EMBEDDING AUTOMATED MACHINE LEARNING WITHIN GENERATIVE AI-BASED CONVERSATIONAL SYSTEMS

Teh, Kian Wooi
Tang, Weng Chin
Gonzales, James

Factory Analytics and Operations Excellence – Global Operations
Western Digital Corp., 109 Technology Ave., SEPZ, Laguna Technopark, Biñan, Laguna, Philippines 4024
KianWooi.Teh@wdc.com, Weng.Chin.Tang@wdc.com, James.Gonzales@wdc.com

ABSTRACT

In the era of Industry 4.0, Machine Learning (ML) has emerged as the cornerstone of analytics, data science, and artificial intelligence applications. Its utility extends beyond Predictive Maintenance to encompass Anomaly Detection, Failure Analysis, Defect Classification, and advanced Data Analysis and Model Development.

Automated Machine Learning (AutoML) streamlines the time-consuming, iterative tasks involved in creating ML models. It democratizes machine learning by making it accessible to users without extensive coding proficiency. By automating the machine learning pipeline, AutoML minimizes the need for programming expertise while maintaining analytical thoroughness.

Concurrent with these developments, generative AI (GenAI) has evolved rapidly. GenAI's Large Language Models (LLMs) provide users with enhanced insights and knowledge through conversational systems that respond to human prompts in natural language.

Our novel approach embeds AutoML capabilities directly into GenAI systems through an AutoML Agent, further democratizing and amplifying the benefits of both technologies. This integration has significantly assisted engineers, particularly those with limited programming skills, not only by automating workflow tasks but also by providing comprehensive reports with actionable insights. The conversational interface lowers the barrier to entry for ML tasks, allowing engineers to focus on result interpretation and data-driven decision-making rather than coding complexities. This streamlined interaction model promotes faster experimentation and broader adoption of ML practices across non-specialist domains.

Despite its advantages, the current AutoML Agent has limitations, including the inability to export trained models for external use. Future enhancements will focus on expanding functionality, such as enabling model download and integration with external ML workflows, to support more advanced and customizable use cases.

1. 0 INTRODUCTION

In manufacturing environment such as in Western Digital (WD), Machine Learning (ML) serves as a critical tool for data analysis, predictive maintenance, quality assurance, and process optimization. These applications often rely on analyzing large volumes of sensor data, production logs, and defect patterns to detect anomalies, predict product or equipment failures, and improve product consistency. As shown in Figure 1, developing traditional ML workflows in such settings remains a significant challenge. It typically demands substantial manual effort, time and technical expertise to perform data preprocessing, feature engineering, model selection, training, evaluation, and deployment. This complexity creates a barrier for domain experts on the factory floor particularly for engineers and analysts who may possess deep operational knowledge but lack programming or data science skills.

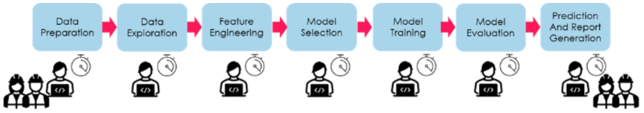


Fig. 1. Traditional Machine Learning Workflow. Typically demands substantial manual efforts, time and technical expertise to perform the ML Pipeline.

Auto-Machine Learning (AutoML) has emerged as a solution to address these challenges by automating many of the tedious and technical stages of the ML pipeline. AutoML frameworks streamline processes such as data cleaning, feature transformation, algorithm selection, hyperparameter tuning, and model validation, making it faster and more consistent to build high-performing models as shown in Figure 2. However, despite these advances, most AutoML systems still require a certain level of data and coding literacy, familiarity with programming environments such as Python or R, and the ability to interpret ML outputs. This limits their accessibility to non-technical users and constrains the widespread adoption of data-driven methods in industrial contexts.

