

ARTICLE

# Analysis of AutoML Tools in the World of Automated Deep Learning

Liam Connor, Rebecca Kelly, Saoirse Murphy, Eoghan McCarthy,\* and Edward Murray

Walton Institute for Information and Communication Systems Science, Waterford Institute of Technology, X91 WR86 Waterford, Ireland

\*Corresponding author: eoghan.mc.carthy@waltoninstitute.ie

(Received: 15 September 2023; Revised: 29 November 2023; Accepted: 03 January 2024; Published: 14 January 2024)

## Abstract

In-depth analysis of AutoML (Automated Machine Learning) tools in the context of automated deep learning, a rapidly evolving field aimed at simplifying and optimizing the development of deep learning models. AutoML tools automate various stages of the machine learning pipeline, including data preprocessing, feature selection, model selection, hyperparameter tuning, and model evaluation, making them accessible to non-experts and enhancing productivity for experts. The article reviews prominent AutoML tools, examining their methodologies, strengths, limitations, and performance across different tasks. Through comparative studies and practical applications, the effectiveness of these tools in producing robust and efficient deep learning models is evaluated. Additionally, the article explores emerging trends and future directions in the integration of AutoML with deep learning, highlighting the potential for these tools to revolutionize machine learning workflows and democratize access to advanced AI technologies.

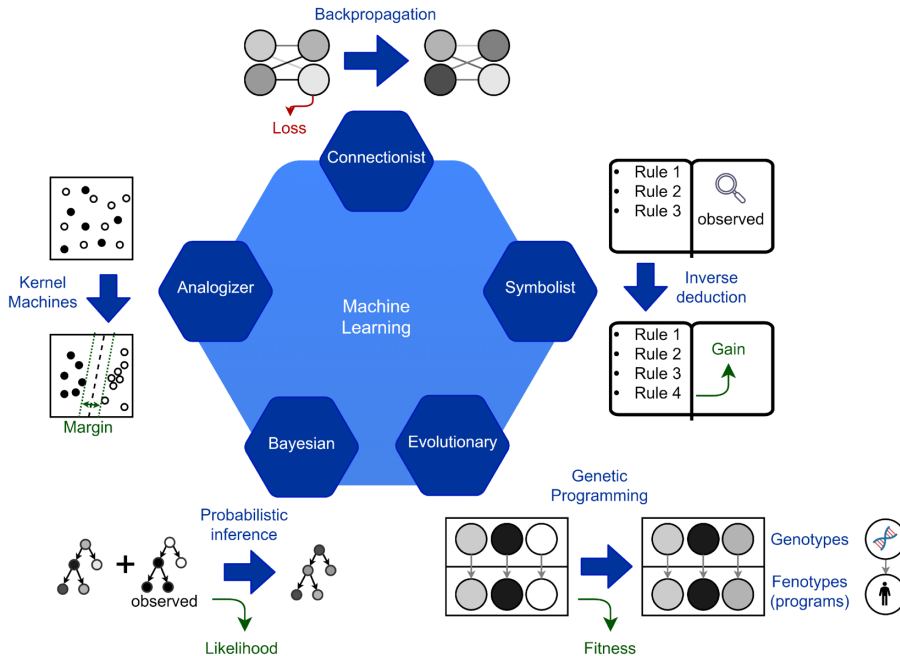
**Keywords:** AutoML; Automated deep learning; Data preprocessing; Hyperparameter tuning; Machine learning pipeline; Model selection; Performance evaluation

**Abbreviations:** AUC-ROC: Area under the receiver operating characteristic curve, AutoML: Automated machine learning, MSE: Mean squared error, MAE: Mean absolute error, NAS: Neural architecture search

## 1. Introduction

The field of automated machine learning (AutoML) has gained significant traction in recent years as a way to simplify the process of building and deploying machine learning models. AutoML tools and platforms aim to automate various tasks involved in the machine learning pipeline, such as data preprocessing, feature selection, model selection, hyperparameter tuning, and model evaluation. By reducing the manual effort required in these steps, AutoML empowers individuals and organizations, even those without extensive expertise in machine learning, to harness the power of data and develop accurate predictive models [1, 2, 3, 4, 5].

AutoML tools typically involve a comprehensive workflow that begins with data annotation and preparation, followed by training multiple machine learning algorithms in parallel. These tools then evaluate the performance of the trained models using appropriate metrics and select the best-performing model for deployment. The inherent goal is to streamline the model development process, improve efficiency, and accelerate the adoption of machine learning across various domains (Fig. 1) [6, 7, 8, 9].



**Figure 1.** The five paradigms of Machine Learning according to the classification proposed by Pedro Domingos.

Automated Machine Learning (AutoML) is a modern approach that leverages automation to streamline the application of machine learning models for real-world problems. It aims to automate the end-to-end process of building machine learning models, including data preprocessing, feature engineering, model selection, and hyperparameter tuning. The primary goal of AutoML is to make it easier for non-experts to develop machine learning models by providing a simple, user-friendly interface for training and deploying models. AutoML covers the following steps in the ML development process [10, 11, 12, 13, 14, 15, 16]:

1. **Data preprocessing:** Automated data cleaning, formatting, and transformation.
2. **Feature engineering and selection:** Automatic identification and selection of relevant features from the data.
3. **Algorithm selection:** Automated selection of appropriate machine learning algorithms based on the problem and data characteristics.
4. **Hyperparameter optimization and model selection:** Automatic tuning of model hyperparameters and selection of the best-performing model.
5. **Neural architecture search (NAS):** Automated design and optimization of neural network architectures for deep learning models.

