International Journal of Advanced Culture Technology Vol.10 No.4 420-426 (2022)
DOI https://doi.org/10.17703/IJACT.2022.10.4.420

IJACT 22-12-50

Modeling of AutoML using Colored Petri Net

[1]Yo-Seob Lee

[1]Professor, School of ICT Convergence, Pyeongtaek University
yslee@ptu.ac.kr

Abstract

Developing a machine learning model and putting it into production goes through a number of steps. Automated Machine Learning(AutoML) appeared to increase productivity and efficiency by automating inefficient tasks that occur while repeating this process whenever machine learning is applied. The high degree of automation of AutoML models allows non-experts to use machine learning models and techniques without the need to become machine learning experts. Automating the process of applying machine learning end-to-end with AutoML models has the added benefit of creating simpler solutions, generating these solutions faster, and often generating models that outperform hand-designed models. In this paper, the AutoML data is collected and AutoML's Color Petri net model is created and analyzed based on it.

Keywords: *Machine Learning(ML), Automated Machine Learning(AutoML), Modeling, Colored Petri Net*

1. INTRODUCTION

The high degree of automation of AutoML models allows non-experts to use machine learning models and techniques without the need to become machine learning experts. Automating the process of applying machine learning end-to-end with AutoML models has the added benefit of creating simpler solutions, generating these solutions faster, and often generating models that outperform hand-designed models.[1] AutoML models can be used to compare the relative importance of each element in a predictive model.

In this paper, AutoML data is collected and AutoML's Color Petri net model is created and analyzed based on it.

2. AUTOMATED MACHINE LEARNING

Automated Machine Learning(AutoML) is the process of automating the application of machine learning to real-world problems. AutoML covers the entire pipeline from raw data sets to deployable machine learning models, and has been proposed as an artificial intelligence-based solution to the ever-growing challenge of machine learning applications.[2]

AutoML aims to alleviate this burden on data scientists by learning from experience designing and applying machine learning models and automating the tuning process, and giving practitioners access to off-the-shelf machine learning skills without extensive experience. The AutoML algorithm consists of three main components: a search space, a search strategy, and an evaluation strategy. Different AutoML systems provide

Manuscript received: October 31, 2022 / revised: November 25, 2022 / accepted: December 3, 2022
Corresponding Author: yslee@ptu.ac.kr
Tel: +82-31-659-8369, Fax: +82-31-659-8011
School of ICT Convergence, Pyeongtaek University, Korea

different levels of APIs that you can configure or customize based on your use case.[3]

The search space is a set of hyperparameters and the range of each hyperparameter to select. The scope of each hyperparameter can be defined according to the user's requirements and knowledge. The search space can be a pool of machine learning algorithms. In this case, we treat the type of machine learning algorithm as a hyperparameter to select. A search space can also be a hyperparameter of a particular machine learning algorithm, such as the structure of an machine learning model. The design of the search space is highly task-dependent, as different machine learning algorithms may need to be employed for different tasks.

The search strategy is a strategy that selects an optimal set of hyperparameters in a search space. Because AutoML is often an iterative trial-and-error process, the strategy often sequentially selects hyperparameters from the search space and evaluates their performance.

The evaluation strategy is a method of evaluating the performance of a specific machine learning algorithm instantiated by a selected hyperparameter. The evaluation criteria are often the same as those used for manual tuning, such as the validation performance of a model trained on a selected machine learning algorithm. Each AutoML algorithm consists of the following three core components as shown Figure 1.

