [10]. This approach has been reported to be cognitively and formatively effective and proposed as an informatics curriculum assessment method [11], [12]. Since PAA performs automatic assessments based on letter and numerical data, it isn't easy to evaluate data in images, datasets, and models [13]. In response to these challenges, this study aims to develop a PAA system based on a dataset specifically designed to facilitate practice in data preprocessing tasks.

II. Materials and Method

A. Data Analysis and Preprocessing

Data analysis is a structured process that typically follows several key steps: data collection, data preprocessing, data exploration, model building, and data visualization. Initially, data is collected from various sources, which might include collecting data directly, internal systems, public datasets, or data purchased from external providers. Next step, data preprocessing is performed to increase accuracy and reliability, so removing duplicates or missing values, transforming normalization. Data exploration involves statistical analysis and hypothesis testing to understand the properties and underlying patterns within the data. Then, predictive or descriptive models are built to analyze the data further. Finally, the results are visualized using charts, graphs, and other visual representation tools to make the insights comprehensible and actionable for decision-makers. These steps form the backbone of effective data analysis, enabling businesses and researchers to derive meaningful conclusions that can drive strategic decisions. In addition, data-driven decision-making in learning can lead to the idea of problem solving and solving the problem based on data [14], [15].

Data preprocessing, a critical stage in the data analysis pipeline, involves cleaning and transforming raw data into a format that is suitable for analysis. This step is essential because the quality of data analysis is directly dependent on the quality of the data processed and AI model. Data Preprocessing addresses issues such as missing values, inconsistencies, and noise in the data, which if left unchecked, can lead to inaccurate and misleading analysis outcomes. Moreover, preprocessing techniques such as normalization, transformation, and feature selection help in enhancing the efficiency of the analysis models, making data easier to work with and often improving the accuracy of predictions [16]. In education fill in missing values, identify or remove outliers, convert numerical values to categorical values, aggregation/summarization, attribute/feature construction, integration of multiple databases, data cubes, or files, normalization, dimensionality reduction, sampling are presented as learning contents related to data pre-processing [17].

B. Data Analysis and Data Preprocessing Curriculum in Korea

In the 2015 revision curriculum, the Ministry of Education in Korea reflected the contents of data analysis in the ‘data and information’ section in the informatics curriculum of secondary education [18]. This curriculum covered data collection, management, visualization, and application software utilization. In response to the evolution of critical technologies such as AI and big data in the intelligent information society, the Ministry of Education announced a comprehensive plan for information education in 2020. The plan presented the contents of systematic information education according to the school level and the establishment of elective courses based on informatics, such as AI basics and data science [19]. Following this plan, AI Basic and Data Science courses had been established in September 2020 through a partial revision of the 2015 curriculum. In the 2022 curriculum, data-related content was added in elementary school. In the informatics curriculum in secondary education, the ‘Data and Information’ section was redefined as ‘Data’ and the content expanded. Within the middle school Informatics, the content of data analysis involves data preprocessing and visualization using application software or programming languages [8]. Within the AI Basic, data preprocessing is deal with to facilitate the generation of suitable data for machine learning in the ‘AI and Learning’ section [7]. And, within the Data Science address data preprocessing and analysis techniques, such as handling outliers and missing values, and normalization, in the ‘Data Preparation and Analysis’ section [6]. Table 1 is related to data preprocessing in the 2022 Curriculum.

TABLE I
Data preprocessing in the 2022 curriculum in Korea

Subject	Section	Achievement standards (·) and explanations (-)
Informatic (Middle)	Data	• Data interpretation based on data analysis − To data analysis, representing data in a format suitable for data analysis.
Informatic (High)	Data	• Visualizing big data and interpreting its meaning and value. − To big data analysis, representing data in a format suitable for data analysis.
AI basic	AI and Learning	• Processing the data collected for machine learning to extract key attributes. − Preprocessing the data into a form suitable for problem-solving.
Data Science	Data Preparation and Analysis	• Minimize error possibilities through preprocessing, including outlier detection, missing value handling, and normalization, and visualize for data analysis.

