KAIST DeepAuto.ai

ICML International Conference On Machine Learning

AutoML-Agent

A Multi-Agent LLM Framework for Full-Pipeline AutoML

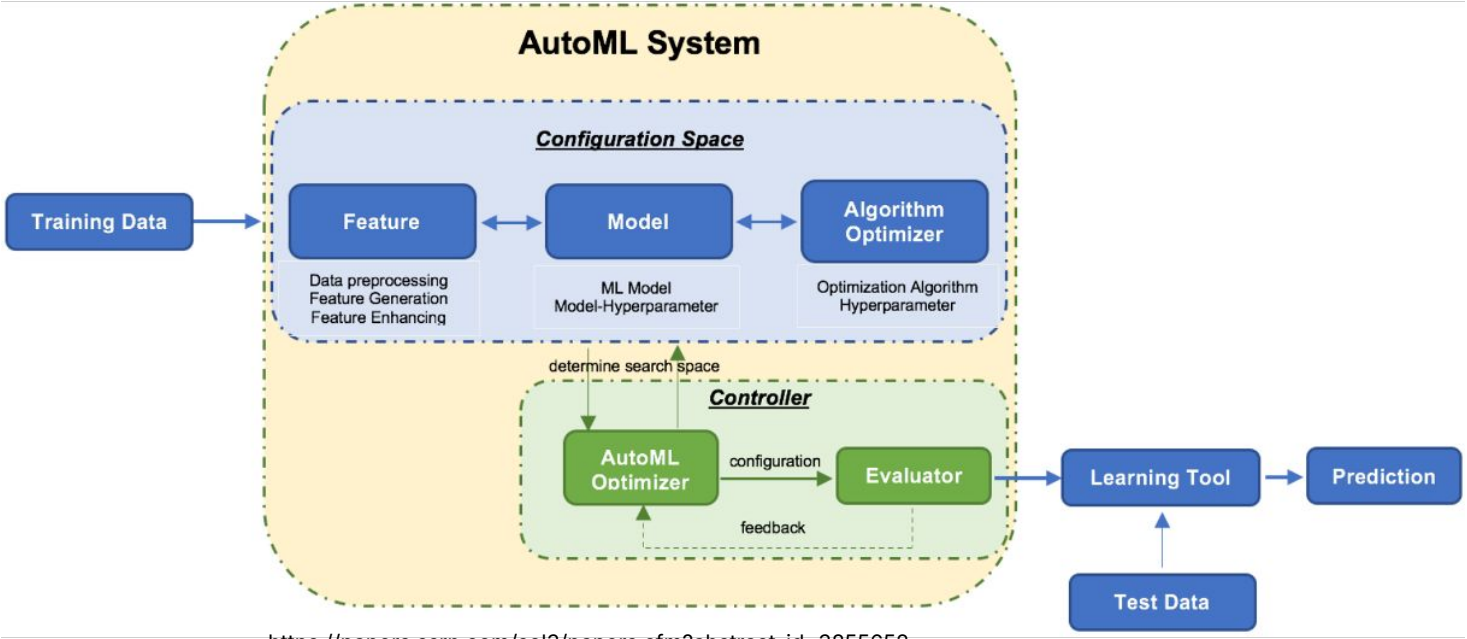
Patara Trirat[1] Wonyong Jeong[1] Sung Ju Hwang[1 2]

