



Introducing the BrewAI AutoML Tool

Siu Lung Ng¹(✉), Fethi A. Rabhi¹, Gavin Whyte², and Andy Zeng²

¹ School of Computer Science and Engineering, University of New South Wales, Sydney 2052, Australia

{siu_lung.ng, f.rabhi}@unsw.edu.au

² BrewAI, Head Office, Suite 03, level 22, 56 Pitt Street, Sydney 2000, Australia

{gavin.whyte, andy.zeng}@brewai.com

Abstract. AutoML tools provide an automation service for data scientists and software engineers to save time from data preprocessing and modeling building. Existing AutoML tools usually require users to have data science knowledge and programming skills to use the services, however, most non-expert and business users do not have such skills to use these AutoML tools. In addition, many AutoML tools require a special infrastructure or cloud provider. In this paper, we introduce BrewAI: a commercial-grade tool that provides an easy-to-use AutoML service for business users. The paper describes how the use of service-oriented computing design principles gives BrewAI flexibility, scalability and performance at a reasonable cost. The paper also describes a case study that shows how BrewAI enables business users to outperform more than three-quarters of Kaggle competitors in an NLP classification task.

Keywords: AutoML · Web application · ML pipeline · ML for business

1 Introduction

The application of Machine Learning (ML) techniques in industry presents many challenges. Although there are many libraries and technologies that provide ML functionalities “out of the box”, many challenges remain such as deciding which technique works with which problem, transforming datasets into the right format expected by each software product, ensuring they are of sufficient quality, fine-tuning the ML parameters and validating the models. Once a model needs to be made operational, constructing complete “ML pipelines” that fit with an organisation’s internal IT systems and that can be operated and maintained by non-specialised staff also presents many challenges. In addition, the performance of such ML pipelines is very sensitive to a number of design decisions, which constitutes a high entry barrier for new users.

The demand for simple solutions that work without the need to be operated by ML-experts has given rise to the field of automated machine learning (AutoML). The goal of AutoML is to make ML more systematic and efficient by automating several human-extensive activities. It enables domain scientists to apply ML without the need to understand and learn the underlying technologies in detail.

Whilst the goal of scientific research is to create AutoML tools that aim for full automation, several companies are now developing their own AutoML systems that aim to offer some form of “semi-automation” in assisting organisations to deploy ML pipelines at lower costs. However, many existing solutions are still expensive or require expert staff to operate them. This paper presents an alternative solution called BrewAI which is designed for providing a care-free user experience and explainability combined with a scalable technological infrastructure at a modest cost.

The rest of the paper is structured as follows. Section 2 defines some basic AutoML concepts. Section 3 describes BrewAI and Sect. 4 a case study of applying the system. Section 5 concludes this paper.

2 Background

AutoML refers to the automation of several activities related to ML such as experiment design, data collection, data cleanup, missing data imputation, feature selection, model discovery, model explanation, hyperparameter optimization (HPO), model monitoring and anomaly detection. A recent survey [1] describes the AutoML pipeline as consisting of several processes: data preparation, feature engineering, model generation, and model evaluation (see Fig. 1).

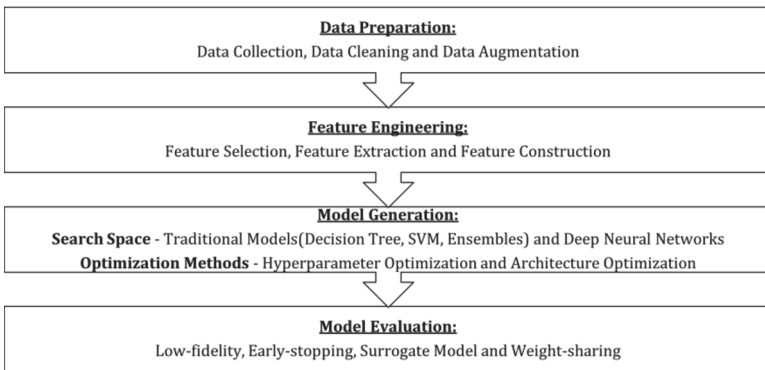


Fig. 1. An overview of AutoML pipeline (adapted from He et al. [1]).