C. Program Automatic Assessment System

The PAA system conducts dynamic assessments functions by inputting input data into code submitted by students and executing the code to obtain output values, which is then compared against correct answer data to determine correctness [20], [21]. This process occurs in real time, allowing for the evaluation of the program code's accuracy. It also provides feedback and helps students learn programming syntax, algorithms, and data structures [22], [23]. The PAA can resolve the delay between submitting assignments and feedback and provide many tasks by reducing the burden of teachers directly scoring many tasks. Studies related to PAA have reported that compared to students who learn by studying examples from textbooks, those who use the PAA system show higher achievement and immersion, indicating a positive impact of this method [10]–[12].

The PAA system utilizes two primary evaluation methods: one based on scoring data and another based on a model answer. The former method processes and evaluates numerical and textual data [22], while the latter assesses submissions based on more complex elements such as images and arrays [13], [24].

1) Assessment based on scoring data: This method utilizes a scoring data set consisting of numbers and texts, namely input data and corresponding answer data. After a student submits their code, this input data is fed into the code, and the code is executed. The outcome of the code is then compared with the answer data to determine correctness. This approach is primarily used for evaluating data structure, programming syntax, algorithms, the advantage of assessing the generality and accuracy of the submitted program through a set of scoring data [22][25].

2) Assessment based on model answer: In this method, the student's submitted code is executed alongside a model answer code, and the results of both codes are compared. This method is a dynamic assessment method and assess the accuracy of the student code because it compares the results obtained by executing the student code and the model answer code. The results obtained from executing both the student's code and the model answer code can be visually displayed, allowing for an intuitive comparison. Moreover, it facilitates more detailed assessment and feedback by utilizing specific methods or properties of libraries. This approach is useful for complex outcomes, such as image or modeling [13], [24].

III. Results and Discussion

A. Design

This study's Data Preprocessing Automatic Assessment (DPAA) module was designed to facilitate learning Panda's syntax in Python. To develop the DPAA module, following an analysis of the literature, requirements and their corresponding functionalities were identified and summarized, as depicted in Table 2.

TABLE II
System Requirements and Features

Requirements	Features
1. Need to learn minimal programming skills for data analysis [26]	1. Use an automatic assessment method to learn programming skills
2. Need feedback after assessments [27]	2. Mark the wrong parts and present the wrong part
3. Need to learn the entire process in data analysis [28]	3. Combined with data visualization module.

DPAA was designed with the following considerations. First, the DPAA uses Jupyter Notebook, executed based on the Programming Automatic Assessment in Jupyter Notebook (PAAinJN) [29]. Second, using Python and Pandas in DPAA is consistent with the actual usage. Third, DPAA conducts assessments based on model answers. Fourth, the results(tables) obtained from executing the students' and model answer codes are displayed in the same row, enabling students to self-assess. Fifth, the results(tables) obtained from executing both the students' and the model answer code are compared to assess the correct answer according to whether the results are the same. Sixth, if it is incorrect, emphasize the wrong part in the output and give feedback on the wrong part at the structure level of the table. Third, DPAA conducts assessments based on model answers. The flowchart of DPAA operates like Fig. 1.

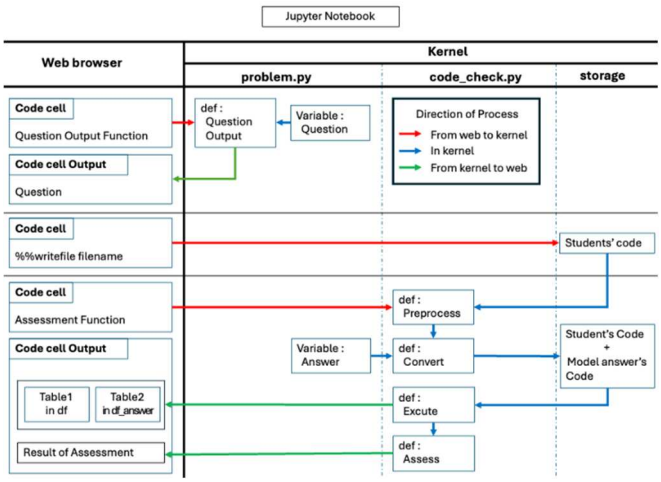

Fig. 1 DPAA module Operating process

B. Composition

The DPAA module consists of the problem module (problem.py), the assessment module (code_check.py), and a folder (CSV file) where external data is stored. Fig. 2 shows each module's configuration, variables, and functions.

