International Journal of Advanced Smart Convergence Vol.9 No.4 179-183 (2020)
http://dx.doi.org/10.7236/IJASC.2020.9.4.179

IJASC 20-4-23

Machine Learning-Based Programming Analysis Model Proposal : Based on User Behavioral Analysis

Seonghoon Jang[1 †], Seung-Jung Shin[2]

[1†]*Student, Department of IT Convergence , Hansei University, Korea*
steezjang@gmail.com
[2]*Professor, Department of ICT Convergence, Hansei University, Korea*
expersin@hansei.ac.kr

Abstract

The online education platform market is developing rapidly after the coronavirus infection-19 pandemic. As school classes at various levels are converted to non-face-to-face classes, interest in non-face-to-face online education is increasing more than ever. However, the majority of online platforms currently used are limited to the fragmentary functions of simply delivering images, voice and messages, and there are limitations to online hands-on training. Indeed, digital transformation is a traditional business method for increasing coding education and a corporate approach to service operation innovation strategy computing thinking power and platform model. There are many ways to evaluate a computer programmer's ability. Generally, piecemeal evaluation methods are used to evaluate results in time through coding tests. In this study, the purpose of this study is to propose a comprehensive evaluation of not only the results of writing, but also the execution process of the results, etc., and to evaluate the programmer's propensity habits based on the programmer's coding experience to evaluate the programmer's ability and productivity.

Key words: *Bigdata, Machine Learning, coding, programmer test*

1. INTRODUCTION

Worldwide Industrial Structure Software-Based Technology With the reorganization centered on , various efforts are being made to enhance students' computing thinking skills. In particular, as software coding education becomes mandatory in elementary, middle, and high schools in Korea, the market for coding education, which aims to improve creativity and problem-solving skills through play, such as games, continues to grow. However, the existing online coding platform is operated and configured to evaluate coding skills through simple success failure through test case setup, so it is not easy to measure the ability of abstraction, logical thinking, synchronization, parallelism, flow control, interactivity, data expression, etc. necessary to develop a programmer's computing thinking skills. This study aims to design and implement a coding pattern analysis model that allows candidates or learners to quantitatively evaluate coding writing ability when writing coding using coding platforms. In this study, the user coding data is collected in big data platforms by producing code-writing programs or developing plug-in-type programs in existing code editor

Manuscript Received: November. 25, 2020 / Revised: December. 2, 2020 / Accepted: December. 1, 2020
Corresponding Author: expersin@hansei.ac.kr
Tel: +82-10-3664-0666, Fax: +82-31-450-5172
Professor, Department of ICT Convergence, Hansei University, Korea

programs, and data-marts are organized by loading them into big data platforms, and user-specific writing code analysis is performed through data classification/cluster/time sequence analysis using machine/deep learning frameworks. Based on the results analyzed above, the code writer's logic and thinking skills in a test case-based coding evaluation. A model is proposed that can assess the capability of programming evaluation indicators, such as data representation. The methodologies and models presented in this study allow us to determine the detailed evaluation indicators and programmers' abilities.

2. COLLECTION TARGET AND METHOD

2.1 Hypothesis and Data Wholesale

To analyze the propensity of S/W learners and measure their performance, the following hypotheses were established and the collection targets and methods were selected.

- What are the general characteristics of a sample (coding creator)?
- What is the correlation between the characteristics of the code writer?

Table 1. Data loading/discovery/analysis method

What is the data loading method? (log, real time)	HDFS/Redis
What is the data navigation method? (log,real-time)	spark/hive
What is the data analytics environment?	Tensorflow/Impala/mahout/sparkML
What are the general characteristics of a sample (coding record)?	time series analysis
What is the correlation between the characteristics of the code writer?	regression analysis
Detection of specific time points and anomalies	Prediction

2.2 Collection Layer

Flume is a software that consists of functions to address various collection requirements when collecting big data. When collecting data from a source, we have a lot of concerns about communication protocols, message formats, frequency of occurrence, data size, etc., and Plum provides features and architectures that can easily solve these problems.