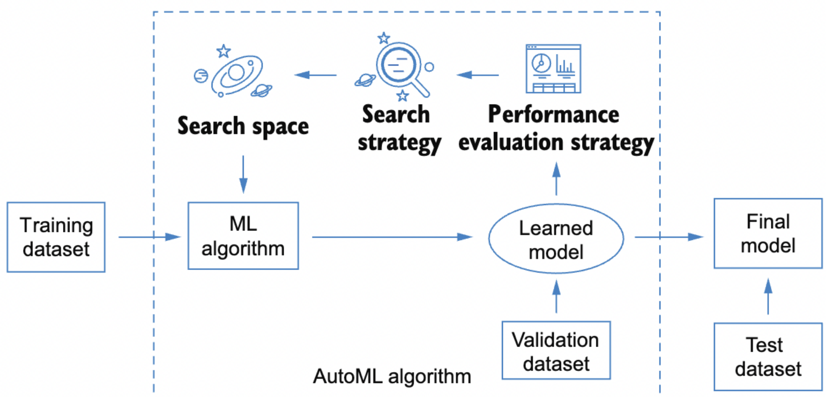

Figure 1. AutoML Process

3. AUTOML MODEL

3.1 AutoML Model

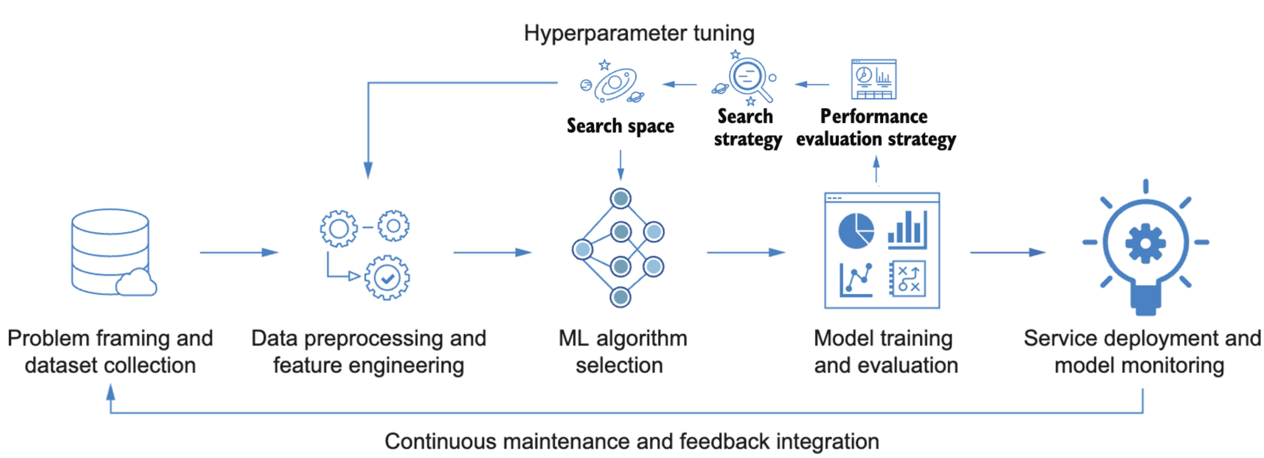

Figure 2. AutoML Model

AutoML model is a model showing a flow chart that automates all processes from problem definition, data collection, preprocessing, model learning, evaluation, and service application, and is composed of several components as shown in Figure 2. The AutoML model is a model to increase productivity and efficiency by automating all processes as much as possible. In particular, it is possible to effectively develop high-quality models by minimizing the intervention of model developers in the process from data preprocessing to algorithm selection and tuning.

3.2 Modeling of AutoML Model using CPN Tools

AutoML model is constructed based on the color petri net by Jensen[4], and the AutoML model is modeled using a tool called CPN Tools[5]. CPN Tools is a program that allows modeling and simulation with color petri nets.