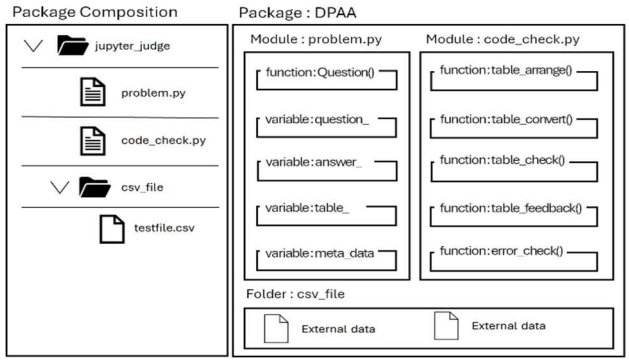

Fig. 2 Composition of DPAA

The problem.py module is structured with several components: variables storing the questions, model answers, and preprocessed results, a function for displaying questions, and metadata linking the file names to be assessed with the model answers. Table 3 describes the functions and variables in problem.py.

TABLE III
Variable and Function in problem.py

Function·variable	Description
question_	- Questions include HTML pages - f-string to output preprocessed results (DataFrame)
table_	- Preprocessed result in code - Convert to HTML and output to question
answer_	- List of model answers saved in string line by line
Question()	- Render and output HTML tags in the question
test_set	- Dictionary to link file names to model answers.

The assessment module comprises the following functions: a function to preprocess files, a function to add model answers to preprocessed code, a function that checks for errors, a function to provide feedback for incorrect, and a function that

integrates all these processes and conducts evaluation. The description of tasks in code_check.py is shown in Table 4.

TABLE IV
FUNCTION IN CODE_CHECK.PY

Function	Description
table_arrange()	− Remove unnecessary whitespace, output command(display(), print()).
table_convert()	− In the preprocessed code. − Add the library required for evaluation. − Add a model answer.
error_check()	− Defined in PAAinJN. − Execute the converted code. − Provide feedback using 'traceback' module and exception handling if error occurs.
table_feedback()	− Provide feedback based on shape, columns, index, value if result is incorrect.
table_check()	− Integrate the entire evaluate process. − Conduct self-assessment (se) and automatic evaluation (aa). − Convert each result into an array (np.array()) and compare each array (np.array_equal()) .

C. Program Automatic Assessment System

To execute the DPAA in PAAinJN, the DPAA module was integrated into PAAinJN and made publicly available on GitHub. Fig. 3 shows the DPAA implemented.

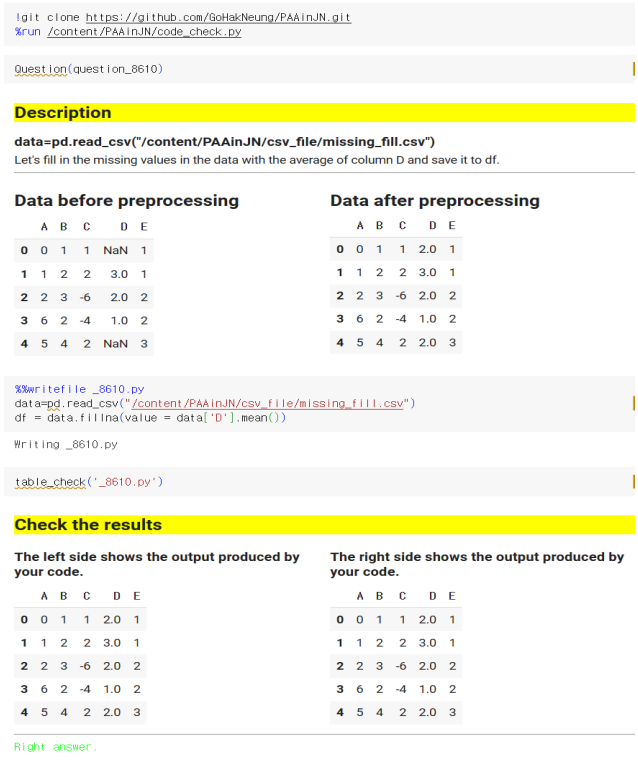

Fig. 3 Implementation of DPAA in Jupyter Notebook

1) Initial setup: The initial setup utilizes the method of PAAinJN methodology. In the Code cell, PAAinJN with DPAA, publicly available on GitHub, is downloaded using cell commands, and the downloaded module is executed using magic commands.