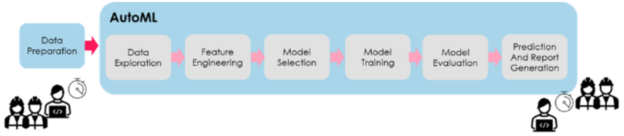
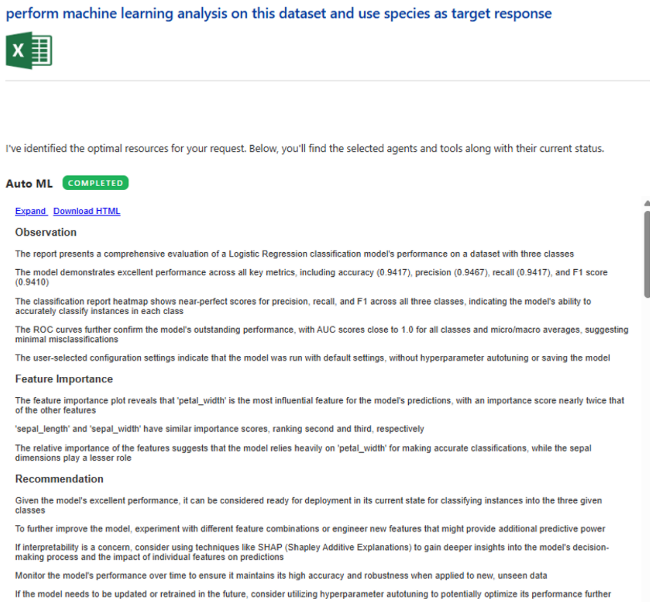


Fig. 2. AutoML Workflow. Automates most of the key ML pipelines however requires users to be literate to programing environments.

In this paper we propose Embedding AutoML into Generative AI (GenAI) conversational system to offer a compelling solution to this problem. GenAI tools, such as large language models (LLMs), are capable of understanding and generating human natural language, enabling interactive conversation with users. When integrated with AutoML, a GenAI-powered system can guide users through the ML workflow, allowing them to upload datasets, define goals, select metrics, and understand model results using plain language. This enables users without coding experience to access the power of machine learning through intuitive, dialogue-based interfaces as shown in Figure 3.



perform machine learning analysis on this dataset and use species as target response

I've identified the optimal resources for your request. Below, you'll find the selected agents and tools along with their current status.

Auto ML COMPLETED

Observation

The report presents a comprehensive evaluation of a Logistic Regression classification model's performance on a dataset with three classes

The model demonstrates excellent performance across all key metrics, including accuracy (0.9417), precision (0.9467), recall (0.9417), and F1 score (0.9410)

The classification report heatmap shows near-perfect scores for precision, recall, and F1 across all three classes, indicating the model's ability to accurately classify instances in each class

The ROC curves further confirm the model's outstanding performance, with AUC scores close to 1.0 for all classes and micro/macro averages, suggesting minimal misclassifications

The user-selected configuration settings indicate that the model was run with default settings, without hyperparameter autotuning or saving the model

Feature Importance

The feature importance plot reveals that 'petal_width' is the most influential feature for the model's predictions, with an importance score nearly twice that of the other features

'sepal_length' and 'sepal_width' have similar importance scores, ranking second and third, respectively

The relative importance of the features suggests that the model relies heavily on 'petal_width' for making accurate classifications, while the sepal dimensions play a lesser role

Recommendation

Given the model's excellent performance, it can be considered ready for deployment in its current state for classifying instances into the three given classes

To further improve the model, experiment with different feature combinations or engineer new features that might provide additional predictive power

If interpretability is a concern, consider using techniques like SHAP (Shapley Additive Explanations) to gain deeper insights into the model's decision-making process and the impact of individual features on predictions

Monitor the model's performance over time to ensure it maintains its high accuracy and robustness when applied to new, unseen data

If the model needs to be updated or retrained in the future, consider utilizing hyperparameter autotuning to potentially optimize its performance further

Fig 3. An engineer could prompt: “Perform machine learning on this data set and use ‘species’ as the target response” and the system would respond by initiating data processing, model training, and result summarization automatically.