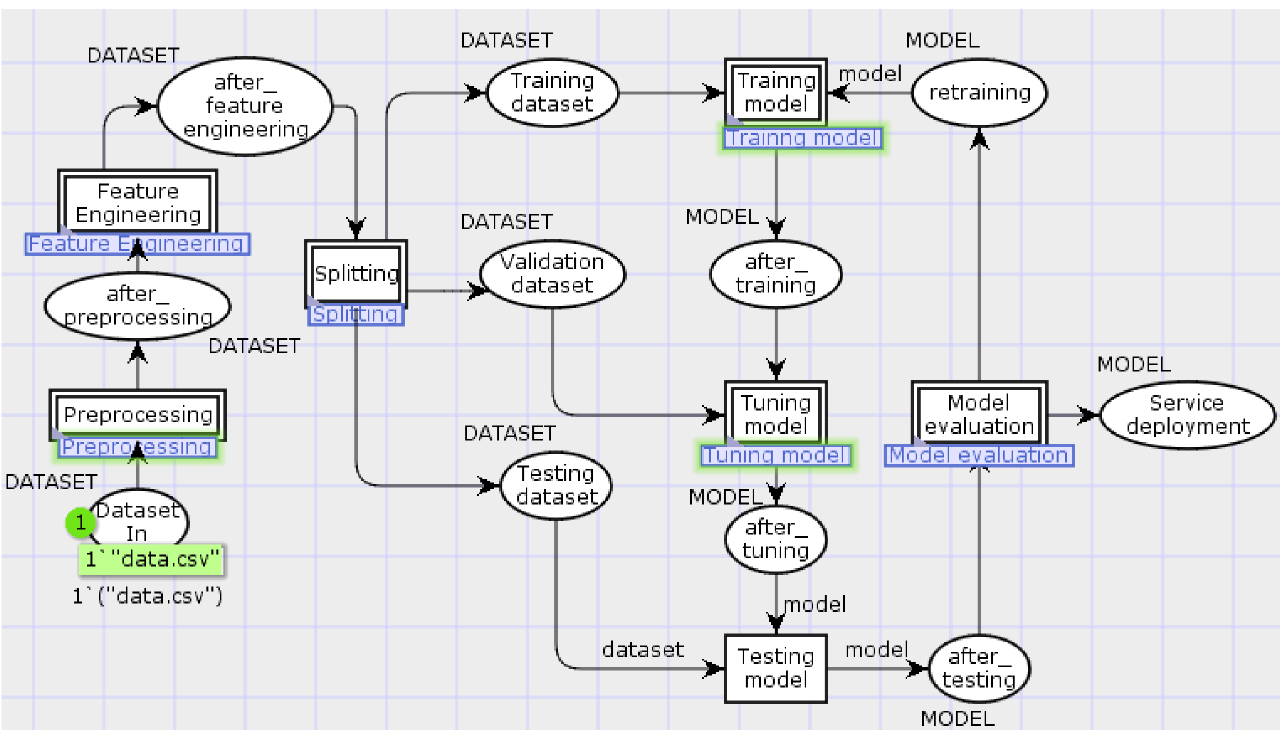

Figure 3. AutoML Model in CPN Tools

Figure 3 shows the AutoML model as a CPN tool. The input pattern is expressed in the form of (Dataset). In 1`("data.csv") of the input pattern of Figure 3, data.csv represents a specific dataset. AutoML models consists of a preprocessing module that preprocesses data, a feature engineering module that extracts features from preprocessed data, a splitting module that separates multiple datasets, a training model module that trains a model with a training dataset, and a tuning model that tunes with a validation dataset, a test model module that tests with test dataset, and a model evaluation module that evaluates the performance of the model.

In the training model module in Figure 4, the model is trained with the training dataset. The machine learning algorithm is selected from the machine learning algorithm pool and retrained if there is a retraining request in the model evaluation module.