Most AutoML software auto-analyzes data, selects algorithms, trains and tests models, and refines them through hyperparameter tuning to provide the best-performing model. AutoML software enables data scientists, machine learning engineers, and non-technical users to automatically build scalable machine learning models. It allows users to trade-off between model complexity and performance, enabling them to build complex high-performance models or simpler, more explainable models [17, 18].

AutoML tools and platforms aim to automate various tasks involved in the machine learning pipeline, such as data annotation, feature selection, model selection, hyperparameter tuning, and model eval-

uation. By reducing the manual effort required in these steps, AutoML tools empower individuals and organizations, even those without extensive expertise in data science and machine learning, to harness the power of data and develop accurate predictive models [19].

## 2. Key Components of AutoML

The key components of AutoML systems typically include (Fig. 1):

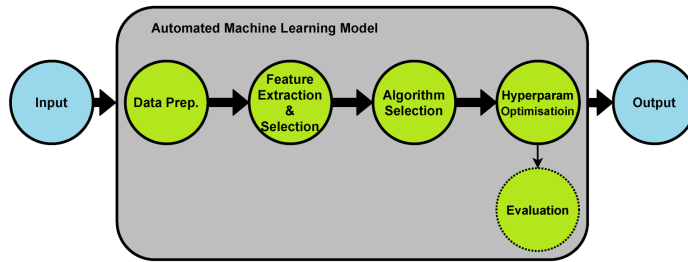


Figure 2. AutoML model main processes.

1. **Automated Data Preprocessing and Cleaning:** AutoML tools automate the process of cleaning and preprocessing data, which includes tasks such as handling missing values, removing outliers, and transforming data into a format suitable for machine learning algorithms.
2. **Automated Feature Selection and Engineering:** AutoML systems can automatically identify and select relevant features from the input data, as well as engineer new features through techniques like feature extraction, feature construction, and dimensionality reduction.
3. **Automated Model Selection:** This component involves the automatic selection of appropriate machine learning algorithms and models based on the characteristics of the data and the problem at hand. AutoML tools can train and evaluate multiple models in parallel, and select the best-performing model based on predefined metrics.
4. **Automated Hyperparameter Optimization:** AutoML tools can automate the process of tuning hyperparameters, which are configuration settings that control the behavior of machine learning algorithms. This is typically done through techniques like grid search, random search, or Bayesian optimization.
5. **Automated Neural Architecture Search (for Deep Learning):** In the context of deep learning, AutoML systems can automatically design and optimize neural network architectures, a process known as Neural Architecture Search (NAS). This involves searching for the optimal combination of layers, activation functions, and other architectural components.
6. **Automated Model Postprocessing and Analysis:** AutoML tools may also provide automated model analysis and interpretation, including techniques like feature importance analysis, model explanations, and performance evaluation on holdout or test data.

Other key components of AutoML systems include:

- **Hyperparameter Tuning across Algorithms:** AutoML helps standardize hyperparameter tuning across different algorithms, enabling fair comparisons and ensuring that each algorithm is optimized for the given task.
- **Automated Hyperparameter Optimization:** AutoML automates the tedious and error-prone process of hyperparameter optimization, which is crucial for achieving optimal model performance.
- **Automated Architecture Search:** For deep learning applications, AutoML enables faster devel-

opment by automating the process of neural architecture search, which can be time-consuming and resource-intensive when done manually.

It's important to note that the quality of the input data is critical, as it directly affects the accuracy and performance of the trained model. Additionally, key inputs, outputs, and hyperparameters are crucial components of the AutoML process. AutoML tools like Auto-PyTorch, AutoKeras, and talos provide architecture search and data annotation capabilities for deep learning models, further streamlining the development process [20, 21].

### 3. AutoML Platforms and Tools

The field of AutoML has seen the emergence of various platforms and tools that aim to simplify and automate the process of building machine learning models. These tools cater to a wide range of users, from data scientists and machine learning engineers to non-technical users, enabling them to leverage the power of automated machine learning techniques (Fig. 3) [22].