[1]DeepAuto.ai [2]KAIST

https://deepauto-ai.github.io/automl-agent

Automating Full-Pipeline ML for Everyone

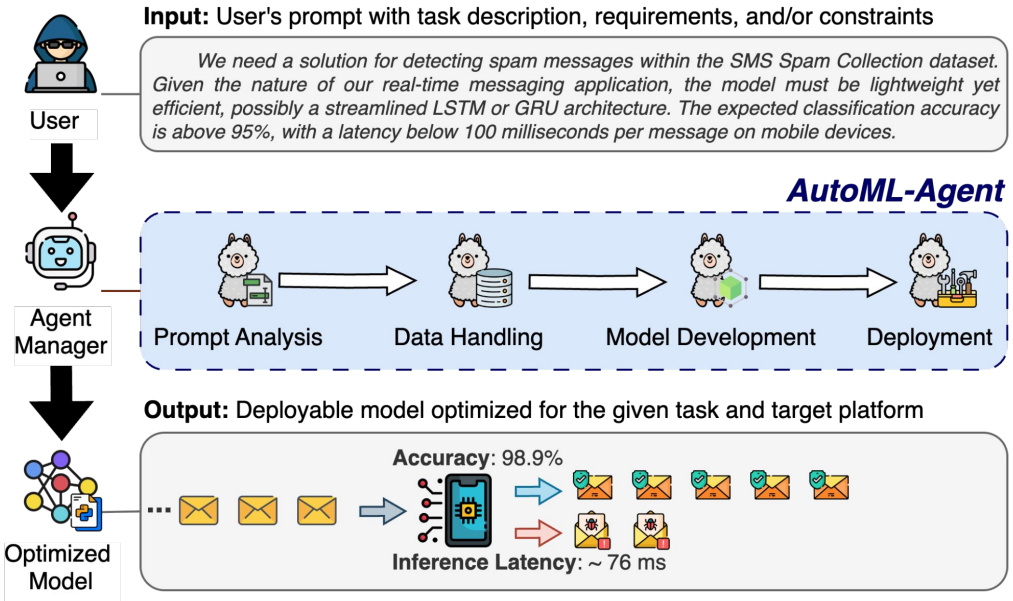
Building effective ML pipelines remains **labor-intensive** and involves **multiple complex steps**, requiring significant expertise and time.



https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3855652

LLMs can Automate the AI Development Pipeline

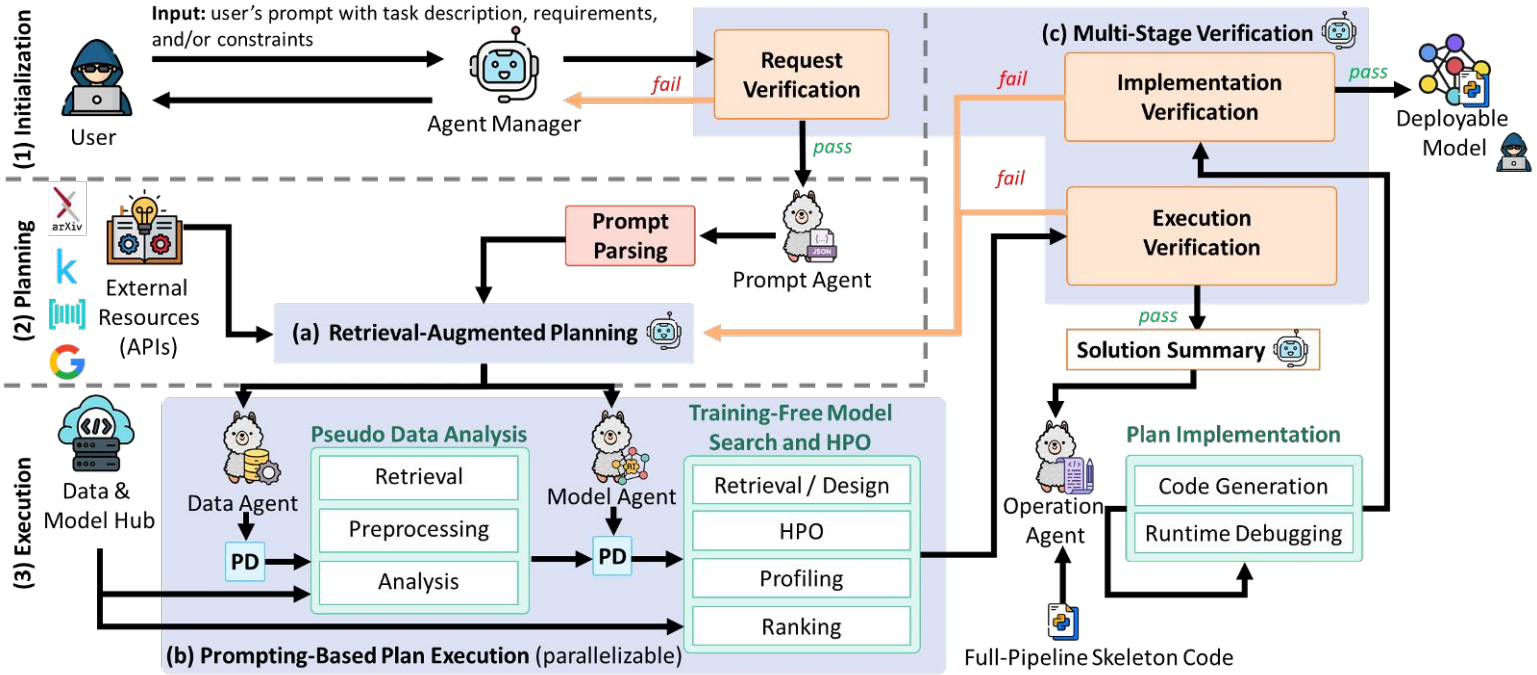
To resolve this issue, we leverage the power of LLMs, which can **understand tasks** expressed in natural language and **generate plans and code** to solve them.