2) Presentation of Question: The question is output to the Code cell Output by using the question variable as a parameter to the question output function (Question()) defined in the problem.py module in the code cell. Questions are presented in text and table. The table output both raw data and preprocessed data. Fig. 4 shows 'table_' variables including preprocessed results and problem variables including HTML tags in f-string format. Fig. 3 shows that question is output from the code cell output using the question output function.

```
table_8610 = pd.read_csv("/content/PAAinJN/csv_file/missing_fill.csv")
table_html_8610 = table_8610.to_html(max_rows = 10, max_cols = 10)
pre_table_8610 = table_8610.fillna(value = table_8610['D'].mean())
pre_table_html_8610 = pre_table_8610.to_html(max_rows = 10, max_cols = 10)

question_8610 = f''' <h2 style = "background-color:yellow; ">Description</h2>
<h3 = "white-space: pre-wrap;">data=pd.read_csv("/content/PAAinJN/csv_file/missing_fill.csv")</h3>
<p>Let's fill in the missing values in the data with the average of column D and save it to df.</p>
<HR>
<div style = "float:left;width:50%">
<h2> Data before preprocessing </h2>
<p>{table_html_8610}<p>
</div>
<div style = "float:right;width:50%">
<h2> Data after preprocessing </h2>
{pre_table_html_8610}
</div>
'''
```

Fig. 4 Question variables for question output

3) Writing code to assess: Writing code to evaluate is conducted in the same method as in the PAAinJN system. The code written in a Code cell is saved to the Kernel as a file using the magic command (`%%writefile filename`). It's important to note that the preprocessed results should be stored in the 'df' variable. In Fig. 3, the code cell starting with '%%writefile _8610.py' is writing the code to evaluate.

4) Code assessment: The result of evaluation is output to the Code cell Output by using the file name stored in kernel as a parameter to the evaluation function('table_check()') defined in the code_check.py module in the Code cell. When the evaluation function('table_check()') is executed, inside the kernel, the evaluation proceeds in the following process.

First, the preprocessing function('table_arrange()') is executed, and the conversion function ('table_convert()') is executed. Fig. 5 shows a file preprocessed and converted from a code file stored in the kernel for evaluation.

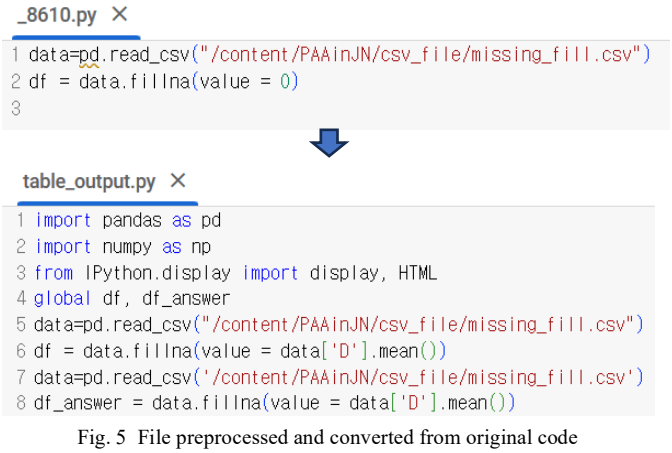

Fig. 5 File preprocessed and converted from original code

Second, executing and error checking function ('error_check()') is executed. When an error occurs, feedback is provided through exception handling. Feedback message on errors is provided by using the traceback module, indicating the location of the error in line number and marking

the wrong part in color. Fig. 6 show that it provides feedback on errors caused by incorrectly writing library methods.

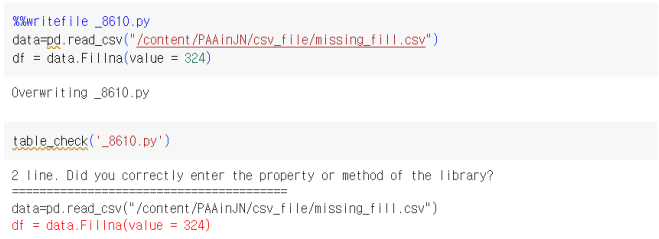

Fig. 6 Feedback for attribute_error

Third, self-assessment and automatic evaluation are conducted if the converted file executes normally. The self-assessment executes the 'df' variable and the 'df_answer' variable and outputs (DataFrame, Value) in the same row. If the shape of the two results tables is the same and values are different, the wrong cell will display the background color yellow. Fig. 3 is the correct case, and Fig. 7 is the wrong case.