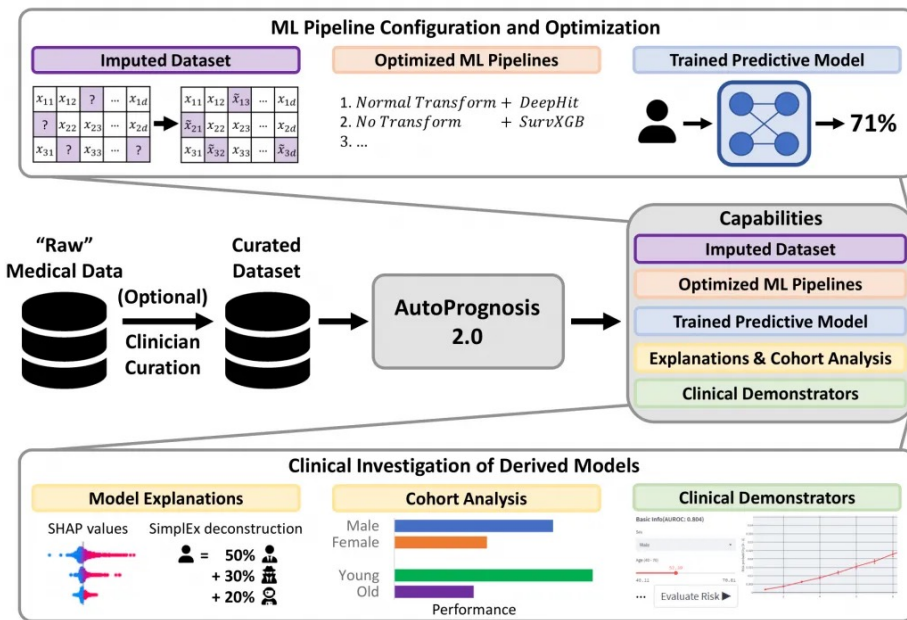


Figure 3. AutoPrognosis.

Here are some of the popular AutoML platforms and tools:

1. **Google Cloud AutoML:** Google’s cloud-based AutoML suite offers services for automating the training of machine learning models for tasks such as image recognition, text classification, and structured data analysis. It provides a user-friendly interface and requires minimal coding.
2. **Auto-Sklearn:** An open-source AutoML library for Python that automates the process of selecting and tuning machine learning algorithms for a given dataset. It supports a wide range of algorithms and can be easily integrated into existing workflows.
3. **AutoKeras:** Another open-source AutoML library for Python that focuses specifically on automating the design and training of deep neural networks. It supports various types of neural network architectures and can be used for tasks like image classification, text classification, and

regression.

4. **Amazon Lex:** Amazon’s AutoML service for building conversational interfaces, such as chatbots and virtual assistants. It handles tasks like automatic speech recognition, natural language understanding, and dialogue management.
5. **H2O AutoML:** An open-source AutoML platform that provides a comprehensive suite of tools for automating the entire machine learning workflow, from data preprocessing to model deployment. It supports a wide range of algorithms and can be used for various tasks, including regression, classification, and time-series forecasting.

Other notable AutoML platforms and tools include:

- AutoWEKA, Auto-PyTorch, AutoGluon, MLBoX, TPOT, and TransmogriAI (open-source)
- DataRobot, Dataiku, Enhancer, Akkio, MLJAR, dotData, JADBIO AutoML, and DMWay Analytics Engine (commercial)

These platforms offer various features and capabilities, such as automated data analysis, algorithm selection, model training, hyperparameter tuning, and model deployment. They aim to streamline the machine learning process and make it more accessible to a broader range of users as given in Table 1.

**Table 1.** Differences of Open-Source AutoML Tools and Commercial AutoML Platforms

Open-Source AutoML Tools	Commercial AutoML Platforms
<ul style="list-style-type: none"> <li>• AutoGluon</li> <li>• FLAML</li> <li>• H2O-3 AutoML</li> <li>• Auto-Sklearn</li> <li>• AutoKeras</li> <li>• MLBox</li> </ul>	<ul style="list-style-type: none"> <li>• DataRobot</li> <li>• Dataiku</li> <li>• H2O.ai</li> <li>• Google Cloud AutoML</li> <li>• Enhancer</li> <li>• Akkio</li> <li>• MLJAR</li> <li>• dotData</li> <li>• JADBIO AutoML</li> <li>• DMWay Analytics Engine</li> </ul>

It’s worth noting that the choice of an AutoML platform or tool depends on various factors, such as the specific requirements of the project, the level of expertise of the user, the type of data and problem being addressed, and the desired level of control and customization [23, 24, 25, 26, 27].

#### 4. Workflow for Developing AutoML Models

The typical workflow for developing AutoML models involves several key steps as given in Table 2:

1. **Data Ingestion and Preparation:** The user provides the input data to be used for training the model. The AutoML system then preprocesses the data, including feature engineering and normalization. Automated feature engineering (featurization) techniques are applied, which can also be customized based on the user’s requirements.
2. **Model Training:** The AutoML system trains multiple machine learning models using different algorithms and hyperparameters. This process involves creating parallel pipelines that iterate through various ML algorithms and feature selections to produce models with training scores.

AutoML supports ensemble models using voting and stacking methods, which can improve overall model performance.

3. **Model Evaluation:** The AutoML system evaluates the performance of the trained models using appropriate metrics. It supports splitting the data into training, validation, and test sets to ensure robust evaluation of model performance.
4. **Model Selection and Deployment:** Based on the evaluation results, the AutoML system selects the best-performing model. The selected model can then be deployed for making predictions on new data. AutoML-generated models can be converted to ONNX format, enabling deployment on various platforms and devices.