There is a huge diversity in the tools available to support AutoML. Auto-WEKA [2] is one of the first AutoML systems based on the well-known WEKA machine learning toolkit. TPOT [3] automatically constructs and optimizes tree-based machine learning pipelines from a small set of fixed ML components that are connected in predefined ways. Auto-sklearn [4] is similar but adds several improvements such as meta-learning for warm starting the optimization and automatic ensembling. Inspired by Auto-sklearn, Auto-PyTorch [5] uses an ensembling method to implement an automated post-hoc ensemble model selection [6] for efficient optimization.

As mentioned earlier, several companies are now developing their own AutoML systems that aim to assist organisations to deploy ML pipelines with minimal effort and costs. Big tech companies are offering AutoML products such as Azure Machine Learning [7] and Amazon SageMaker Autopilot [8] and Google’s AutoML [9]. Other offerings exist from other large companies such as SAS and IBM (Automated Artificial Intelligence or AutoAI¹).

One of the problems with these solutions is that they are tied to a particular platform or cloud infrastructure or installed on a user’s desktop. Some solutions are more specialised and can work in different environments. They include H2O Driverless AI [10, 11] which supports fully- or semi-automated feature engineering and selection, model tuning and training of predictive models and DataRobot² which has recently acquired Algorithmia.³ Still, these solutions may be tied to specific infrastructures that bring high costs to SMEs and government organisations and may not be easy to deploy and operate by non-expert staff. In this paper, we investigate new opportunities for addressing these issues via alternative AutoML tools such as BrewAI.

3 Introduction to BrewAI

3.1 General overview

BrewAI⁴ is designed to be a simple and cost-effective solution that delivers ML functionalities for organisations that don’t have a specialised staff or alternatively used by specialized staff with the intention of reducing time to market with AI models. Like other AutoML systems, BrewAI simplifies the creation and deployment of ML models. Starting from just a simple spreadsheet, a user can train, build and deploy a commercial-grade ML model within an IT infrastructure with minimal efforts [12].

Its software architecture is based on service-oriented design principles in which autonomous software services can operate and communicate independently from each other. This architecture is illustrated in Fig. 2.

The BrewAI engine which is at the heart of the system is responsible for tackling supervised learning problems using deep learning methods. Driven by feature Importance, the business user has a say in fine-tuning, can deal with different data types including numerical, text, categorical and binary. New data types such as images will be released on the roadmap. The engine is built over several other systems. Its code base relies on the PyTorch library. Hyperparameter optimization is automatically conducted using Optuna [13] and HyperOpt [14]. Feature importance is developed utilising Optuna so a user can dynamically construct the search spaces for the hyperparameters.

BrewAI has the ability to aggregate data from different sources, each data source can be independently serviced by an API that can operate autonomously to encode and feed data into the model. All BrewAI software components are virtualised in containers using Kubernetes [15]. This allows them to be deployed on a scalable cloud platform (e.g.,

¹ <https://dataplatfom.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-overview.html>.

² <https://www.datarobot.com/>.

³ <https://algorithmia.com/>.

⁴ www.brewai.com.

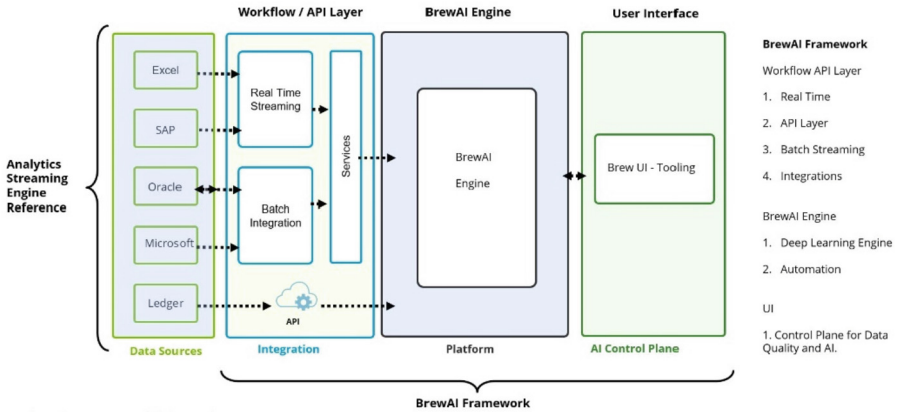


Fig. 2. Components of the BrewAI architecture (adapted from Rabhi et al. [12]).