Such integration not only lowers the entry barrier for ML adoption in manufacturing but also enhances decision-making by combining automation with contextual understanding. The system can tailor its recommendations based on the user's role, the problem context, and even previous interactions, making AutoML not just automatic but adaptive and explainable.

This paper aims to explore the integration of AutoML with GenAI conversational systems in the context of manufacturing analytics. Specifically, it investigates how this synergy can democratize access to machine learning, reduce the technical burden on domain experts, and accelerate model development and deployment. The key contributions of this work include: (1) a conceptual framework for embedding AutoML workflows within a GenAI-driven dialogue system; (2) an analysis of the benefits and limitations of this approach in industrial settings; and (3) a demonstration illustrating practical use cases such as defect prediction and process optimization. The remainder of the paper is organized as follows: Section 2 reviews related work in AutoML and GenAI systems; Section 3 outlines the proposed system methodology and architecture; Section 4 presents a manufacturing-focused case study; and Section 5 discusses the implications, challenges, and future directions.

2. 0 REVIEW OF RELATED WORK

The integration of generative AI (GenAI) into automated machine learning (AutoML) systems is transforming the design of intelligent systems. This integration is giving rise to a new generation of autonomous agents capable of performing end-to-end machine learning tasks including the generation of analysis reports with minimal human intervention.

Conventional AutoML systems have primarily focused on automating specific aspects of the machine learning pipeline, such as model selection, feature engineering, and hyperparameter optimization, typically using search algorithms and statistical heuristics [1 and 2]. While effective, these systems are limited in scope and often require pre-defined pipelines and structured input.

In contrast, recent advancements in GenAI have introduced transformative capabilities, such as natural language understanding, task reasoning, program synthesis, and dynamic API interaction. These capabilities allow agents to act as general-purpose, conversational tools that can be prompted in natural language to perform complex machine learning workflows. By incorporating mechanisms for iterative feedback, self-evaluation, and chain-of-thought reasoning, GenAI-powered AutoML agents can refine their outputs over time, improving the quality and reliability of their results.

Several emerging frameworks illustrated this convergence. Projects such as AutoGen [3], LangChain [4], and AgentVerse [5] demonstrate how GenAI models, particularly large language models (LLMs), can be embedded within multi-agent systems. In these architectures, specialized agents collaborate on subtasks such as data preprocessing, model training, evaluation, and visualization through conversational or code-executing interfaces. These multi-agent systems represent a significant evolution from earlier AutoML platforms, moving from rigid pipelines to dynamic, goal-driven collaboration.

This paradigm shift introduces a number of new research challenges. Key among these are issues are controllability: guiding agent behavior toward desired outcomes, trust: ensuring model reliability and robustness, and evaluation: developing benchmarks for GenAI-embedded ML systems. Moreover, as these tools become more accessible, they hold significant promise for democratizing data science, lowering the barrier to entry for non-experts, and accelerating ML development in various domains.

3.0 METHODOLOGY

This paper implements a Generative AI-based system that automates statistical insight extraction to address manual analysis limitations in large-scale manufacturing. The system supports natural language queries for performing exploratory analysis, anomaly detection, and experiment design without coding.

In this section, describes how AutoML is embedded as an Agent to the Generative AI based system. The GenAI system supports natural language prompt to trigger and execute the AutoML agent.

3.1 System Architecture

In Figure 4, illustrates the System Architecture overall architecture and where the AutoML resides. The user prompts using natural human language and attaching the dataset to the User Interface (UI Web browser) which communicates via HTTP protocol to the Web Server. The Web Server routes the request and response to the Application Program Interface (API) which mediates requests and response for multiple agents and pipelines. Database is utilize for the logging, memory storage purposes.