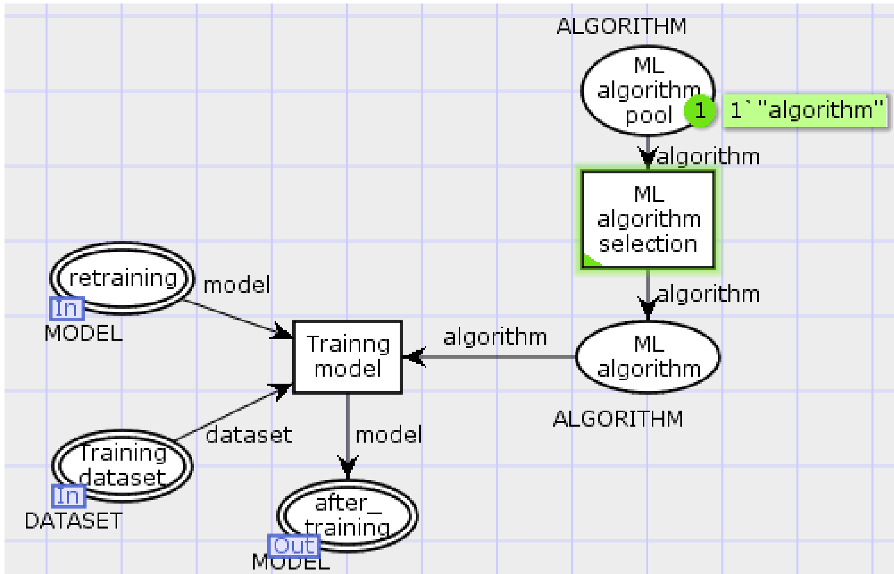

Figure 4. Training Model Module

In the Tuning model module shown in Figure 5, the model is tuned with the validation dataset. The hyperparameter is selected from the hyperparameter pool and used.

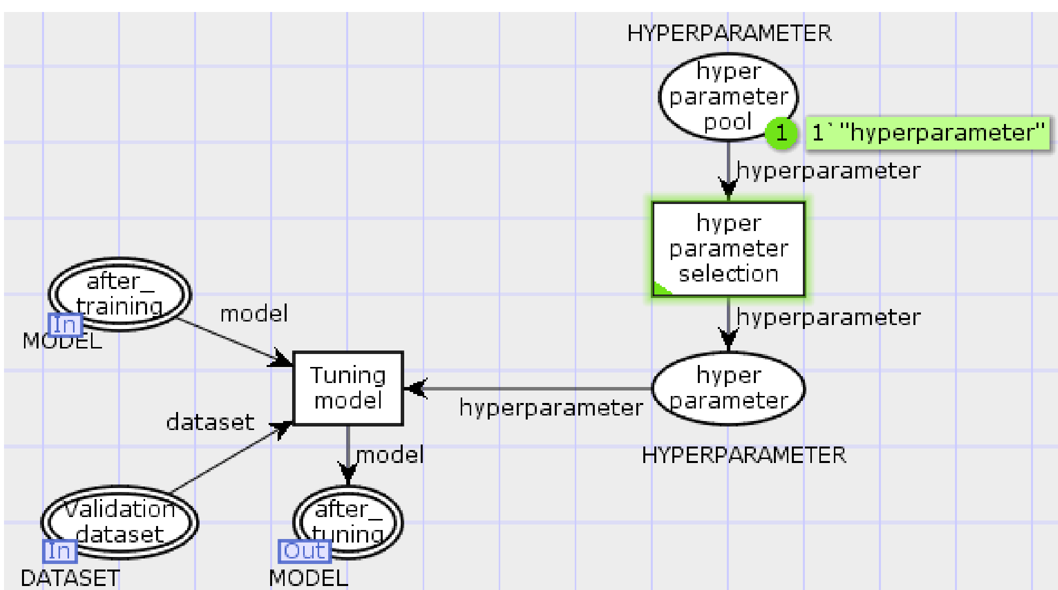

Figure 5. Tuning Model Module

The Model evaluation module in Figure 6 evaluates the machine learning model. After checking whether the machine learning model is a classification model or a regression model, if it is a classification model, AUC PR, AUC ROC, Accuracy, and Log loss are measured. When the machine learning model is a regression model, MAE, RMSE, and RMSLE are measured. If the evaluation result of the machine learning model is "good", it is deployed as a service, but if the evaluation result is "poor", it is retrained.