**Table 2.** Procedural steps

Step	Description
Data Ingestion and Preparation	<ul style="list-style-type: none"> <li>• User provides input data</li> <li>• Data preprocessing (feature engineering, normalization)</li> <li>• Automated feature engineering (featurization)</li> </ul>
Model Training	<ul style="list-style-type: none"> <li>• Training multiple ML models with different algorithms and hyperparameters</li> <li>• Parallel pipelines for iterating through algorithms and feature selections</li> <li>• Support for ensemble models (voting, stacking)</li> </ul>
Model Evaluation	<ul style="list-style-type: none"> <li>• Evaluation of trained models using appropriate metrics</li> <li>• Support for training, validation, and test data splits</li> </ul>
Model Selection and Deployment	<ul style="list-style-type: none"> <li>• Selection of best-performing model</li> <li>• Deployment for making predictions on new data</li> <li>• Support for ONNX format for cross-platform deployment</li> </ul>

**Table 3.** Data types

Data Type	Considerations
Tabular Data	Determine model objective (classification, regression, forecasting) Ensure data is properly formatted and labeled
Image Data	Minimum of 100 examples per category/label Capture variation in the problem space
Video Data	Choose appropriate objective (action recognition, classification, object tracking) Ensure data is properly formatted and labeled

For deep learning applications, the AutoML workflow typically involves data ingestion, data preparation, data engineering, model selection, model training, hyperparameter tuning, model deployment, and model updates. Platforms like Vertex AI provide an end-to-end workflow that includes gathering and preparing the data, training the model, evaluating the model, and deploying and using the model for predictions [28, 29, 30, 31].

## 5. Data Set Considerations

Data quality is a critical factor that directly impacts the accuracy and performance of AutoML models. Poor data quality or biased datasets can lead to inaccurate models that perpetuate existing biases,

undermining the effectiveness of AutoML solutions [17]. To mitigate this, it is essential to carefully prepare and preprocess the data before feeding it into an AutoML system as given in Table 3 [32].

Comparisons of the AUC score and training duration of the best model built using AutoML and non-AutoML frameworks. When preparing datasets for AutoML, it is crucial to assess the use case and determine the desired outcome. The following considerations should be taken into account [33, 34]:

1. **Labeled Data:** For supervised learning tasks, gather enough labeled examples per category or label, with a minimum of 1,000 examples per label. Distribute the examples equally across categories to avoid class imbalance.
2. **Capturing Variation:** Ensure that the dataset captures the variation in the problem space to enable better generalization. For image data, the bare minimum is 100 examples per category or label.
3. **Data Format and Objectives:**
  - For tabular data, determine the appropriate model objective (binary classification, multi-class classification, forecasting, regression).
  - For video data, choose the appropriate objective (action recognition, classification, object tracking).
4. **Data Formatting and Labeling:** Prepare the data properly and ensure it is formatted and labeled correctly. Inconsistent or incorrect labeling can lead to poor model performance.

AutoML performs well for structured data, small to medium datasets, and when a quick assessment of the model is needed. However, it may not be the best choice for larger and more complex datasets, as it can be resource-intensive and slow. Key considerations when using AutoML include iteration speed, data requirements, cost, and use case suitability [35, 36, 37, 38].

## 6. Model Training and Optimization

Optimizing hyperparameters, which control the behavior of machine learning models, is a crucial part of the AutoML process [2]. AutoML techniques like hyperparameter optimization and algorithm configuration help automate the tedious process of tuning models for new tasks, improving the reproducibility and fairness of ML research. By automating hyperparameter tuning and algorithm configuration, AutoML enables more rigorous comparisons between techniques, advancing ML research as given in Table 4 [39, 40, 41].

AutoML tools like Vertex AI use supervised learning to train models on labeled example data (Fig. 4). The specific algorithm and training method depend on the data type and use case. For effective data preparation, Vertex AI recommends [42, 43]:

1. Assessing the use case and desired outcome.
2. Gathering enough labeled examples, with a minimum of 1,000 examples per label for supervised learning tasks.
3. Distributing examples evenly across categories to avoid class imbalance.
4. Capturing the diversity and variation in the problem space.

Vertex AI provides AutoML capabilities to automatically train and optimize models, as well as support for custom training, allowing users to bring their own training code and containers.