This approach **reduces the barrier to entry**, enabling users to build data-driven solutions through conversational interactions.

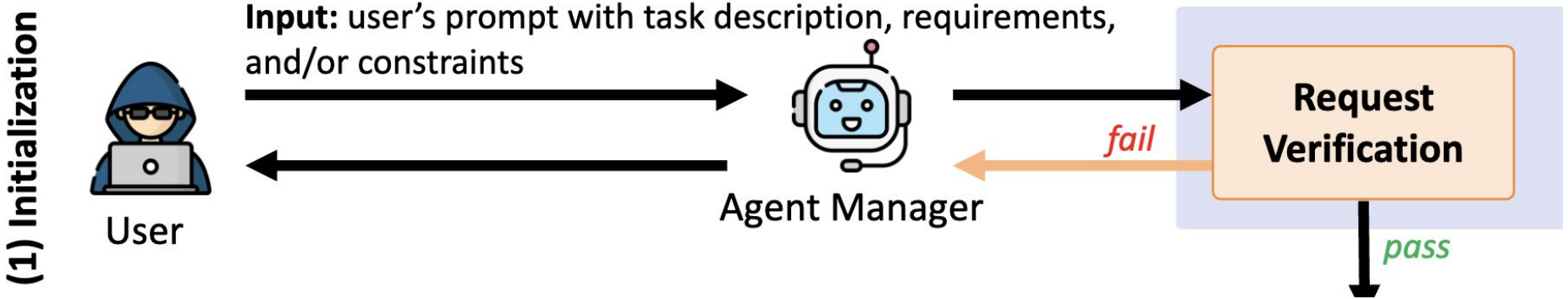
A Multi-Agent System for Full-Pipeline AutoML

To ensure the efficient and accurate automation of the entire ML pipeline, we introduce a multi-agent framework in which **specialized LLM agents handle distinct tasks**.

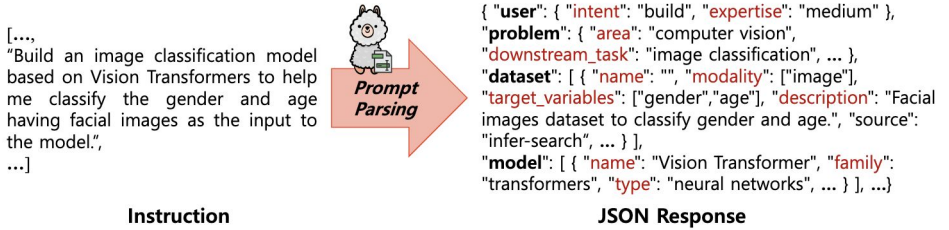


Initialization

The **Agent Manager** first **checks the user instruction**: if valid, it forwards it to the **Prompt Agent**, and otherwise otherwise, it asks the user for clarification.