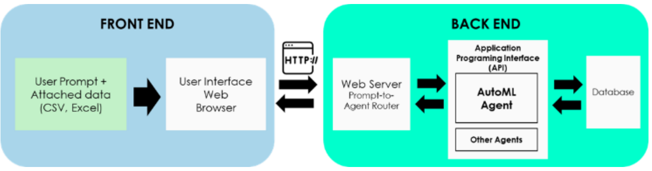

FRONT END
User Prompt + Attached data (CSV, Excel)
User Interface Web Browser
HTTP://
BACK END
Web Server Prompt-to-Agent Router
Application Programing Interface (API)
AutoML Agent
Other Agents
Database

Fig 4. The System Architecture shows the user interacts using the user interface to communicate with the webserver to communicate with the AutoML agent..

The development of AutoML agent involves several key stages, each contributing to the model's ability to understand and respond effectively to user inputs. These stages include design specification, data preprocessing, model training, model selection and evaluation.

The first step in developing an AutoML agent is the specification of system requirements. The scope of this system will be a domain-specific assistant which eases engineers to perform ML analysis even without any ML analysis tool required.

Figure 5 illustrates the flow of user request and AutoML response. The API receives the request from the Webserver as a Payload and performs the AutoML workflow.

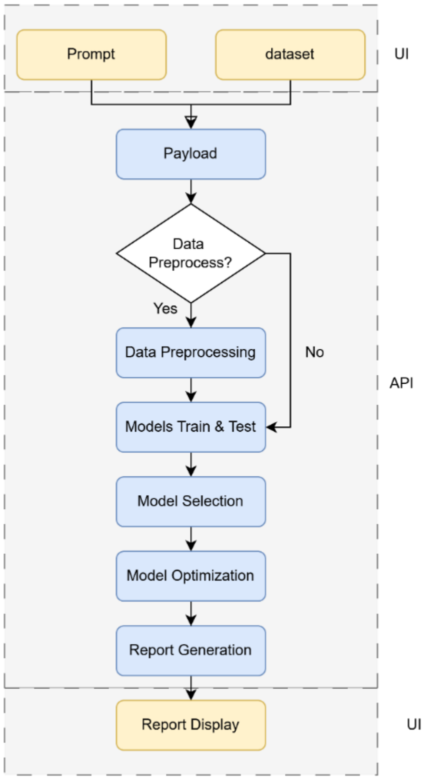


Fig. 5. AutoML agent work flow.

3.2 User Interface

The GenAI-based conversational system UI will receive the user's inputs by file attachment and user's prompt. The file attachment allows users to submit and upload the dataset whereas the user's prompt will be the instruction of user what to be done.

Once the user submits the dataset and the user's prompt, the dataset will be uploaded into file server and generate a dataset link. At the same time, the GenAI model will process the user's prompt and generate payload which includes the dataset link. Then trigger the API to process the data according to the payload and return the result of analysis to UI for user viewing.

3.3 Application Programming Interface

The API is the core of AutoML agent where will process dataset according to the payload generated by GenAI model according to user's input. There are few steps embedded in API so it able to perform the ML analysis automatically.

Step 1: Data Preprocessing
This step will help the user to prepare a clean dataset such remove or impute empty data, remove duplicated data, data transformation etc. This is very important to ensure the data was clean before ML analysis and provide trustable results. This step will help to perform the feature selection which helps to identify the high correlation features and reduce the features to aim to improve the performance of ML analysis.

This step has flexibility that allows user to instruct whether to perform. Users can skip this step if the input dataset was a cleaned dataset and can reduce overall analysis processing time.