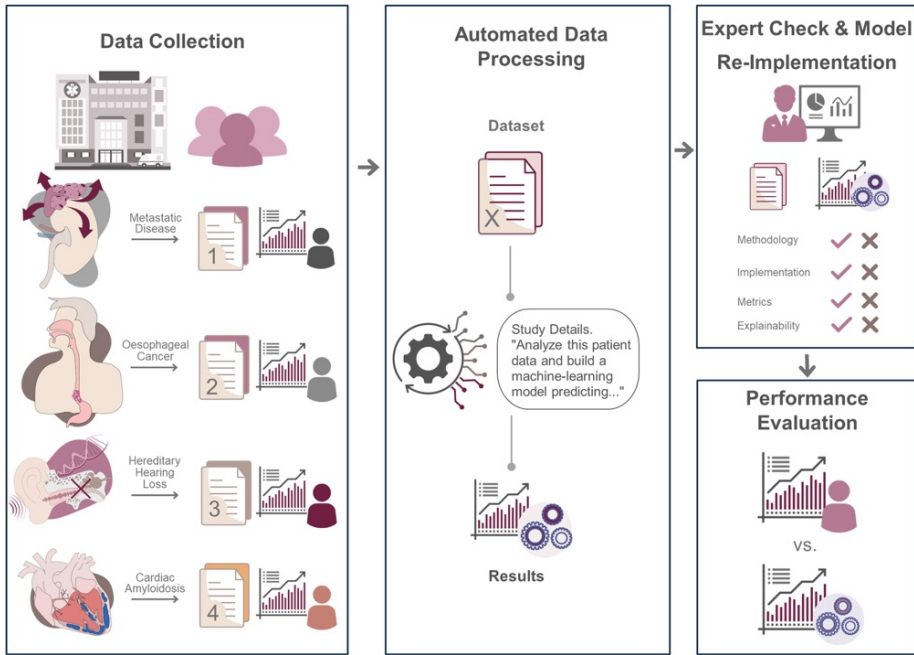


Figure 4. Study design.

Table 4. Data preparation

Data Preparation	Recommendations
Use Case Assessment	Determine desired outcome
Labeled Examples	Minimum 1,000 examples per label
Class Distribution	Even distribution across categories
Variation Capture	Capture diversity in the problem space

Google Cloud Platform’s AutoML Vision is a tool that allows users to train custom machine learning models without extensive ML expertise. The training process involves uploading the dataset, labeling the images, and initiating the training process. AutoML Vision automatically evaluates the trained model’s performance and provides metrics like precision and recall. Users can further optimize the model by adjusting the score threshold, and AutoML Vision automatically sets aside a portion of the data for testing the trained model [44, 45, 46].

AutoML tools can provide better-performing models by iterating through different models and performing hyperparameter optimization. This automated process of model training and optimization can significantly improve the performance and accuracy of machine learning models, making AutoML tools a valuable asset for individuals and organizations seeking to leverage the power of data and machine learning [47, 48, 49].

## 7. Performance Evaluation and Interpretation

Evaluating the performance of AutoML models is a crucial step in the development process. Vertex AI provides a comprehensive set of tools to assess model performance, enabling users to make

informed decisions about model deployment and optimization. Vertex AI supports various evaluation metrics, depending on the task and data type. For classification tasks, common metrics include precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). For regression tasks, metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared are typically used. These metrics help quantify the model’s accuracy, generalization ability, and overall performance as given in Table 5 [50, 51, 52].

Table 5. Evaluation Metrics

Feature Engineering Tools	Description
Classification	Precision, Recall, F1-score, AUC-ROC
Regression	Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared

Classification Precision, Recall, F1-score, AUC-ROC Regression Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared

In addition to these standard metrics, Vertex AI also provides advanced capabilities for model evaluation, such as:

- **Model Monitoring:** Continuous monitoring of deployed models to detect performance degradation or data drift, enabling proactive maintenance and retraining.
- **Explainability:** Tools for interpreting and understanding the behavior of machine learning models, including feature importance analysis and model explanations.
- **Pipeline Orchestration:** Automated end-to-end pipelines for model training, evaluation, and deployment, ensuring consistency and reproducibility [53, 54].

Once the model has been evaluated and deemed satisfactory, Vertex AI allows for seamless deployment to various environments, including online (for real-time predictions) and batch (for large-scale predictions on historical data). This streamlined process ensures that the best-performing model is readily available for production use, enabling organizations to leverage the power of AutoML effectively [55, 56].

## 8. Potential Applications

AutoML is fundamentally transforming how businesses and organizations operate by automating the intricate tasks involved in machine learning model development. As data volumes continue to surge, the demand for efficient, accurate, and scalable machine learning solutions will grow, further amplifying the relevance of AutoML. AutoML is poised to play a pivotal role in shaping the future trajectory of machine learning by democratizing the field and catalyzing innovation across various sectors (Fig. 1) [57].

AutoML can be leveraged by software engineers, data scientists, ML engineers, and AI enthusiasts to rapidly build and deploy ML models for a wide array of applications, including [58, 59, 60, 61]:

- **Image Classification:** Automating the process of categorizing and labeling images, useful for applications like quality control in manufacturing and medical image analysis.
- **Customer Churn Prediction:** Building models to predict customer attrition, enabling proactive retention strategies.
- **Fraud Detection:** Developing models to identify fraudulent activities in domains like finance and e-commerce.

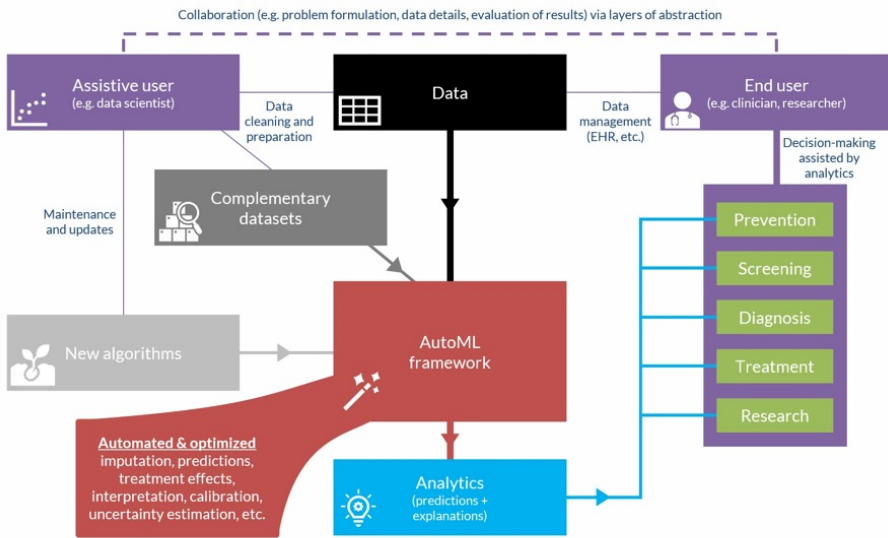


Figure 5. AutoML in healthcare is commonization of components.