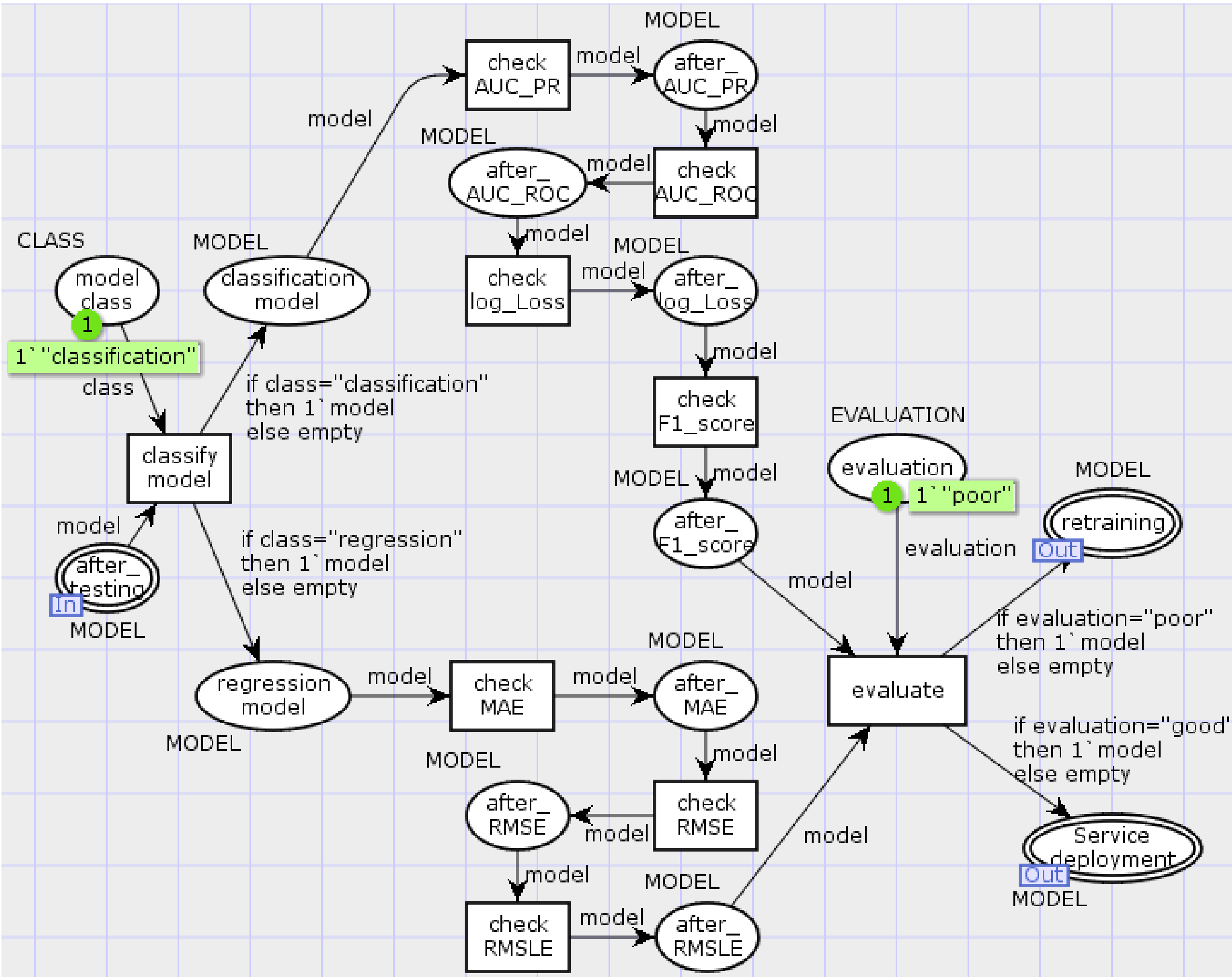

Figure 6. Model Evaluation Module

AUC PR, AUC ROC, Accuracy, and Log loss are evaluation metrics for classification models.[6] AUC PR is the area under the precision-recall(PR) curve. This value ranges from zero to one, where a higher value indicates a higher-quality model. AUC ROC is the area under the receiver operating characteristic(ROC) curve. This ranges from zero to one, where a higher value indicates a higher-quality model. Accuracy is the fraction of classification predictions produced by the model that were correct. Log loss is the cross-entropy between the model predictions and the target values. This ranges from zero to infinity, where a lower value indicates a higher-quality model.