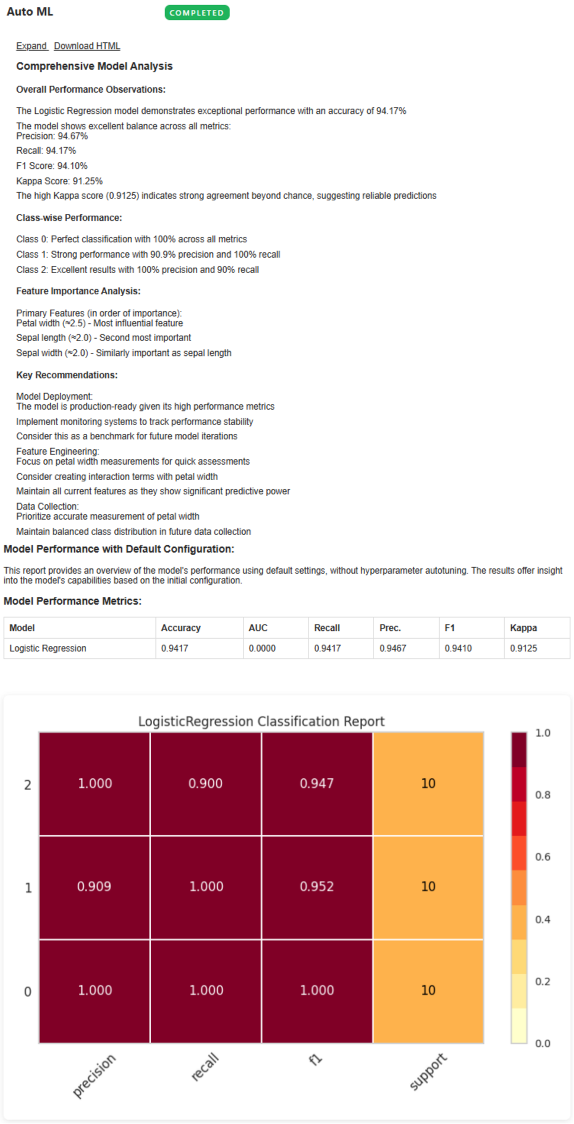
Step 2: Model Train & Test
Once the cleaned dataset is obtained, the dataset will be split into train and test dataset according to user defined ratio or use the predefined ratio if user didn't define the ratio. Then, API will trigger to perform model train and test, then provide the model performance matrix for each model.

Step 3: Model Selection
The API is capable to collect the model performance matrix for all models, then compares and proposes the best models. Besides automated model selection, users are allowed to choose the preferred model.

Step 4: Model Optimization
This purpose is to further optimize the model performance by tuning the hyperparameters. This step is an optional ML data processing based on user preferences.

Step 5: Report Generation
This final step in AutoML API where the result compilation has been done and generates a report with insight.

Once the report has been generated, then return this report to UI and show it as illustrated in Figure 5. This will allow users to understand the ML analysis result easily and be able to make a better decision based on the report.



Auto ML

Comprehensive Model Analysis

Overall Performance Observations:

The Logistic Regression model demonstrates exceptional performance with an accuracy of 94.17%

The model shows excellent balance across all metrics:
Precision: 94.67%

Recall: 94.17%

F1 Score: 94.10%

Kappa Score: 91.25%

The high Kappa score (0.9125) indicates strong agreement beyond chance, suggesting reliable predictions

Class-wise Performance:

Class 0: Perfect classification with 100% across all metrics

Class 1: Strong performance with 90.9% precision and 100% recall

Class 2: Excellent results with 100% precision and 90% recall

Feature Importance Analysis:

Primary Features (in order of importance):
Petal width (≈2.5) - Most influential feature

Sepal length (≈2.0) - Second most important

Sepal width (≈2.0) - Similarly important as sepal length

Key Recommendations:

Model Deployment:
The model is production-ready given its high performance metrics

Implement monitoring systems to track performance stability

Consider this as a benchmark for future model iterations

Feature Engineering:
Focus on petal width measurements for quick assessments

Consider creating interaction terms with petal width

Maintain all current features as they show significant predictive power

Data Collection:
Prioritize accurate measurement of petal width

Maintain balanced class distribution in future data collection

Model Performance with Default Configuration:

This report provides an overview of the model's performance using default settings, without hyperparameter autotuning. The results offer insight into the model's capabilities based on the initial configuration.

Model Performance Metrics:

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Logistic Regression	0.9417	0.0000	0.9417	0.9467	0.9410	0.9125

Fig. 5. Snapshot of AutoML agent's on portion of report on sample dataset. The report includes confusion matrix, ROC curve plot, feature importance plot and other charts.

