AI-Democratization: From Data-first to Human-first AI

Dennis Rall, Thomas Fraunholz
WOGRA AG
Hery-Park 3000, 86368 Gersthofen
{dennis.rall, thomas.fraunholz}@wogra.com

Bernhard Bauer
University of Augsburg
Software Methodologies for Distributed Systems
Universitaetsstrasse 6a, 86159 Augsburg
bernhard.bauer@informatik.uni-augsburg.de

Abstract. *In recent years, there have been significant advances in artificial intelligence (AI) research. However, not everyone can benefit from these advancements because it requires a significant amount of expertise to apply them. The concept of democratizing AI, known as AI democratization, aims to make AI accessible to everyone. One exciting development in this field is the shift from a model-first to a data-first approach. Previously, the focus was on building and training complex models, which required a high level of expertise. However, new high-level frameworks have drastically simplified the process, making it possible to create machine learning (ML) models based on data descriptions automatically. This has paved the way for data-first AI, where the focus is on the data engineering rather than model creation. The next logical step in this progression is a human-first approach to AI. By simplifying the process of creating ML models even further and leveraging the insights gained from data-first AI, we can make AI accessible to a even broader range of users. We underpin our theoretical considerations with the development of the Open space for Machine Learning (Os4ML) platform and summarize our achievments so far in this endeavor. Our approach has the potential to significantly lower the barrier to entry for using AI and to accelerate AI adoption across a wide range of industries and domains.*

Keywords. Human-first AI, data-first AI, Artificial Intelligence, AutoML, No-Code ML, Low-Code ML

1 Introduction

The field of artificial intelligence (AI) has seen tremendous progress, with the potential for AI to be helpful in many different areas ("The AI Index Report", 2023). However, due to the high level of expertise required and the shortage of AI experts, many fields have not yet been able to benefit from this technology. To address this issue, there has been an increasing focus on AI democratization, which aims to make AI accessible to as many users as possible (Allen et al., 2019). The goal of AI democratization is to empower individuals who lack expertise in the field to leverage AI for their own benefit and to bring the benefits of AI to a wider audience.

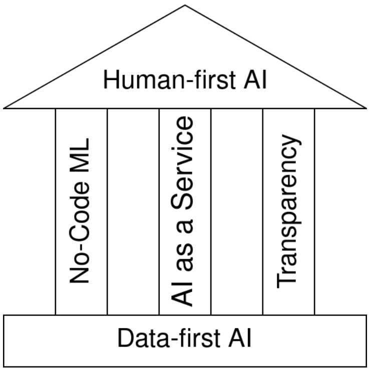


Figure 1: In order to advance from data-first AI to human-first AI, it is imperative to provide users with a no-code ML solution, make it available as a service, and ensure complete transparency throughout the development process.

One approach that has emerged in this context is data-first AI, also known as data-centric AI (Zha et al., 2023, "Data-centric Artificial Intelligence: A Survey"). Unlike traditional approaches that focused on designing complex ML models, data-first AI emphasiszes the importance of optimizing both the data's quality and quantity. Once high-quality data is obtained, low-code tools can be used to create ML models (Cabot, 2020). This approach has shown promising results, as it can lead to better accuracy and more robust models.

The data-first approach significantly reduces the need for AI expertise, but some level of technical knowledge is still required. To further democratize AI, the next step is to move towards human-first AI, which aims to make access to AI even easier. In this article, we will first discuss the data-first approach to AI and then explain how we can build human-first AI from it. Specifically, we will focus on three key components, as illustrated in Figure 1. First, we need to reduce the required coding effort even further and move from low-code to no-code ML. Second, we need to make access to this AI easier by offering it as a service. Third, the entire process needs to be transparent and trustworthy. Throughout this article, we will delve into each of these components in detail. Finally, we will introduce our platform, Open space for Machine Learning, which incorporates these components and can help to build effective AI solutions.