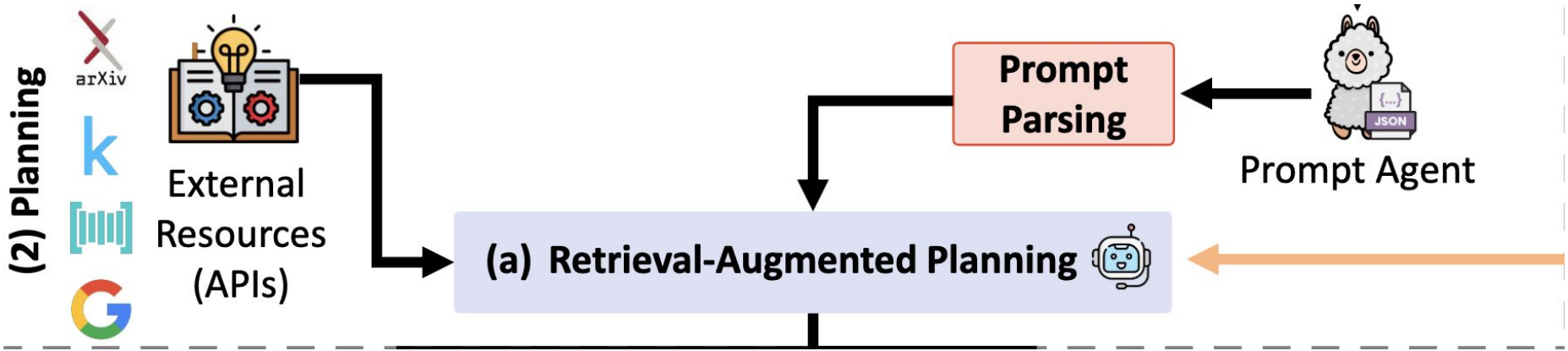


The **Prompt Agent** parses the instruction into structured JSON, which the **Agent Manager** uses to **retrieve relevant knowledge** and **generate global plans**.



Planning

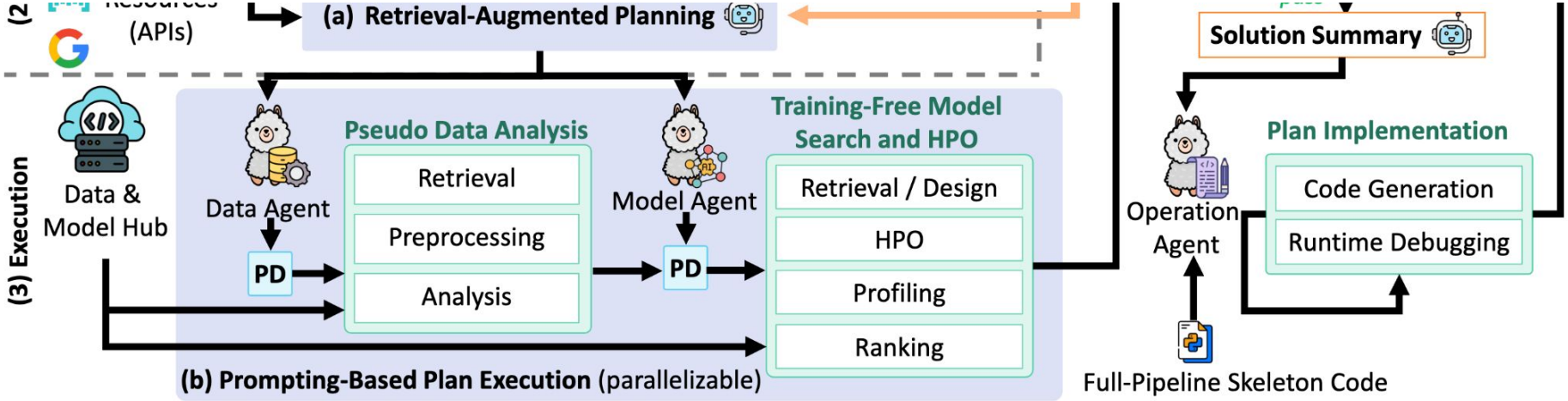
We present a **retrieval-augmented planning** strategy that generates multiple plans to **explore optimal solutions by accessing external knowledge sources**.



Each global plan is then **decomposed into smaller sub-tasks** (e.g., data loading, model search, HPO) and assigned to the **Data** and **Model Agents**, which simulate **execution through prompting**.