3.4 Agent Assessment

To ensure the agent is working well and comparable with existing practice, the assessment of the ML analysis result has been carried out. The existing practice refers to the ML analysis which is carried by engineers manually using any data analysis tools or software.

Both AutoML agent and existing practice carried similar ML steps where they start from data preprocessing until obtain

ML performance matrix. Both used the same dataset and used logistic regression as model of ML analysis. So, the assessment of AutoML agent will compare with the existing practice based on ML performance matrix such as accuracy, recall score and F1 score. The result discussed in section 4.

4.0 RESULTS AND DISCUSSION

This section presents the results of implementing and evaluating the proposed generative AI-embedded AutoML conversational system. The system evaluated with 10 participants (5 novice, 5 ML practitioners). Using benchmark datasets, the AI-embedded AutoML agent were assessed across multiple performance dimensions including task accuracy, usability, and user experience by rating.

Table 1. AI-embedded AutoML Performance Matrix

Dataset	Task	ML Metric	Time (min)
Iris	Classification	94.7%	4
Titanic	Classification	81.5%	4
Housing	Regression	0.89 (R^2)	6

A set of benchmark datasets were used to evaluate the system. Metrics included, Accuracy, and user satisfaction

Performance metrics from models selected by the Modeling Agent were comparable to or better than manually tuned baselines, demonstrating the system's ability to automate ML workflows effectively.

Table 2. Participants Feedback

Evaluation Metric	Novices (Avg)	Practitioners (Avg)
Ease of Use	4.2	4.1
Usefulness of Explanation	4.4	3.9
Trust in System Decision	4	4.2

Participants provided usability feedback through post-task surveys and interviews.

Novice users appreciated the system’s conversational interface, which minimize the technical complexity of the process, and provided guided ML analysis. On the other hand, Practitioners valued the flexibility of natural language prompting and the ability to refine the conditions of ML parameters.
Participants cited the ease of use, and interactive result such as plots and summaries as key strengths.

While the system performed reliably in most test cases, several limitations were observed. Such as (1) Prompt Ambiguity which sometimes vague or underspecified prompts led to unintended task flows, suggesting a need for tighter prompt-template alignment or active clarification mechanisms. (2) Execution Errors due to data formatting expectations and issues.

The results demonstrate that embedding AutoML within a GenAI-based conversational system is not only feasible but also practical and effective. The system supports end-to-end ML tasks with minimal user guidance and shows significant promise in democratizing access to machine learning.

From a systems design perspective, the modular architecture enabled robust delegation of tasks for multiple current and potential agents, with the LLM and APIs orchestrating complex workflows in a flexible, user-aligned manner. The UI where users interact conversationally to improve model performance or adjust parameters, emerged as a powerful differentiator from traditional ML and conventional AutoML pipelines.

The observed results also highlight the need for future research in areas such as:

- **Prompt engineering frameworks** for more controlled task specification.
- **Validation layers** to handle invalid dataset format and catch unacceptable or hallucinated response.
- **Explainability modules** to improve trust and interpretability of agent decisions.

5.0 CONCLUSION

This paper introduced a novel concept and architecture that integrates Automated Machine Learning (AutoML) capabilities within a generative AI-driven conversational system. By embedding AutoML as agent under the orchestration of a large language model, the system enables intuitive, natural language-based interaction for executing complex machine learning workflows ranging from data ingestion and preprocessing, model selection, model training, evaluation, and report generation.

The experimental results demonstrate the effectiveness of this approach in delivering accurate ML outputs while significantly reducing the technical barrier for users. Both Novice and ML Practitioners users were able to interact with the system to process machine learning analysis and generate visual outputs and refine results using the conversational UI. The results validate the system’s potential to democratize machine learning, enhance productivity, and serve as a powerful tool for fast paced ML prototyping, thus, it is expected to benefit the WD engineers and analyst to focus more on the result interpretation and decision making based on the ML analysis report which included some insights.