MAE, RMSE, and RMSLE are evaluation metrics for regression models.[6] MAE is the mean absolute error (MAE) is the average absolute difference between the target values and the predicted values. This metric ranges from zero to infinity; a lower value indicates a higher quality model. RMSE is the root-mean-square error metric is a frequently used measure of the differences between the values predicted by a model or an estimator and the values observed. This metric ranges from zero to infinity; a lower value indicates a higher quality model. RMSLE is the root-mean-squared logarithmic error metric is similar to RMSE, except that it uses the natural logarithm of the predicted and actual values plus 1. RMSLE penalizes under-prediction more heavily than over-prediction. It can also be a good metric when you don't want to penalize differences for large prediction values more heavily than for small prediction values.

4. ANALYSIS of AUTOML MODEL

4.1 Modeling of AutoML Model using CPN Tools

AutoML model expresses the process of learning a dataset with a machine learning algorithm through the Training model module, and it expresses the tuning of the machine learning model with hyperparameters through the Tuning model module. In addition, it evaluates the performance of the machine learning model through the model evaluation module and expresses retraining when the evaluation result is poor.

4.2 Analysis of Proposed Model using the Occurrence Graph

The AutoML model is modeled by a color Petri net, and uses the occurrence graph to construct a directional graph representing the state of the system that the node can reach and the state where the arc can change. A complete analysis of the model is possible using this graph.

Figure 7 shows the graph as it happens, from data collection, preprocessing, model training, evaluation, through to service application.

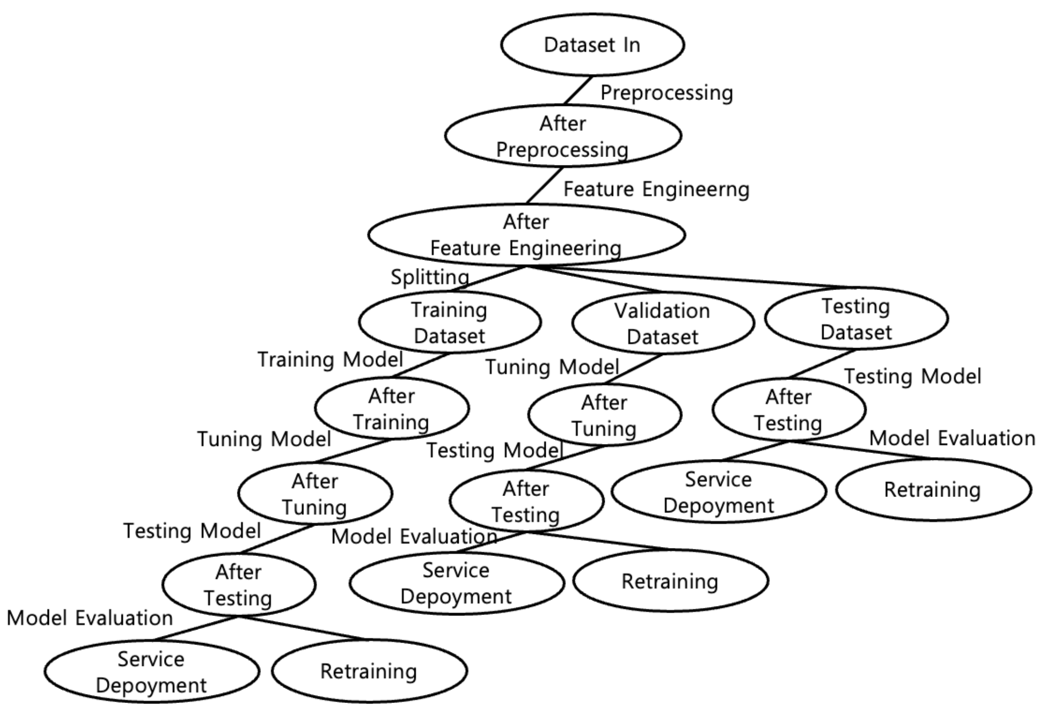

Figure 7. Occurrence Graph when a service is deployed

Figure 8 shows the graph that occurs when the machine learning model enters the evaluation process after testing the machine learning model.