2 Data-first AI

The data-first approach has shifted the focus of machine learning from designing models to data engineering tasks. This approach represents a significant improvement over the traditional model-first approach, which required developers to write low-level ML code to design models. Over time, higher-level tools and frameworks have emerged, which provide a more abstract interface for building ML models.

The data-first approach emphasizes the importance of developing and maintaining high-quality training data. This includes tasks such as data collection, data validation, data cleaning and data integration (Whang et al., 2023). By focusing on these tasks, developers can ensure that their models are based on accurate, diverse, and relevant data, which can improve their performance and accuracy.

In addition to developing the training data, data maintenance is also essential in the data-first approach. This includes tasks such as data understanding, data quality assurance, and data acceleration. Developers must ensure that the data they are working with is trustworthy and reliable, as poor data quality can lead to incorrect predictions or biased models (Zha et al., 2023, "Data-centric AI: Perspectives and Challenges").

To create ML models for high-quality data, declarative Machine Learning Frameworks (Molino & Re, 2021) are used. For instance, we can take the PetFinder dataset (Zhang & Zhang, 2019) as a case study, containing information about animals in a shelter. Figure 2 illustrates an example of a dog (type 1) named Fenny, five years old, and not adopted for 100 days (AdoptionSpeed 4).

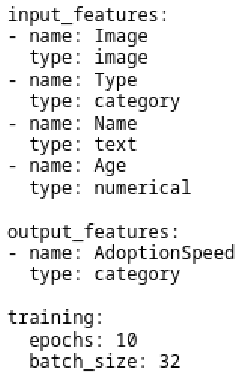
In contrast to the model-first approach, which demands considerable knowledge and experience in machine learning algorithms and model architecture, the data-first approach requires expertise in the domain of the problem. Once the data engineering tasks are completed, describing the data is sufficient to train ML models. An description for the PetFinder dataset would entail the presence of multiple features for each data point, including an image, a numerical age value, a string name value, and categorical values indicating the animal's type and adoption speed. By providing this data description, a declarative ML tool can be used to create an ML model that is tailored to the specific problem and capable of accurately predicting the adoption speed. Overall, the data-first approach represents a significant advancement in the field of machine learning, providing a more streamlined and effective way to develop machine learning models.

Image	Type	Name	Age	AdoptionSpeed
	1	Fenny	5	4

Figure 2: Example of a pet of the PetFinder (Zhang & Zhang 2019) dataset. Fenny is a 5-year-old dog (Type 1) that has not been adopted within 100 days (AdoptionSpeed 4).

2.1 Ludwig as an example of a declarative ML framework

Declarative ML frameworks, such as Ludwig (Molino, Dudin & Miryala, 2019), Overton (Re, 2020), Fastai (Howard & Gugger 2020), and AutoGluon (Erickson et al., 2020), allow for the streamlined creation and training of machine learning models. Ludwig, for example, employs an Encoder-Combiner-Decoder model, where inputs are first processed by tailored encoders based on their data type. The encoded values are then combined using a combiner, and the decoder is used to make predictions on the target output. The specific decoder employed is dependent on the data type of the output features. This approach optimizes the model for the given dataset, resulting in precise and efficient predictions. Figure 3 shows a possible configuration for training a Ludwig model for the PetFinder dataset.



```
input_features:
- name: Image
  type: image
- name: Type
  type: category
- name: Name
  type: text
- name: Age
  type: numerical

output_features:
- name: AdoptionSpeed
  type: category

training:
  epochs: 10
  batch_size: 32
```

Figure 3: An example Ludwig configuration for the PetFinder dataset. The input and output features are listed together with some additional training information.

2.2 Limitations of the data-first AI approach

Although the data-first AI approach with its declarative nature simplifies the creation and training of ML models, a few challenges persist. First, despite reducing the required coding skills, some degree of coding is still necessary. Second, installing the framework and setting it up to work on specialized hardware, such as GPUs, requires a certain level of ongoing technical expertise, which is typically provided as a service deliverable. Third, some data-first approaches already provide a high level of transparency by incorporating open source principles and providing additional materials, such as explanatory documentation. This inherent transparency fosters user trust, so it is critical to maintain or improve this level of openness to maintain their confidence in the system. In figure 1, we have illustrated our approach to tackle these challenges and shift from data-first to human-first AI. In the upcoming sections, we will describe this in detail.