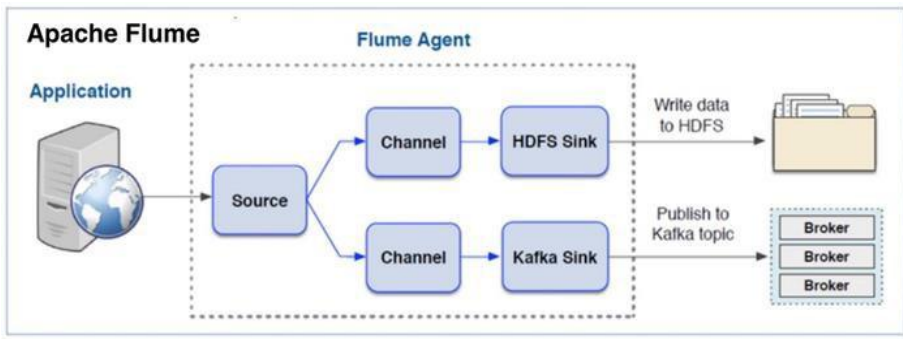

Figure 1. Collect large logarithmic data

The data collected may vary in processing and loading position depending on the nature of the data. Largely, it should be determined whether the data are batch batch data or real-time stream data according to the data generation cycle, and whether it will be processed or pre-validated according to the format of the data.

2.2 Processing Layer

In the process of understanding data, you find patterns, relationships, trends, and so on in the data, which is also known as the Exploration Data Analysis (EDA). The exploration process is a very important step in securing the quality and insight of big data before entering the analysis. After undergoing sophisticated postprocessing work (filtering, cleaning, integration, separation, etc.) of large unstructured data, the search results are immediately utilized as basic data for analytical marts.

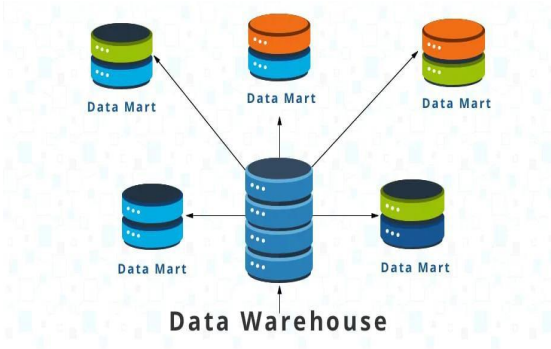

Figure 2. Big Data Warehouse

Spark: In-memory caching, normal batch processing(Batch Processing), Streaming, Machine Learning Hive: Data storage systems such as HDFS and HBase Analyze large data sets stored Impala: In-memory engine with relative scale parallel processing

Results:

Configure parallel navigation based on the characteristics of the source data(Detailed filtering, cleaning, consolidation, and separation of large unstructured data)

2.3 Search Layer

The large log data will be viewed, filtered, cleaned, combined, separated, and converted to HQL code log data. Real-time data will be viewed and refined further master data using Spark SQL.