Despite these promising outcomes, several challenges remain. Current limitations around prompt ambiguity and execution reliability highlight the need for more robust validation mechanisms and contextual safeguards.

By continuing to refine the integration between GenAI and AutoML in WD, empower users across domains to become active participants in data-driven problem solving.

6.0 RECOMMENDATIONS

It is recommended to further improve the system with research and development to focus on the following directions. (1) Develop code validation modules to for stronger error handling and improve reliability. (2) Extend the system to support other data types (e.g., images, time-series) and incorporate multi-modal inputs and outputs, allowing for more flexible ML applications. (3) Ability to export trained model to be applied inline of the manufacturing process for monitoring and process control.

7.0 ACKNOWLEDGMENT

The authors would like to thank the leadership and management of Operations and Analytics Excellence Department for their support and guidance of this project. Appreciation is also extended to the Stakeholders for their invaluable feedback.

8.0 REFERENCES

1. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F., *Efficient and robust automated machine learning*, Advances in Neural Information Processing Systems, 28, 2015.

2. Elsken, T., Metzen, J. H., & Hutter, F., *Neural architecture search: A survey*, Journal of Machine Learning Research, 20(55), 2019, 1-21.

3. Wu, Y., Zhang, J., Lin, S., et al., *AutoGen: Enabling next-generation large language model applications via multi-agent conversation*, arXiv:2308.08155v2, 2023.

4. Chase, H, *LangChain: Building applications with LLMs through composability*, https://www.langchain.com/, 2023.

5. Liu, X., Zhu, H., Zhang, Y., et al., *AgentVerse: Facilitating multi-agent collaboration and exploration with LLMs*, arXiv:2309.07864, 2023.

9.0 ABOUT THE AUTHORS

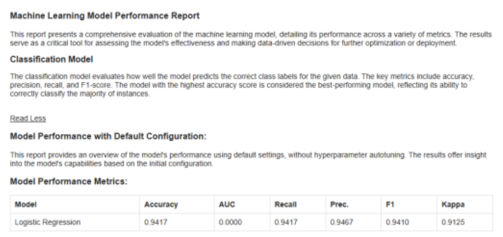
Kian Wooi, Teh is a Principal Engineer at Western Digital, Penang Media Plant. He specializes in data analytics, machine learning and AI projects.

Weng Chin, Tang is a Senior Manager at Western Digital, Penang Media Plant. He specializes in AI projects.

James Gonzales is a Data Scientist at Western Digital, Philippine Head Office. He holds a master’s degree in data science from Asian Institute of Management and bachelor’s degree in Electronics and Communications Engineering from Polytechnic University of the Philippines – Taguig. He has over 15 years combined experience working in the magnetic head industry as a Test Engineer and Data Scientist.

10.0 APPENDIX

Appendix A – Expanded Visualizations for Embedded AutoML Report.



Machine Learning Model Performance Report

This report presents a comprehensive evaluation of the machine learning model, detailing its performance across a variety of metrics. The results serve as a critical tool for assessing the model's effectiveness and making data-driven decisions for further optimization or deployment.

Classification Model

The classification model evaluates how well the model predicts the correct class labels for the given data. The key metrics include accuracy, precision, recall, and F1-score. The model with the highest accuracy score is considered the best-performing model, reflecting its ability to correctly classify the majority of instances.

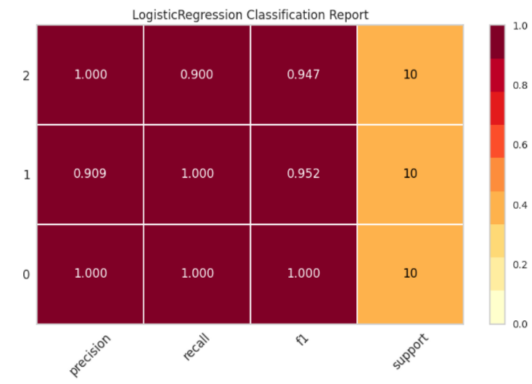
Read Less

Model Performance with Default Configuration:

This report provides an overview of the model's performance using default settings, without hyperparameter autotuning. The results offer insight into the model's capabilities based on the initial configuration.