3 Simplified Terminology for Human-first AI

In order to enhance the accessibility of AI, we consulted with professionals in technical fields who have limited or no expertise in AI but could benefit from its usage. Our initial finding was that the terminology poses a barrier. This means that technicians are unfamiliar with concepts like input and output features, preprocessing, learning rate, and batch size.

Therefore, our first step was to develop a simplified vocabulary to discuss how to address the outstanding challenges of the data-first approach. In Figure 4, we present this user-friendly language.

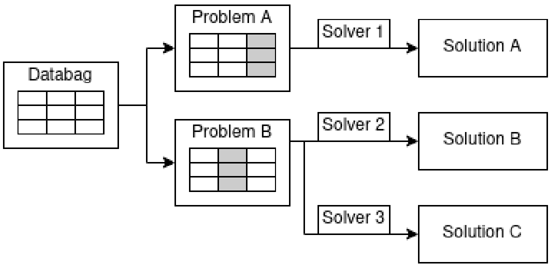


Figure 4: Simplified terminology to model AI problems: A Databag holds some multi-modal data. By specifing what you want to predict you create a Problem. A Solver is an abstraction of an ML algorithm and is used to create a Solution for the Problem.

A Databag is an abstraction of data and its associated metadata, which we suggest representing using a multimodal data format, similar to a table or dataframe (e.g. Pandas). In this format, each column corresponds to a specific datatype, and each row represents an entry for that column. As illustrated in Figure 2, the PetFinder-Dataset is an excellent example of such a multi-modal dataset with various columns, including images, numerical, categorical, and string values. However, this concept can also be applied to represent datasets with fewer columns, such as image classification datasets, where only two columns typically exist, the image and a corresponding categorical label. The multi-modal approach offers a great abstaction to deal with very different types of data.

To define a Problem on a Databag, one or more columns can be specified for prediction. For instance, in an image classification dataset, the label column would be the output of a typical image classification task. Similarly, in the PetFinder-Dataset, Adoption-Speed, represented as a categorical value, would be the output for a classification task. However, if Adoption-Speed were represented as the actual days it takes for a pet to be adopted, the Problem would be a regression one. Although single-output Problems are more common, the ability to specify multiple outputs enables us to model more complex Problems, such as an image classification with two independent labels. With a textual Databag, can represent various NLP tasks, including machine translation and sentiment analysis. In essence, this straightforward abstraction enables us to model a wide range of ML problems without needing specialized terminology for each.

The Solver serves as an abstraction layer for the ML algorithm that is used to tackle a Problem. It is responsible for abstracting all the complex elements that are important for the internals of the specific ML algorithm, such as learning rate and batch size, but are not relevant for its usage. Its main function is to create and train an ML model based on the data provided in the Databag and the defined Problem.

Once the ML model is trained, the resulting data including the trained model and relevant metrics, are stored in a Solution. This Solution can then be used to predict the output columns of the Problem for unseen data.

By using this terminology, users can benefit from a user-friendly interface to utilize AI technology without requiring a deep understanding of the internal workings of the Solver. At the same time, ML experts can still maintain the necessary flexibility to design and optimize ML algorithms according to specific needs and constraints.

4 Three Pillars of Human-first AI

With this simplified terminology, we can now delve into how to transition from a data-first approach to a human-first approach in AI. Figure 1 illustrates three pillars to achieve this goal.

4.1 No-Code ML

The first pillar aims to shift from a low-code approach to a no-code approach by providing a user-friendly interface that leverages declarative ML frameworks. By doing so, users will be able to create and deploy ML models without the need to write any code.