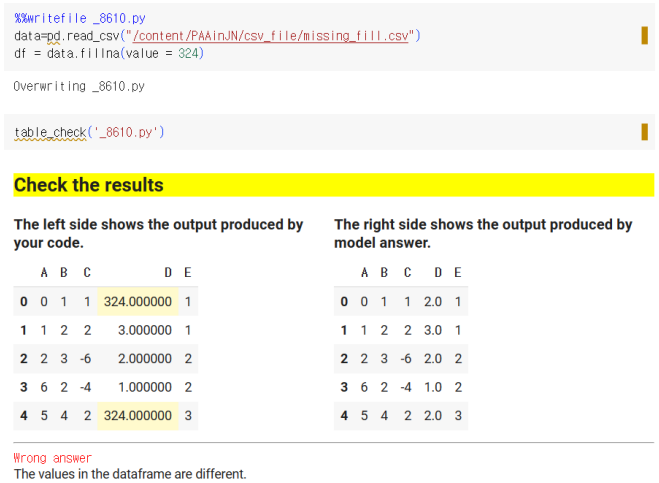

Fig. 7 Case of wrong answer

The automatic evaluation result ('Correct', or 'Incorrect') is output in the next row. In case it is incorrect, the feedback function('table_feedback()') is executed, and feedback is provided based on shape, columns, index, and value.

D. Verification

The validation of the DPAA developed in this study presents challenges due to the absence of comparable systems within the existing literature. So, the Expert review was conducted by creating questions based on tutorial examples provided on the official Pandas website [30]. Creating questions [31] included Dataframe creation and importing external data, exploratory data analysis (EDA), Operation and group by selecting specific columns, Boolean indexing(outlier), handling missing values, operation to normalization, data merging, reshaping (pivot tables, stack). Table 5. shows the questions developed based on tutorials.

Two informatic teachers who work at a high school for gifted were solicited to conduct a review of the DPAA module, utilizing the provided examples as a basis for their evaluation. As a result of the expert review, it was confirmed that the actual use of Python and Pandas and the use of Python and Pandas in the DPAA were consistent [7], [8], [32].

TABLE V
EXAMPLES BASED ON TUTORIALS

Question	Content	Number of questions
Creation	Creation, importing	2
EDA	Head, tail, describe, sort	4
Operation, Groupby	Min, max, mean	3
Selection, Normalization	Selection columns, Scaling, Normalization	4
Outlier data	Boolean index	2
Missing data	Fillna, dropna	2
Merge	Concat, merge	2
Reshaping	Stack, pivot tables	2
Total		21

Also, it was more effective than learning book-type materials in the stage of learning basic grammar related to data preprocessing. Visually presenting the evaluation results and emphasizing the wrong parts could focus students on learning [33], [34]. On the other hand, it was suggested that data analysis would be more effective when connected with data visualization and modeling because it is learned throughout the entire process.

IV. CONCLUSION

Data preprocessing is a crucial step in data analysis, affecting the accuracy and reliability of the results. The revised 2022 curriculum presents data preprocessing as part of data analysis. Data analysis and preprocessing using programming offers advantages such as flexibility, handling large volumes of data, speed, and reproducibility, but learning programming can pose challenges. This study focuses on the data preprocessing phase of data analysis and designs and implements tools that support learning data preprocessing using programming languages. The DPAA module based on PAAinJN has been developed to learn Python and Pandas, which are widely used for data analysis.

The DPAA module comprises a problem module (problem.py), an evaluation module(code_check.py), and an external data folder. The problem module is designed to display problems along with raw data and preprocessed results. The evaluation module defines functions for preprocessing the code to be assessed, code conversion, execution and error verification, assessment, and providing feedback. Evaluation is conducted by self-assessment and automatic evaluation. Self-assessment displayed the student's and model answer's outputs side by side in a single row for visual comparison by the students. The automatic evaluation compares the results from the student and the model answer by converting them into arrays and then comparing their size, columns, index, and values to evaluate correctness and provide feedback on incorrect answers.