Amazon’s EC2). The use of an elastic cloud means the system can adapt to different data sizes and loads.

An important component is the BrewAI User Interface which displays results at different stages of the ML pipeline in a way that is easily comprehended by the user. The user can also direct the different stages like training via simple button clicks.

3.2 BrewAI Model Building Process

To use BrewAI, users should firstly log in to the web server and the software will automatically go through five stages to compute the prediction results from the AutoML model. Figure 3 illustrates the five stages in the BrewAI model building process. There is no restriction in the order to follow when performing these five stages, users can jump into any stage to check the previous actions in that specific stage [12].

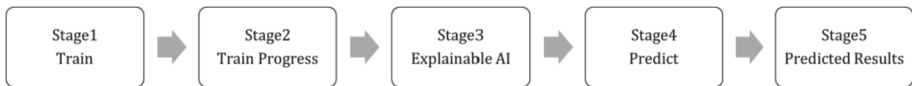


Fig. 3. BrewAI’s five stages for AutoML.

Stage 1 - Train. In this stage, users can upload the tabular data file to the BrewAI webpage through the interface. Figure 4 shows BrewAI’s interfaces for training data upload (top) and model training submission (bottom). After clicking the “Load Data” button (see Fig. 4 Top), a dataset preview will be shown, users will then select the target column for prediction. The model will be built after clicking the “Submit Model” button (see Fig. 4 Bottom). BrewAI will automatically define the type of machine learning tasks (regression or classification), handle the data pre-processing, and build the AutoML model.

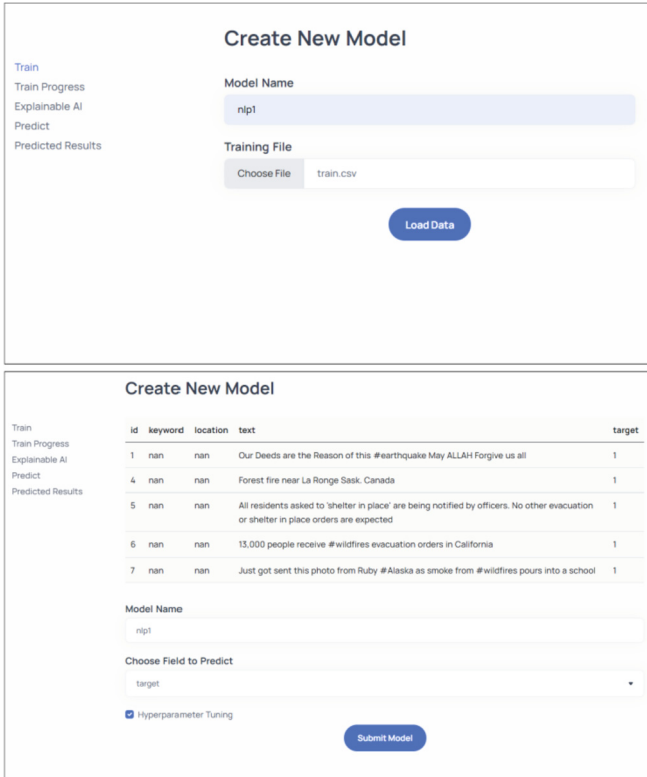


Fig. 4. Top: BrewAI’s interface for training data upload. Bottom: BrewAI’s interface for loaded dataset review and model training submission.

Stage 2 – Train Progress. Users can see the status of data processing and model building. No action is required from the users in this stage.

Stage 3 – Explainable AI. This stage is to show explainable details of the data and model after model training is completed. The explainable data shows the details of data quality and data type for each input feature. The explainable model shows the details of the class distribution, model performance, confusion matrix, performance by class, and feature importance. Figure 5 shows BrewAI’s interface for data and model explanation.

BrewAI also provides intuitive visualization (see Fig. 6) for prediction results such as different charts for confusion matrix and feature importance.