To achieve this goal, the first step is to create a Databag. The user interface should support the uploading of data in various formats, including excel, csv, or zip files. Once uploaded, the system should automatically inspect the data to determine the appropriate data types for each column. This information can then be stored in the Databag for later use.

To define a Problem on this Databag, the user selects the desired output columns. Based on this the system can then choose an appropriate Solver, such as a Solver based on Ludwig (Molino, Dudin & Miryala, 2019), which can be integrated seamlessly. The input and output sections for the Ludwig Solver can be derived from the multi-modal data definition of the Databag and the associated Problem.

The parameters of the training algorithms, also known as hyperparameters, can be set to appropriate default values or optimized using more expensive hyperparameter optimization strategies, depending on the available computing power. Examples of such hyperparameters include learning rate, batch size and number of hidden layers in a neural network.

Hyperparameter optimization (HPO) requires less human interaction and knowledge, but more computational power (Yu & Zhu, 2020). HPO techniques can range from simple examples such as random and grid search to more complex algorithms such as Bayesian optimization (Mockus, 1972) and Asynchronous Successive Halving Algorithm (Li et al., 2020), which have shown better performance.

This no-code ML solution enables users without coding skills or technical knowledge to benefit from AI technology, increasing its accessibility.

4.2 AI as a Service

We have demonstrated the feasibility of creating a no-code AI system using a data-first approach and simplified terminology. However, we recognize that the setup of such a system may still require expert knowledge. To address this challenge, we propose the development of a human-first AI platform that offers no-code ML as a service, with a user interface that abstracts away the underlying hardware.

To enable scalability, the platform must be deployed in the cloud. However, we understand that data privacy and security concerns may arise. To address these concerns, we suggest designing the platform in a way that allows users to host a private instance of the platform. This approach provides greater control over their data and ensures that sensitive information is not accessible to unauthorized parties. This can be achieved simply by making the deployment scripts for the platform readily available to users.

By implementing a hybrid approach that includes a public instance and an option for users to self-host the platform, we can address these two critical issues.

4.3 Transparency

In the previous section, we emphasized the importance of making the platform deployment accessible to users. However, this is just the first step towards transparency and trust in the technology. Transparency goes beyond deployment and includes making the entire platform accessible and understandable to users, independent experts, and other stakeholders. This is particularly important as artificial intelligence is often perceived as an incomprehensible and esoteric technology by non-experts.

By publicly sharing the platform's source code, we provide independent experts with the ability to scrutinize and monitor its internal workings, which enhances the platform's credibility. Furthermore, it allows users to gain a deeper understanding of the technology's mechanisms, leading to increased adoption and better outcomes. Thus, open-sourcing the platform is a fundamental step towards fostering widespread acceptance and trust in the technology.

In addition to the benefits of increased transparency and trust, open-sourcing the platform also facilitates collaboration among developers and the wider community. By making the source code available to others, developers can work together to identify and fix issues, implement new features, and enhance the overall functionality of the platform. This collaboration can result in more rapid development and deployment of new features, which in turn can lead to a more robust and adaptable platform. Therefore, open-sourcing the platform not only fosters trust and understanding, but also encourages innovation and community-driven improvement, making it a vital step towards creating a truly successful and impactful technology.

In the quest to increase trust and transparency in the technology, open-sourcing the platform is an important first step. In addition, utilizing explainable AI represents another promising approach. By leveraging explainable AI, the internal workings of ML models become more understandable to users, resulting in a greater level of transparency. This, in turn, can increase user confidence in the technology and encourage adoption. Therefore, incorporating explainable AI into the platform's development improves the overall trustworthiness and user-friendliness of the technology.

5 Comparison of existing AutoML Tools

In table 1 you can see a comparison of existing AutoML tools (Calefato et al., 2023). We have updated the entry for Ludwig (XAI features are already available in the codebase) and added additional rows for Orange Data Mining, KNIME Analytics Platform and Os4ML.