To review the DPAA module, examples based on tutorials from the Pandas website were created and reviewed by two teachers who work for the informatics subjects at a high school for gifted, confirming their validity. The developed DPAA module was examined to be useful for basic grammar learning through automatic evaluation and to allow students to focus on learning through self-assessment. The following recommendations are proposed for future research: First, a data pre-processing and data visualization evaluation module

was developed during the data analysis stage. This development should be extended to include an automatic assessment system for data modeling steps in data analysis to ensure the entire data analysis process. Second, developing an educational program using an automatic evaluation module and studying how to use it educationally is necessary.

REFERENCES

[1] I. Ahmed, M. Ahmad, G. Jeon, and F. Piccialli, "A Framework for Pandemic Prediction Using Big Data Analytics," *Big Data Research*, vol. 25, p. 100190, Jul. 2021, doi: 10.1016/j.bdr.2021.100190.

[2] W. Jang and S. Kim, "A review on trends of programming(algorithm) automated assessment system and it's application," *The Journal of Korean Association of Computer Education*, vol. 20, no. 1, pp. 13-26, Jan. 2017.

[3] S. Kim, Y. Jeon and T. Kim, "Research on the Development and Utility Analysis of K-12 Artificial Intelligence Educational Datasets Using Synthetic Datasets Generation Method" *Korean Association of Computer Education*, vol. 25, no. 3, pp. 9-22, 2022.

[4] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Frontiers in Energy Research*, vol. 9, Mar. 2021, doi: 10.3389/fenrg.2021.652801.

[5] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and Software Technology*, vol. 67, pp. 108–127, Nov. 2015, doi: 10.1016/j.infsof.2015.07.004.

[6] Ministry of Education, "Informatics curriculum in 2022 revised curriculum," 2022. [Online]. Available: http://ncic.re.kr. Accessed on: Mar. 1, 2024.

[7] W. Jang and S. Kim, "Differences in self-efficacy between block and textual language in programming education using online judge," The *Journal of Korean Association of Computer Education*, vol. 23, no. 4, pp. 23-33, Jul. 2020.

[8] W. Jang, "The Effects of Online Judge System on Motivation and Thinking in Programming Education : Structural Relationships between Factors," *The Journal of Korean Association of Computer Education*, vol. 24, no. 5, pp. 1-16, Sep. 2021.

[9] L. Y.-H. Lo, Y. Ming, and H. Qu, "Learning Vis Tools: Teaching Data Visualization Tutorials," *2019 IEEE Visualization Conference (VIS)*, pp. 11–15, Oct. 2019, doi: 10.1109/visual.2019.8933751.

[10] Y. Watanobe, Md. M. Rahman, T. Matsumoto, U. K. Rage, and P. Ravikumar, "Online Judge System: Requirements, Architecture, and Experiences," *International Journal of Software Engineering and Knowledge Engineering*, vol. 32, no. 06, pp. 917–946, Jun. 2022, doi:10.1142/s0218194022500346.

[11] D. Bilegjargal and N.-L. Hsueh, "Understanding Students' Acceptance of Online Judge System in Programming Courses: A Structural Equation Modeling Approach," *IEEE Access*, vol. 9, pp. 152606–152615, 2021, doi: 10.1109/access.2021.3126896.

[12] J. Wang, P. Lin, Z. Tang, and S. Chen, "How problem difficulty and order influence programming education outcomes in online judge systems," *Heliyon*, vol. 9, no. 11, p. e20947, Nov. 2023, doi:10.1016/j.heliyon.2023.e20947.

[13] H. Go and Y. Lee, Design and Implementation of a Data Visualization Assessment Module in Jupyter Notebook, *Journal of The Korea Society of Computer and Information*, vol. 28, no. 9, pp. 167-176, Oct. 2023.

[14] M. Islam, "Data Analysis: Types, Process, Methods, Techniques and Tools," *International Journal on Data Science and Technology*, vol. 6, no. 1, p. 10, 2020, doi: 10.11648/j.ijdst.20200601.12.

[15] S. Shin, "A Study on the Instructional Model in Elementary School for Data Science Education using Public Data," *Journal of The Korean Association of Information Education*, vol. 27, no. 1, pp. 57–69, Feb. 2023, doi: 10.14352/jkaie.2023.27.1.57.

[16] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, Nov. 2016, doi: 10.1186/s41044-016-0014-0.