- **Personalization:** Creating personalized recommendations for products, content, or services based on user preferences and behavior.
- **Anomaly Detection:** Identifying unusual patterns or outliers in data, applicable in areas like cybersecurity, predictive maintenance, and financial monitoring.

AutoML supports various learning paradigms, including supervised learning (regression and classification), unsupervised learning, and reinforcement learning, across diverse data modalities like structured data, text, images, sound, and temporal/multimodal data as given in Table 6 [62, 63].

Table 6. Learning Paradigms

Learning Paradigm	Data Modalities
Supervised Learning (Regression, Classification)	Structured Data, Text, Images
Unsupervised Learning	Sound, Temporal/Multimodal Data
Reinforcement Learning	-

Moreover, AutoML can empower ‘citizen data scientists’ – domain experts who can leverage the power of machine learning to automate and support their daily tasks without needing extensive machine learning expertise. This democratization of machine learning can foster innovation and enable organizations to extract valuable insights from their data more efficiently [64].

AutoML offers several advantages over traditional manual model development approaches:

1. **Cost-Efficiency:** AutoML reduces the costs associated with skilled employees, services, and infrastructure required for end-to-end machine learning model development.
2. **Accessibility:** AutoML lowers the barrier to entry for companies new to deploying AI/ML projects, enabling them to leverage machine learning capabilities more easily.

3. **Time-Saving:** AutoML significantly reduces the time and resources needed to create machine learning models compared to traditional development methods.
4. **Accuracy:** AutoML leverages state-of-the-art algorithms and techniques to produce more accurate predictions and insights.
5. **Scalability:** AutoML enables scalable use of machine learning models across a range of applications and data sets.
6. **Interpretability:** AutoML integrates features that help users understand and justify model predictions, important for regulatory compliance and trust in AI systems.

## 9. Conclusion

The field of AutoML is revolutionizing how organizations approach machine learning model development. By automating the intricate processes involved, AutoML empowers individuals and businesses to harness the power of data and machine learning more efficiently. It simplifies the model development lifecycle, reduces costs, and accelerates time-to-deployment, making machine learning accessible to a broader audience. As data volumes continue to soar, the demand for scalable and accurate machine learning solutions will surge. AutoML, with its ability to automate tasks like data preprocessing, feature selection, model training, and hyperparameter tuning, is poised to play a pivotal role in shaping the future of machine learning. By democratizing the field and catalyzing innovation across various sectors, AutoML holds immense potential for driving transformative applications that can unlock valuable insights and drive better decision-making.

## References

- [1] Georgios Tournavitis, Zheng Wang, Björn Franke, and Michael FP O'Boyle. "Towards a holistic approach to auto-parallelization: integrating profile-driven parallelism detection and machine-learning based mapping". In: *ACM Sigplan notices* 44.6 (2009), pp. 177–187.
- [2] Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. "Evaluation of a tree-based pipeline optimization tool for automating data science". In: *Proceedings of the genetic and evolutionary computation conference 2016*. 2016, pp. 485–492.
- [3] Kasthurirangan Gopalakrishnan. "Deep learning in data-driven pavement image analysis and automated distress detection: A review". In: *Data* 3.3 (2018), p. 28.
- [4] Apapan Pumsirirat and Yan Liu. "Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine". In: *International Journal of advanced computer science and applications* 9.1 (2018).
- [5] ChoHee Kim, Donghyun Gwan, and Minh Sena Nam. "Beyond the Atmosphere: The Revolution in Hypersonic Flight". In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 2.1 (2021), pp. 152–163.
- [6] Nikhil Ketkar and Eder Santana. *Deep learning with Python*. Vol. 1. Springer, 2017.
- [7] Iqbal H Sarker. "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions". In: *SN computer science* 2.6 (2021), p. 420.
- [8] Tharindu Kaluarachchi, Andrew Reis, and Suranga Nanayakkara. "A review of recent deep learning approaches in human-centered machine learning". In: *Sensors* 21.7 (2021), p. 2514.
- [9] Darren Cook. *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI*. " O'Reilly Media, Inc.", 2016.
- [10] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. "A survey of deep learning and its applications: a new paradigm to machine learning". In: *Archives of Computational Methods in Engineering* 27 (2020), pp. 1071–1092.