Model Performance Metrics:

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Logistic Regression	0.9417	0.0000	0.9417	0.9467	0.9410	0.9125



Observations:

The logistic regression model shows excellent performance across all classes (0, 1, and 2)

Class 0 shows perfect scores with 1.000 for precision, recall, and F1-score, indicating flawless classification

Class 1 demonstrates strong performance with:
Precision: 0.909

Recall: 1.000

F1-score: 0.952

Class 2 also shows robust performance with:
Precision: 1.000

Recall: 0.900

F1-score: 0.947

All classes have equal support of 10 samples, indicating a balanced dataset

Recommendations:

The model is performing exceptionally well and is suitable for deployment

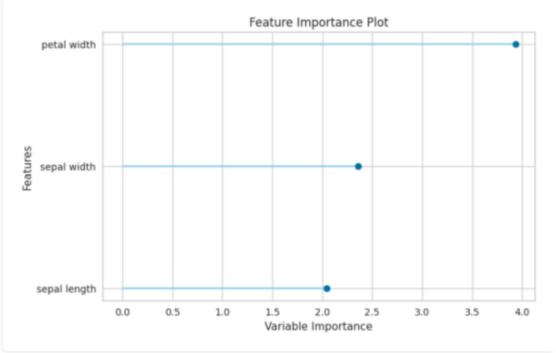
Consider using this model as a benchmark for comparing other algorithms

To further validate the model's robustness:
Test on a larger dataset if available

Implement cross-validation to ensure consistency

Monitor performance on new data to ensure maintained accuracy

The balanced support suggests good data distribution, maintain this balance for future training iterations



Key Observations:

Petal width appears to be the most influential feature with an importance score of approximately 4.0, suggesting it is the strongest predictor for classification

Sepal width and sepal length show similar importance levels around 2.0-2.5, indicating moderate influence on the model's predictions

There is a clear hierarchy in feature importance, with petal width being roughly twice as important as the other features

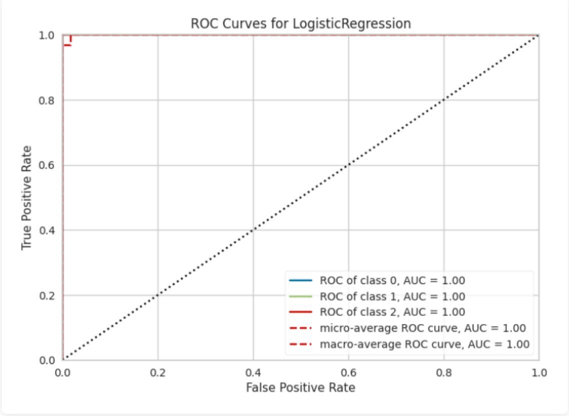
Recommendations:

Focus on petal width measurements for quick and efficient iris classification

Consider developing a simplified model using primarily petal width if resource optimization is needed

While sepal measurements are less important, retain them for model robustness as they still contribute meaningful information

For future data collection, prioritize accurate petal width measurements



ROC Curve Analysis Insights:

Key Observations:

Perfect Classification Performance: All classes (0, 1, and 2) show an AUC (Area Under Curve) score of 1.00, which indicates perfect classification

Both micro-average and macro-average ROC curves also achieve perfect AUC scores of 1.00

The curves for all classes are perfectly aligned at the top-left corner of the plot, indicating optimal discrimination between classes

Recommendations:

While perfect classification is impressive, it's important to:
Check for potential data leakage

Validate the model on completely independent test data

Consider if the model might be overfitting

If this is production data, consider:
Implementing model monitoring to ensure continued performance

Regular model retraining to maintain this level of performance

Cross-validation to ensure robustness across different data splits