After the machine learning model is classified, the classification model is evaluated according to the evaluation measurement method of the classification model, and the regression model is evaluated according to the evaluation measurement method of the regression model. If the evaluation result is "good", it is placed as a service, and if it is "poor", it is confirmed that it enters the retraining process.

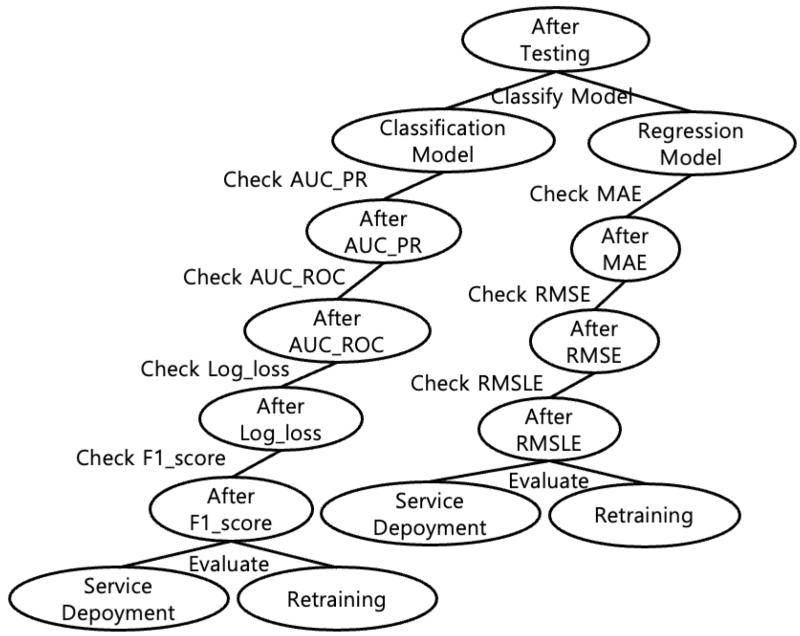

Figure 8. Occurrence Graph of Model Evaluation Module

5. CONCLUSION

AutoML appeared to increase productivity and efficiency by automating as much as possible inefficient tasks that occur while repeating the process of developing machine learning models and introducing them to actual operations whenever machine learning is applied. The high degree of automation of AutoML models allows non-experts to use machine learning models and techniques without the need to become machine learning experts. Automating the process of applying machine learning end-to-end with AutoML models has the added benefit of creating simpler solutions, generating these solutions faster, and often generating models that outperform hand-designed models.

In this paper, we collected data on AutoML, and based on it, we created and analyzed AutoML's color Petrinet model. This model allows you to apply a variety of machine learning algorithms, hyperparameters, evaluation metrics, and more. I also think it will help in designing and developing software for AutoML.

ACKNOWLEDGEMENT

This paper was supported by the Research Fund, 2021, Pyeongtaek University in Korea.

REFERENCES

[1] Y. Lee, "Analysis of Automatic Machine Learning Solution Trends of Startups," Vol.8, No.2, International Journal of Advanced Culture Technology, 2020, *https://doi.org/10.17703/IJACT.2020.8.2.297*.
[2] Automated machine learning, *https://en.wikipedia.org/wiki/Automated_machine_learning*.
[3] Q. Song, et. al, "Automated Machine Learning In Action," Manning Publication, 2022.
[4] Kurt Jensen and Lars M. Kristensen, “Colored Petri Nets — Modeling and Validation of Concurrent Systems,” Springer-Verlag Berlin, 2009.
[5] CPN Tools, *http://cpntools.org/.*
[6] Evaluating models, *https://cloud.google.com/automl-tables/docs/evaluate*.