Stage 4 – Predict (Model Inference). In this stage, users can select a specific model trained in stage 2 to predict the test dataset. All previously trained models will be saved in BrewAI’s server for future inference, users are allowed to come back to this page anytime and select a trained model to do predictions for test datasets.

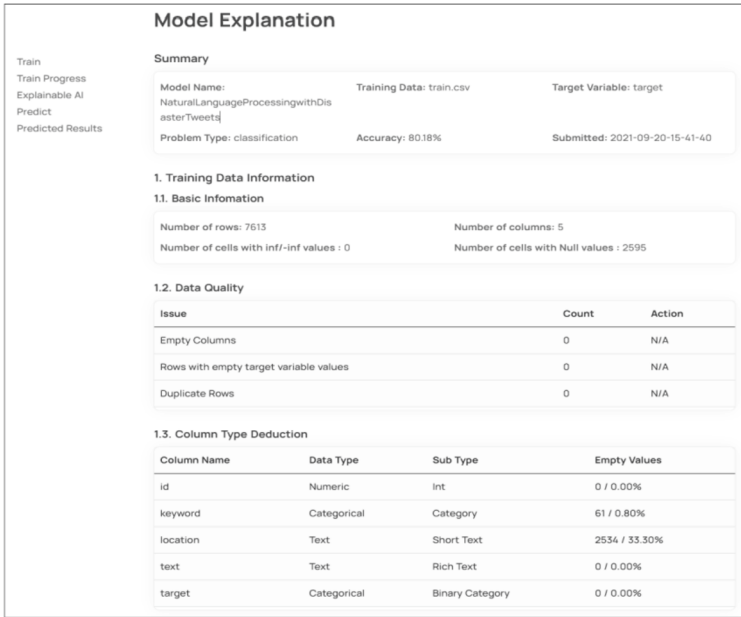


Fig. 5. BrewAI’s interface for data and model explanation.

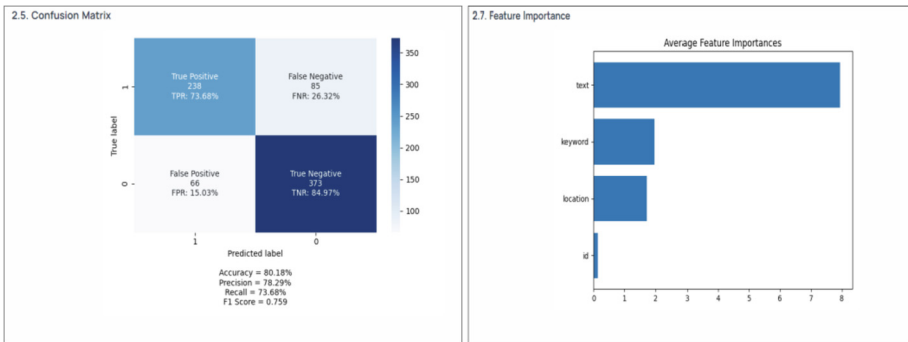


Fig. 6. Left: interface for confusion matrix. Right: interface for feature importance.

Stage 5 – Predicted Results. After finishing the prediction in stage 4, users can explore the prediction results in this stage. BrewAI also allows users to preview and download previously predicted results to csv files by clicking buttons (see Fig. 7).

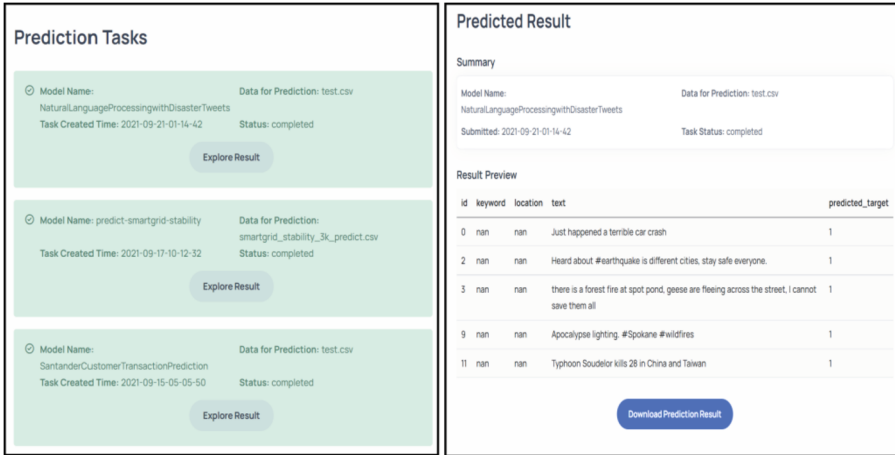


Fig. 7. Left: interface for result exploration. Right: interface for result preview and download.