Automated machine learning (AutoML) aims to streamline the entire ML pipeline, encompassing tasks

Tool	Infrastruct.	Solution	Data types	Preparation		Analysis			Dissemination		XAI
				Data clean.	Data label.	Feat. engin.	Model train.	Model eval.	Model deploy.	Model monitor.	
AutoGluon*	On prem.	API	img, tab, txt	Y	N	Y	Y	Y	N	N	N
AutoGoal*	On prem.	API, CLI	img, tab, txt	N?	N	Y?	Y	Y	N	N	N
AutoKeras*	On prem.	API	img, tab, txt	Y?	N	Y?	Y	Y	N	N	N
Amazon SageMaker AutoPilot	Cloud	API, Web	img, tab, txt	Y	Y	Y	Y	Y	Y	Y	Y
BigML	Cloud	API, Web	img, tab, txt	Y	N	Y	Y	Y	Y	Y	Y
DataRobot AI Cloud	Cloud	API, Web	img, tab, txt	Y	Y	Y	Y	Y	Y	Y	Y
Google Vertex AI	Cloud	API, Web	img, tab, txt	Y	Y	Y	Y	Y	Y	Y	Y
H20 Driverless AI	Cloud, on prem.	API, Web	img, tab, txt	Y	Y	Y	Y	Y	Y	Y	Y
IBM Watson AutoAI	Cloud	API, Web	img, tab, txt	Y	N	Y	Y	Y	Y	Y	Y
Ludwig AI AutoML*	On prem.	API, CLI	img, tab, txt	Y?	N	Y	Y	Y	Y	N	Y?
MS Azure AutoML	Cloud	API, Web	img, tab, txt	Y	Y	Y	Y	Y	Y	Y	Y
Rapid Miner Studio	On prem.	Desktop	img, tab, txt	Y	N	Y	Y	Y	Y	Y	N
Orange Data Mining*	On prem.	API, Desktop	img, tab, txt	Y	N	Y	Y	Y	N	N	Y
KNIME Analytics Platform*	On prem.	Desktop	img, tab, txt	Y	N	Y	Y	Y	N	N	Y
Os4ML*	Cloud, on prem.	API, Web	img, tab, txt	Y	(Y)	Y	Y	Y	(Y)	(Y)	(Y)

Table 1: Table is taken from (Calefato et al., 2023), the Ludwig entry is updated and the Orange Data Mining, KNIME Analytics Platform and Os4ML entries are added. All open source tools are marked with a * behind the name in the first column. Y? means that the feature was present in the code, but not in the documentation. N? means that is was present in the documentation, but not in the code. Features, that are currently under development, are marked with (Y).

like data preparation, data analysis, and model generation (He et al., 2021). However, unlike the human-first AI approach, its primary focus is not merely on reducing the technical expertise required to use these tools. Contrary to the notion that "automated" implies a no-code solution like the introduced human-first AI approach, AutoML tools can still demand considerable technical expertise to set up and operate. Take Ludwig for example, as mentioned in section 2.1, it simplifies certain aspects of the ML pipeline, but users still need significant technical knowledge to utilize its API or CLI interface effectively.

Therefore, it is essential to recognize that human-first AI encompasses more than just AutoML. Besides automating the ML pipeline, human-first AI is dedicated to making the entire process accessible through a simple, no-code interface that can be accessed as a service.

To our knowledge, Os4ML is currently the only tool that provides a no-code web interface, supports both cloud and on-premise infrastructures, and remains open source. This unique combination makes Os4ML an outstanding solution in the field of human-first AI. In the following section we will look at the details of Os4ML.

6 The Open space for Machine Learning Platform

We are excited to provide an update on the progress of the Open space for Machine Learning (Os4ML) platform, which represents our commitment to human-first AI. We have achieved significant milestones in the development of the platform, including the successful implementation of a frontend with our proposed terminology and the integration of a Ludwig Solver, as outlined in section 4.1.