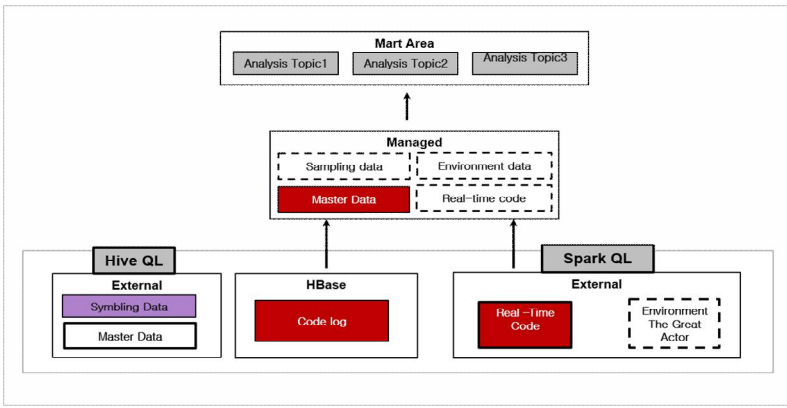

Figure 2. Model Summary

3. ANALYTICAL MODEL VERIFICATION

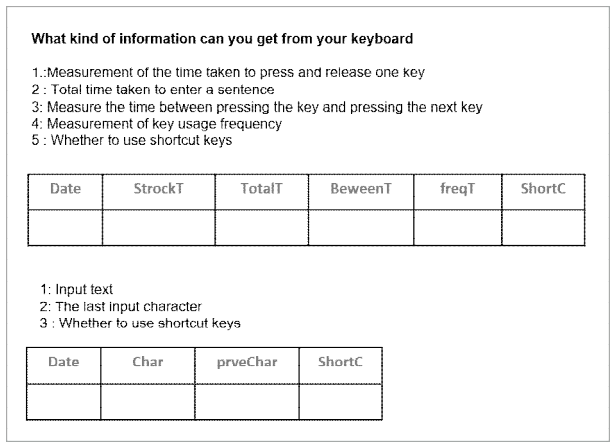

Figure 3. Big Data Warehouse

Based on this system, the following keyboard input situations were set and key input data was obtained from the user to analyze the behavior patterns between users.

The time difference between user A and user B's input value was obtained, and the first input value multiplied by 100 could be obtained by the error of the behavior pattern between users.

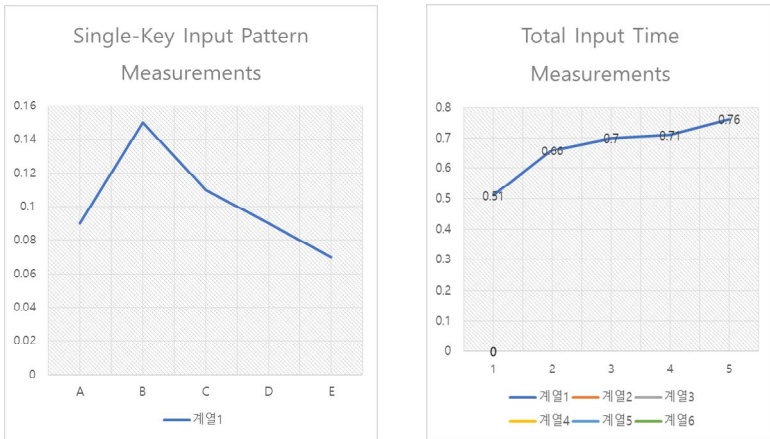

Figure 4. Behavior Patterns and Errors Between Users

5. CONCLUSION

We designed a big data solution system to collect log data generated and implement a whole series of processes from loading, navigation and processing to analysis. The large batch file data produced during the programming process and the log data generated on the big data platform based on the Hadoop ecosystem were either loaded in Hadoop or utilized real-time in-memory platforms. Data discovery has proposed a model that can quickly and effectively process and explore the loaded bulk data.

We build a data warehouse and leverage various exploration solutions to construct a data warehouse and build a data mart so that it can be used in the application service and data analysis stages. We proposed a model that can simultaneously analyze and provide real-time generated data at the same time as loading, and attempt to utilize statistical-based data analysis tasks and machine learning/dip learning frameworks to apply them to systems such as clusters, classifications, predictions, and recommendations, in some models, meaningful

results.

REFERENCES

[1] Hwan Soo Kang. "Design of Teaching · Learning Model for Programming Language Education." Journal of Digital Contents Society 13.4 (2012): 517-524.
UCI(KEPA) : I410-ECN-0101-2014-560-002753680

[2] Jeong Won Choi, Young Jun Lee. "The analysis of Learners` difficulties in programming Learning." The Journal of Korean Association of Computer Education 17.5 (2014): 89-98.
UCI(KEPA) : I410-ECN-0101-2018-037-003274455

[3] Jeong Won Choi, Young Jun Lee. "The analysis of Learners` difficulties in programming Learning." The Journal of Korean Association of Computer Education 17.5 (2014): 89-98. UCI410-ECN-0101-2018-037-003274455

[4] Seung-Won Lee, Yu-Hyun Choi. "Effects of Instructing Unlugged Activities in light of Self-Regulated Learning on Computational Thinking in Elementary Practical Arts Software Education." Journal of Korean Practical Arts Education 32.2 (2019): 105-121.
UCI(KEPA) : I410-ECN-0101-2019-374-000874104

[5] Junghwan Kim, Sukjae Lee, Rohae Myung. "Analysis of text entry task pattern according to the degree of skillfulness." HCI . (2007): 1081-1086.

[6] Kim, Jae-soo, Park suho, Lee minseok, Choi jihun. "The Design and Implementation of Python Education Coding WEB." Proceedings of the Korean Society of Computer Information Conference 27.1 (2019): 331-332.