- [11] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. “Deeptest: Automated testing of deep-neural-network-driven autonomous cars”. In: *Proceedings of the 40th international conference on software engineering*. 2018, pp. 303–314.
- [12] Adrichem De Jong, Mark Jansen, Jeffrey Van Dijk, and Johannes Meyer. “Analysis of Innovative Practices in Advanced Materials and Structural Engineering”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 2.1 (2021), pp. 178–188.
- [13] Giang Nguyen, Stefan Dlugolinsky, Martin Bobák, Viet Tran, Álvaro López García, Ignacio Heredia, Peter Malík, and Ladislav Hluchý. “Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey”. In: *Artificial Intelligence Review* 52 (2019), pp. 77–124.
- [14] William Grant Hatcher and Wei Yu. “A survey of deep learning: Platforms, applications and emerging research trends”. In: *IEEE access* 6 (2018), pp. 24411–24432.
- [15] Hamza Harkous, Kassem Fawaz, Rémi Lebre, Florian Schaub, Kang G Shin, and Karl Aberer. “Polisis: Automated analysis and presentation of privacy policies using deep learning”. In: *27th USENIX Security Symposium (USENIX Security 18)*. 2018, pp. 531–548.
- [16] Trang T Le, Weixuan Fu, and Jason H Moore. “Scaling tree-based automated machine learning to biomedical big data with a feature set selector”. In: *Bioinformatics* 36.1 (2020), pp. 250–256.
- [17] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. “Deepxplore: Automated whitebox testing of deep learning systems”. In: *proceedings of the 26th Symposium on Operating Systems Principles*. 2017, pp. 1–18.
- [18] Theodore Elijah, James Clarence, Benjamin Anthony, and Christopher William. “The Journey and Potential of Organ-on-a-Chip Technology”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 2.2 (2021), pp. 211–223.
- [19] Randal S Olson and Jason H Moore. “TPOT: A tree-based pipeline optimization tool for automating machine learning”. In: *Workshop on automatic machine learning*. PMLR. 2016, pp. 66–74.
- [20] Xin He, Kaiyong Zhao, and Xiaowen Chu. “AutoML: A survey of the state-of-the-art”. In: *Knowledge-based systems* 212 (2021), p. 106622.
- [21] Marc-André Zöllner and Marco F Huber. “Benchmark and survey of automated machine learning frameworks”. In: *Journal of artificial intelligence research* 70 (2021), pp. 409–472.
- [22] Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. “Taking human out of learning applications: A survey on automated machine learning”. In: *arXiv preprint arXiv:1810.13306* 31 (2018).
- [23] Marc Heylen, Patrick Bossuyt, Philippe Provoost, David Borremans, and Christine Rampelberg. “Making Antennas for 6G”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 3.1 (2022), pp. 235–247.
- [24] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.
- [25] Zhuowen Tu and Xiang Bai. “Auto-context and its application to high-level vision tasks and 3D brain image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.10 (2009), pp. 1744–1757.
- [26] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. “Deep learning for smart manufacturing: Methods and applications”. In: *Journal of manufacturing systems* 48 (2018), pp. 144–156.
- [27] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D Lawrence. “Challenges in deploying machine learning: a survey of case studies”. In: *ACM computing surveys* 55.6 (2022), pp. 1–29.
- [28] Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu, S Yu Philip, and Weixiong Zhang. “A survey of community detection approaches: From statistical modeling to deep learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.2 (2021), pp. 1149–1170.

- [29] Tania Lorido-Botran, Jose Miguel-Alonso, and Jose A Lozano. "A review of auto-scaling techniques for elastic applications in cloud environments". In: *Journal of grid computing* 12 (2014), pp. 559–592.
- [30] Valdemar Johansen, Malthe Rasmussen, and Arne Knudsen. "Dielectric Constants and Their Role in Plasma Simulation". In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 3.1 (2022), pp. 248–260.
- [31] Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. "Multilingual subjectivity analysis using machine translation". In: *Proceedings of the 2008 conference on empirical methods in natural language processing*. 2008, pp. 127–135.
- [32] Matthias Feurer and Frank Hutter. "Hyperparameter optimization". In: *Automated machine learning: Methods, systems, challenges* (2019), pp. 3–33.
- [33] Jiajia Li, Guangming Tan, Mingyu Chen, and Ninghui Sun. "SMAT: An input adaptive auto-tuner for sparse matrix-vector multiplication". In: *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation*. 2013, pp. 117–126.
- [34] GSDMA Sreenu and Saleem Durai. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis". In: *Journal of Big Data* 6.1 (2019), pp. 1–27.
- [35] Josef Baumgartner, Alexandra Schneider, Ulugbek Zhenis, Franz Jager, and Josef Winkler. "Mastering Neural Network Prediction for Enhanced System Reliability". In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 3.1 (2022), pp. 261–274.
- [36] Mary Lynn Hamilton, Laura Smith, and Kristen Worthington. "Fitting the methodology with the research: An exploration of narrative, self-study and auto-ethnography". In: *Studying teacher education* 4.1 (2008), pp. 17–28.
- [37] Haifeng Jin, Qingquan Song, and Xia Hu. "Auto-keras: An efficient neural architecture search system". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 1946–1956.
- [38] Fei Shan, Yaozong Gao, Jun Wang, Weiya Shi, Nannan Shi, Miaofei Han, Zhong Xue, Dinggang Shen, and Yuxin Shi. "Lung infection quantification of COVID-19 in CT images with deep learning". In: *arXiv preprint arXiv:2003.04655* (2020).
- [39] Sheng Li, Jaya Kawale, and Yun Fu. "Deep collaborative filtering via marginalized denoising auto-encoder". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 811–820.
- [40] Long Wen, Liang Gao, and Xinyu Li. "A new deep transfer learning based on sparse auto-encoder for fault diagnosis". In: *IEEE Transactions on systems, man, and cybernetics: systems* 49.1 (2017), pp. 136–144.
- [41] William Coetzee, Reiner Khumalo, Brendan Le Roux, and Ebrahim Van Wyk. "Sickle Cell Disease: Causes, Symptoms, and Treatment". In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 3.1 (2022), pp. 275–286.
- [42] Shengfeng Liu, Yi Wang, Xin Yang, Baiying Lei, Li Liu, Shawn Xiang Li, Dong Ni, and Tianfu Wang. "Deep learning in medical ultrasound analysis: a review". In: *Engineering* 5.2 (2019), pp. 261–275.
- [43] Sebastian Raschka, Joshua Patterson, and Corey Nolet. "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence". In: *Information* 11.4 (2020), p. 193.
- [44] Hamza Hussain, Khalid Al Tajir, Rashid Habib, Saiyyad Abboud, and Syed Fadel. "The Effects of Political Polarization on Financial Decision Making". In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 4.1 (2023), pp. 420–431.
- [45] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 847–855.