To illustrate, let's consider the PetFinder Dataset. Creating a Databag for this dataset is a seamless process. All you need to do is upload your data in one of our supported formats, such as an excel file with tabular data and a zip file containing images, onto our system. Our platform automatically detects the datatype of each column, although you can adjust the selection if necessary. Once you've selected the output column, simply choose a Solver (currently, only the Ludwig Solver is available, but we plan to add more options in the future) and let the platform train your ML model.

Once the training process is finished, you have the freedom to either download the model in the onnx format ("Open Neural Network Exchange", 2023) or directly upload new, unseen data for prediction. As we look ahead, we are actively enhancing the platform's capabilities to allow you to serve the model and grant straightforward access to it via API. This feature will enable you to seamlessly incorporate the model into your current systems, unlocking its full potential for your business or organization.

Additionally, we are proud to offer our platform to the public through a public instance[1] than can handle the high scalability demands of a cutting-edge AI platform. Moreover, in line with our commitment to transparency and openness, we have made the platform's deployment scripts publicly available along with the complete codebase of the platform[2]. These efforts reflect our dedication to create a trustworthy and accessible platform, and we look forward to continue to build on this initial progress to achieve our goals.

Although the integration of explainable AI into the human-first AI approach holds great promise for increasing transparency and trust in the technology, it is still an area that requires further research and development. We acknowledge that there is still much work to be done in this regard, and we remain committed to explore this avenue in future efforts. We will continue to report on our progress as we work towards incorporating explainable AI into our platform, with the ultimate goal of creating a technology that is both trustworthy and user-friendly.

[1] https://www.os4ml.com/

[2] https://github.com/WOGRA-AG/Os4ML

7 A Vision for the Future: Chatbot Interface

The emergence of large language models and chat systems (Fan et al., 2023), such as the GPT models (Eloundou et al., 2023), has opened up exciting possibilities for user interface design. We propose a vision for leveraging such systems as an interface to the Os4ML platform. However, it is important to acknowledge that this is a rapidly evolving field and the ideas presented here are intended to spark further discussion and exploration rather than to offer a definitive solution.

By leveraging a chat system, such as a standalone Os4ML chatbot or a plugin for an existing system like ChatGPT ("Chat Plugins", 2023), users can easily describe their data in natural language. The system would then automatically convert the data into a format compatible with Os4ML. If the user has not specified outputs, they would be prompted to do so. Once the Problem is defined, a Solver could be utilized to generate a Solution. The final model would be presented to the user in the same way as with traditional user interfaces, providing options to download the model, access it through API, or directly upload new data for evaluation.

A good starting point for this can be the GPT4All project, which provides a chatbot that is privacy aware and uses minimal computing ressources (Anand et al., 2023).

Utilizing a chat system as the user interface for the Os4ML platform offers several advantages over a traditional interface. Firstly, users can input information using natural language through various methods such as typing or speech-to-text tools like whisper (Radford et al., 2022), making the platform more accessible to users with varying preferences and abilities. Secondly, the chat system can automatically format data to the Os4ML standard, relieving users from the need to learn how to format data themselves.

Moreover, the simplicity of natural language makes it one of the easiest user interfaces, ensuring that the platform is user-friendly and easy to use for everyone. An additional benefit of using a chat system is that users can receive immediate answers to their questions, eliminating the need to search through documentation and enabling them to obtain the information they need quickly and efficiently.

In summary, the chat system streamlines the interface, making it more flexible and accessible for all users, while also simplifying the process of data formatting and enabling users to receive real-time support.

8 Summary & Outlook

In this article, we have demonstrated how the principles of data-first AI can be leveraged to create human-first AI. Our approach involved building a no-code ML solution around the data-first AI, offering it as a service, and making the codebase open source to promote transparency and collaboration. The Open space for Machine Learning platform represents a practical example of our approach in action.

As we look ahead, our focus remains steadfast on the continued development of our platform, with the goal of making AI more accessible to a broader range of users. We are thrilled to witness how users utilize our platform and how this influences their work. Through our efforts, we strive to empower individuals with the tools they need to realize their full potential and achieve their goals.