Execution

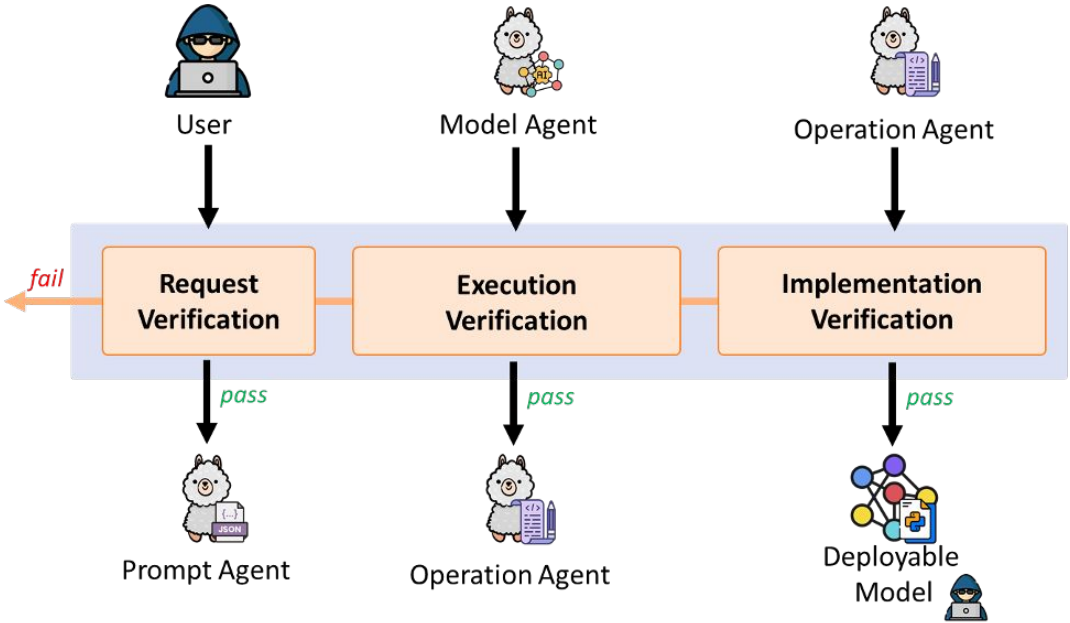
Data Agent performs dataset retrieval, preprocessing, and analysis
Model Agent searches for suitable models, performs HPO, model profiling and ranking.



The best plan selected by the **Agent Manager** during the simulation is then executed by the **Operation Agent**, which **writes the actual code and performs runtime debugging**.

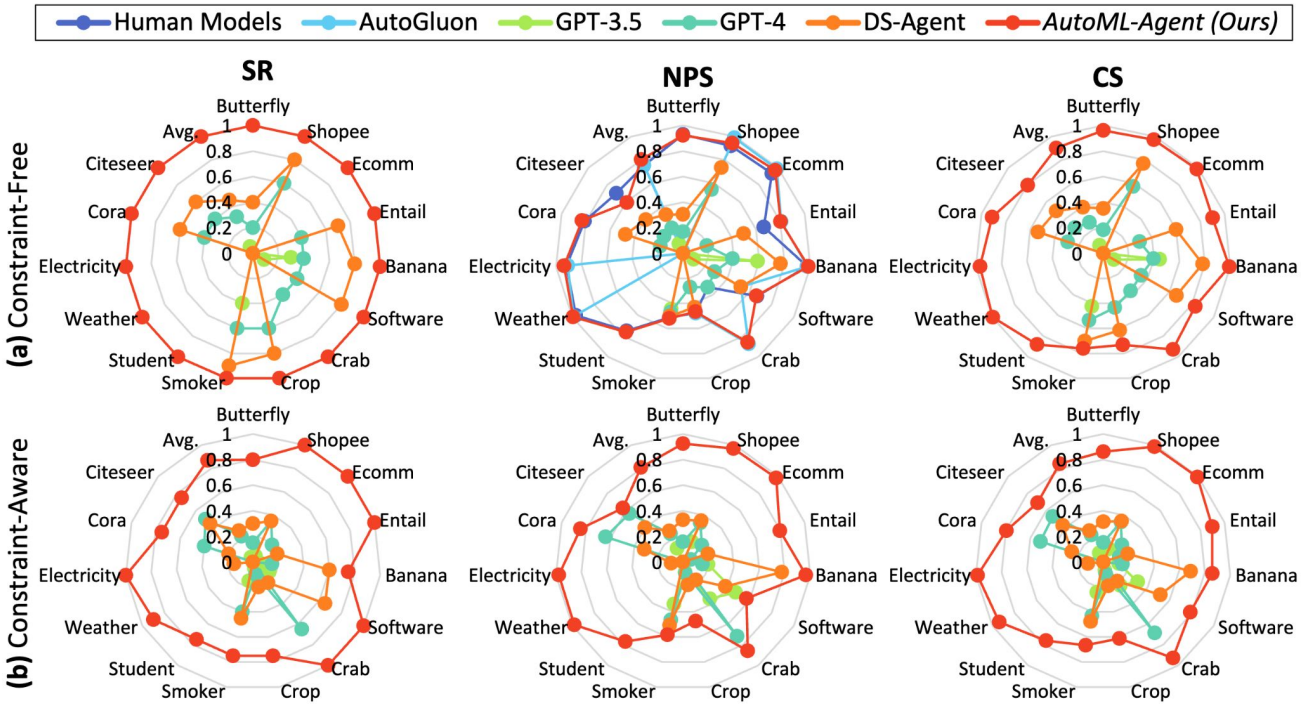
Multi-Stage Verification for Reliable Code