4 Case Study of Using BrewAI

In this case study, we went through the model-building process in the BrewAI web application and applied the BrewAI service to a selected dataset from the Kaggle competition [16]. The purpose of this case study is to show how BrewAI enables non-expert and business users to apply the machine learning model to do the prediction through AutoML.

4.1 Experimental Setting

Kaggle competitions include various types of machine learning tasks made by Kaggle or other companies like Google and Microsoft. The competitions allow people to compete for machine learning predictions based on the given datasets. In our case study, we selected the dataset from the “Natural Language Processing with Disaster Tweets” Kaggle competition for the experiment. This dataset is a csv file which consists of a table of 4 columns (features) and 7613 rows (samples) in the training dataset including integers, strings, and random empty values. The competition task is to classify whether a given tweet is about a real disaster or not for the test dataset with 3263 samples.

Solving such a natural-language-processing (NLP) problem requires advanced data-preprocessing and model-building techniques. The input features of this dataset are mixed with a column of integers, a column of a single word, a column of short text, and a column of long text, some features include a small number of empty values and the prediction target is a column of Boolean type data. Without AutoML, data scientists usually apply multiple data pre-processing strategies for this kind of machine learning task, such as normalization, one-hot encoding, lemmatization, tokenization, word embedding, etc. Also, designing good a NLP model requires advanced machine learning knowledge. Unfortunately, business analysts or non-expert users usually do not have such skills for NLP machine learning tasks. In this case study, we want to

demonstrate how BrewAI enables business users to apply machine learning models to solve NLP classification tasks.

We simply uploaded the training dataset (the csv file) to the BrewAI webpage and then went through the five stages mentioned earlier. The predicted result for the test dataset was downloaded from another BrewAI server and then submitted to the Kaggle “leaderboard” for comparison among other competitors. The Kaggle system shows that our BrewAI model achieved an F1 score of 0.80937 and ranked 295 out of 1241 competitors (tested on Sep23 2021). This means without any effort in environment setup, data pre-processing and model building, our BrewAI AutoML model beat more than three quarters (76.2%) of Kaggle competitors in this NLP classification task.

4.2 Evaluation

The evaluation is based on the usability, data/model explainability, and performance for non-expert and business users.

- For the usability aspect, BrewAI does not require any data pre-processing and modeling skills to apply machine learning models. The interface consists of only dropdown selection controls and confirm buttons which are easy enough for anyone with a basic level of computer proficiency to use. The model building processing is fully automatic without worrying about parameter settings. The only requirement for using BrewAI is users need to understand which is the target they want the AutoML model to learn. Although BrewAI only works with tabular/structural data and does not work with non-tabular data, users can still transform any form of data into a tabular format for classification and regression tasks.
- For the model explainability aspect, BrewAI can show necessary details about the data and model that business users can understand. The users can have a summary of their datasets without any programming skills or manual data analysis. There has no explanation about the model parameters or model generation process in BrewAI. Nevertheless, most business users only focus on the data and results such as data quality, performance, and feature importance that BrewAI can provide.
- For the model performance aspect, although the result did not shine in this Kaggle competition, non-expert users with the BrewAI model still beat more than three quarters (>900) of competitors and ranked top 25% with simple button clicks, and with no data pre-processing and model building efforts.

5 Conclusions and Future Work

Recent years have witnessed the rise in using AutoML tools in industry as they provide automation facilities for data preprocessing and modeling building tasks associated with the use of machine learning methods. Despite the availability of a wide range of tools and the promise of costs savings, these advantages have not necessarily materialised in practice as there are hidden staff and infrastructure costs associated with many solutions.