Acknowledgments

Os4ML is a project of the WOGRA AG research group in cooperation with the German Aerospace Center and is funded by the Ministry of Economic Affairs, Regional Development and Energy as part of the High Tech Agenda of the Free State of Bavaria.

References

Allen, B., Agarwal, S., Kalpathy-Cramer, J. & Dreyer, K. (2019) Democratizing AI. *Journal Of The American College Of Radiology*, 961-963

Anand, Y., Nussbaum, Z., Duderstadt, B., Schmidt, B. & Mulyar, A. (2023) *GPT4All: Training an Assistant-style Chatbot with Large Scale Data Distillation from GPT-3.5-Turbo.* https://github.com/nomic-ai/gpt4all

Cabot, J. (2020) Positioning of the low-code movement within the field of model-driven engineering. *Proceedings Of The 23rd ACM/IEEE International Conference On Model Driven Engineering Languages And Systems: Companion Proceedings*, https://doi.org/10.1145/3417990.3420210

Calefato, F., Quaranta, L., Lanubile, F. & Kalinowski, M. (2023) *Assessing the Use of AutoML for Data-Driven Software Engineering.*

Chat Plugins (2023). Retrieved from https://platform.openai.com/docs/plugins/

Eloundou, T., Manning, S., Mishkin, P. & Rock, D. *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models.* (2023)

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M. & Smola, A. *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.* (2020)

Fan, L., Li, L., Ma, Z., Lee, S., Yu, H. & Hemphill, L. A *Bibliometric Review of Large Language Models Research from 2017 to 2023.* (2023)

He, X., Zhao, K. & Chu, X. (2021) *AutoML: A survey of the state-of-the-art. Knowledge-Based Systems* 212 pp. 106622, https://doi.org/10.1016%252Fj.knosys.2020.106622

Howard, J. & Gugger, S. (2020) Fastai: A Layered API for Deep Learning. *Information*, 11, 108, https://doi.org/10.3390

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B. & Talwalkar, A. *A System for Massively Parallel Hyperparameter Tuning.* (2020)

Mockus, J. (1972) On Bayes methods for seeking an extremum. *Avtomatika I Vychislitelnaja Technika.*, 3, pp. 53-62

Molino, P., Dudin, Y. & Miryala, S. *Ludwig: a type-based declarative deep learning toolbox.* (2019)

Molino, P. & Re, C. *Declarative Machine Learning Systems.* (2021)

Open Neural Network Exchange (2023). Retrieved from https://onnx.ai/

Radford, A., Kim, J., Xu, T., Brockman, G., McLeavey, C. & Sutskever, I. *Robust Speech Recognition via Large-Scale Weak Supervision.* (2022)

Re, C. (2020) Overton: A Data System for Monitoring and Improving Machine-Learned Products. *10th Conference On Innovative Data Systems Research*, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings., http://cidrdb.org/cidr2020/papers/p33-re-cidr20.pdf

Shi, X., Mueller, J., Erickson, N., Li, M. & Smola, A. *Benchmarking Multimodal AutoML for Tabular Data with Text Fields.* (2021)

The AI Index Report 2023 (2023). Retrieved from https://aiindex.stanford.edu/report/

Whang, S., Roh, Y., Song, H. & Lee, J. (2023) Data collection and quality challenges in deep learning: a data-centric AI perspective. *The VLDB Journal*, https://doi.org/10.1007/s00778-022-00775-9

Yu, T. & Zhu, H. *Hyper-Parameter Optimization: A Review of Algorithms and Applications.* (2020)

Zha, D., Bhat, Z., Lai, K., Yang, F. & Hu, X. *Data-centric AI: Perspectives and Challenges.* (2023)

Zha, D., Bhat, Z., Lai, K., Yang, F., Jiang, Z., Zhong, S. & Hu, X. *Data-centric Artificial Intelligence: A Survey.* (2023)

Zhang, S. & Zhang, K. *PetFinder Challenge: Predicting Pet Adoption Speed.* (2019)