- [46] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. “Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020”. In: *NeurIPS 2020 Competition and Demonstration Track*. PMLR. 2021, pp. 3–26.
- [47] Nima Mostafa, Arman Mohsen, Shahrad Mardin, and Ameen Elyas. “Deciphering the Mysteries: Seasonal Influenza and the Role of Flu Vaccines in Public Health”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 4.1 (2023), pp. 380–392.
- [48] DK Thara, BG PremaSudha, and Fan Xiong. “Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques”. In: *Pattern Recognition Letters* 128 (2019), pp. 544–550.
- [49] Shawni Dutta and Samir Kumar Bandyopadhyay. “Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release”. In: *Iberoamerican journal of medicine* 2.3 (2020), pp. 172–177.
- [50] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. “Network intrusion detection system: A systematic study of machine learning and deep learning approaches”. In: *Transactions on Emerging Telecommunications Technologies* 32.1 (2021), e4150.
- [51] Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, and Helge Janicke. “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study”. In: *Journal of Information Security and Applications* 50 (2020), p. 102419.
- [52] Poojan Oza and Vishal M Patel. “C2ae: Class conditioned auto-encoder for open-set recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2307–2316.
- [53] Chenhao Qu, Rodrigo N Calheiros, and Rajkumar Buyya. “Auto-scaling web applications in clouds: A taxonomy and survey”. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), pp. 1–33.
- [54] Shira Rubin, Daniel Mizrahi, Noam Friedman, Hila Edri, and Tamar Golan. “The World of Advanced Thin Films: Design, Fabrication, and Applications”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 4.1 (2023), pp. 393–406.
- [55] Jinhua Wang, Xi Yang, Hongmin Cai, Wanchang Tan, Cangzheng Jin, and Li Li. “Discrimination of breast cancer with microcalcifications on mammography by deep learning”. In: *Scientific reports* 6.1 (2016), p. 27327.
- [56] Eloi Figueiredo, Gyuhae Park, Charles R Farrar, Keith Worden, and Joaquim Figueiras. “Machine learning algorithms for damage detection under operational and environmental variability”. In: *Structural Health Monitoring* 10.6 (2011), pp. 559–572.
- [57] Junyuan Xie, Linli Xu, and Enhong Chen. “Image denoising and inpainting with deep neural networks”. In: *Advances in neural information processing systems* 25 (2012).
- [58] Noreen Fatima, Li Liu, Sha Hong, and Haroon Ahmed. “Prediction of breast cancer, comparative review of machine learning techniques, and their analysis”. In: *IEEE Access* 8 (2020), pp. 150360–150376.
- [59] Shi Dong, Ping Wang, and Khushnood Abbas. “A survey on deep learning and its applications”. In: *Computer Science Review* 40 (2021), p. 100379.
- [60] Sean Tan, Wei Lee, Chloe Wong, and Yi Jia Chua. “Exploring the Potential and Advancements of Hydrogen Fuel Cells: Outcomes and Applications”. In: *Fusion of Multidisciplinary Research, An International Journal (FMR)* 4.1 (2023), pp. 407–419.
- [61] Jelle W Boumans and Damian Trilling. “Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars”. In: *Rethinking Research Methods in an Age of Digital Journalism* (2018), pp. 8–23.
- [62] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. “Human-AI collaboration in data science:

- Exploring data scientists' perceptions of automated AI". In: *Proceedings of the ACM on human-computer interaction* 3.CSCW (2019), pp. 1–24.
- [63] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting". In: *Advances in neural information processing systems* 34 (2021), pp. 22419–22430.
- [64] Tong Yu and Hong Zhu. "Hyper-parameter optimization: A review of algorithms and applications". In: *arXiv preprint arXiv:2003.05689* (2020).