Some AutoML tools such as H2O and Auto-Pytorch require users to have programming skills for setting up the python environment, which business users usually do not

have. On the other hand, paid AutoML services by big technology companies such as Google AutoML and AWS SageMaker have better usability than open-source products, but they are usually bundled with other related cloud services and that could cost more than just an AutoML service. These cloud services also require tedious cloud-platform setup by cloud engineers before enabling AutoML services.

Owing to its architectural design based on service-oriented principles, BrewAI can be deployed and configured in different ways and thus able to satisfy different needs, including a low-cost solution for Small and Medium Enterprises (SMEs). It does not have a huge panoply of ML techniques, focusing on regression and classification using deep learning techniques. This is because it is primarily targeting business users with a good understanding of the data but without programming skills and advanced data science knowledge, such as business analysts.

Research work that facilitates the application of AutoML tools in industry is expected to intensify in the near future addressing many remaining challenges. Amongst them, BrewAI is expected to deal with those that contribute to enhancing the user experience and the quality of the underlying software services:

- **Inclusion of business objectives:** non-technical users usually want business insights from the machine learning model instead of accuracy and training time that most technical users are concerned about.
- **Need for transparency:** non-technical users do not understand the black-box nature of deep learning. Current explainable AI is still lacking the capability to explain business decision-making.
- **Data quality:** most progress has been done on the model building side, the bottleneck in many AutoML tools is on the data side to ensure high data quality during the acquisition.
- **Performance:** to achieve good performance and avoid input pipeline stalls, businesses need more control over how to achieve the best balance between accuracy and cost considerations (making optimum use of resources available).
- **Data security:** provide data protection or encryption solutions for data uploading and result downloading.
- **Cost-effectiveness:** optimize the utilization of cloud resources and provide more flexible usage plans for the users.

References

1. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. knowledge-based systems. **212** (2021). <https://doi.org/10.1016/j.knosys.2020.106622>
2. Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K.: Auto-WEKA: automatic model selection and hyperparameter optimization in WEKA. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated Machine Learning: Methods, Systems, Challenges, pp. 81–95. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_4
3. Olson, R.S., Moore, J.H.: TPOT: a tree-based pipeline optimization tool for automating machine learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated Machine Learning: Methods, Systems, Challenges, pp. 151–160. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_8.

4. Feurer, M., Klein, A., Eggenberger, K., Springenberg, J.T., Blum, M., Hutter, F.: Auto-sklearn: efficient and robust automated machine learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) *Automated Machine Learning: Methods, Systems, Challenges*, pp. 113–134. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_6
5. Zimmer, L., Lindauer, M., Hutter, F.: Auto-pytorch: multi-fidelity MetaLearning for efficient and robust AutoDL. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 3079–3090 (2021). <https://doi.org/10.1109/TPAMI.2021.3067763>
6. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: *Ensemble selection from libraries of models* (2004)
7. Barga, R.: *Predictive Analytics with Microsoft Azure Machine Learning*. Apress, Berkeley (2015)
8. Das, P., et al.: Amazon sagemaker autopilot: a white box AutoML solution at scale. In: *Proceedings of the 4th Workshop on Data Management for End-To-End Machine Learning, DEEM 2020 - In conjunction with the 2020 ACM SIGMOD/PODS Conference*. Association for Computing Machinery, Inc (2020). <https://doi.org/10.1145/3399579.3399870>.
9. Wong, C., Houlsby, N., Lu, Y., Gesmundo, A.: *Transfer Learning with Neural AutoML*. (2018)
10. H2O.ai | AI Cloud Platform, <https://www.h2o.ai/>. Accessed 22 Sep 2021
11. Cook, D.: *Practical Machine Learning with H2O*. O'Reilly Media, Newton (2016)
12. Rabhi, F., Ng, A., Mehandjiev, N.: *AutoML applications in business: a case study using BrewAI*. FinanceIT@UNSW Research Group Internal report, Sydney (2021)
13. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631. Association for Computing Machinery (2019). <https://doi.org/10.1145/3292500.3330701>
14. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: a python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **8**, 14008 (2015)
15. Kubernetes. <https://kubernetes.io/>. Accessed 22 Sep 2021
16. Kaggle: Your Machine Learning and Data Science Community <https://www.kaggle.com/>. Accessed 23 Sep 2021