[17] S. Kim, Y. Jeon and T. Kim, "A Study on Data Preprocessing Content Knowledges According to School Level for Artificial Intelligence Education", *The Journal of Korean Association of Computer Education*, vol. 24, no. 4, pp. 1-12, 2021.

[18] Ministry of Education, "Informatics curriculum in 2015 revised curriculum", 2015. [Online]. Available: http://ncic.re.kr. Accessed on: Mar. 1, 2024.

[19] Ministry of Education, "Informatics Education Master Plan", 2020. [Online]. Available: http://moe.go.kr. Accessed on: Mar. 1, 2024.

[20] A. Joo and M. R. Kim, "Effects of Discussion Classes Using Data Visualization Materials on Data Literacy of Elementary School Students," *The Journal of Korean Association of Computer Education*, vol. 27, no. 2, pp. 37–47, Mar. 2024, doi:10.32431/kace.2024.27.2.004.

[21] A. Kurnia, A. Lim, and B. Cheang, "Online Judge," *Computers & Education*, vol. 36, no. 4, pp. 299–315, May 2001, doi: 10.1016/s0360-1315(01)00018-5.

[22] S. Jeong, H. Go, and Y. Lee, "Development and Application of Non-linear Search Questions Using an Online Judge System," *The Journal of Korean Association of Computer Education*, vol. 27, no. 2, pp. 1–11, Mar. 2024, doi: 10.32431/kace.2024.27.2.001.

[23] S. Kim and T. Kim "A Study on Educational Dataset Standards for K-12 Artificial Intelligence Education," *The Journal of Korean Association of Computer Education*. vol. 25, no. 1, pp. 29–40, Jan. 2022.

[24] H. Go, J. H. Jeon, and Y. Lee, "A Study on the Development of Problem Bank for Programming·Math Convergence Education in Programming Automatic Assessment System," *Journal of The Korean Association of Information Education*, vol. 27, no. 2, pp. 141–152, Apr. 2023, doi: 10.14352/jkaie.2023.27.2.141.

[25] W. Y. Chang, "The Effects of Online Judge System on Motivation and Thinking in Programming Education : Structural Relationships between Factors," *The Journal of Korean Association of Computer Education*, vol. 24, no. 5, pp. 1–16, 2021.

[26] J. Seo, "A Case Study on the Teaching and Learning Method of SW Education for Data Analysis Problem Solving," *Journal of Digital Contents Society*, vol. 20, no. 10, pp. 1953–1960, Oct. 2019, doi:10.9728/dcs.2019.20.10.1953.

[27] B. Wisniewski, K. Zierer, and J. Hattie, "The Power of Feedback Revisited: A Meta-Analysis of Educational Feedback Research," *Frontiers in Psychology*, vol. 10, Jan. 2020, doi: 10.3389/fpsyg.2019.03087.

[28] A. Nguyen, L. Gardner, and D. Sheridan, "Data analytics in higher education: An integrated view," *Journal of Information Systems Education*, vol. 31, no. 1, pp. 61-71, 2020.

[29] H. Go, S.-W. Kim, and Y. Lee, "Design and Implementation of a Programming Automatic Assessment System in Jupyter Notebook," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 13, no. 3, pp. 1080–1086, Jun. 2023, doi:10.18517/ijaseit.13.3.18457.

[30] Pandas, Preprocessing library in Python. [Online]. Available : https://pandas.pydata.org. accessed: Jan. 1, 2024.

[31] G. H. Neung, "PAAinJN: Question Example," GitHub repository, https://github.com/GoHakNeung/PAAinJN/blob/main/question_example/example.ipynb, accessed Jan. 1, 2024.

[32] S. Choi, "Designing LLM-based Code Reviewing Learning Environment for Programming Education," *The Journal of Korean Association of Computer Education*, vol. 26, no. 5, pp. 1-11, Sep. 2023.

[33] S. Kim, "Developing Code Generation Prompts for Programming Education with Generative AI," *The Journal of Korean Association of Computer Education*, vol. 26, no. 5, pp. 107-117, Sep. 2023.

[34] Y. S. Cho, "Research on Design for the Assessment System and Knowledge Tracing Methods Based on Generative AI," *The Journal of Korean Association of Computer Education*, vol. 27, no. 1, pp. 143-156, Jan. 2024.