To ensure the reliability of generated code, we also incorporate a **multi-stage verification** process.



This process includes **intermediate checks and validations**, guiding agents to **refine outputs and adhere to user requirements**.

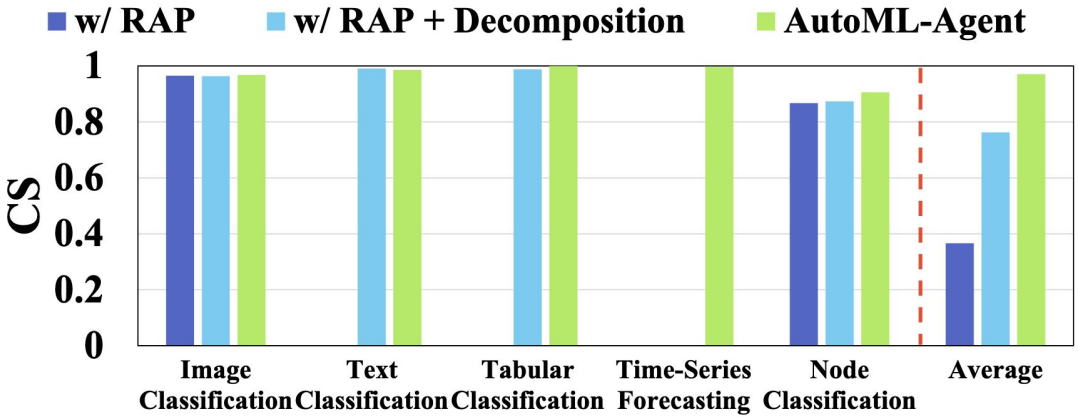
Results on Diverse Tasks and Settings

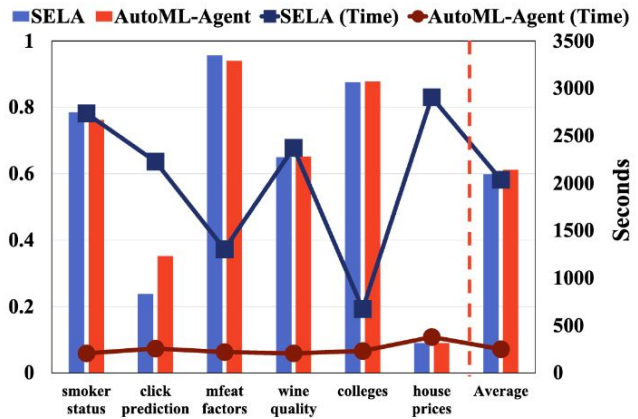


Evaluations on **seven downstream tasks** using **fourteen datasets** demonstrate that *AutoML-Agent* successfully writes executable code and produces models that outperform human-built ones.

Effectiveness and Efficiency of AutoML-Agent

The ablation study (left) shows that the plan decomposition and verification stages are indeed effective, surpassing the performance of naïve retrieval-based planning (RAP)





Compared to training-based methods like SELA (right), our framework is **8× faster** while achieving similar or better results, with reduced computational resources and time.

Limitations

- While training-free, AutoML-Agent could incur non-trivial time and inference cost for planning, multi-agent coordination, iterative plan refinement, and multi-step verification.
- LLMs can hallucinate or generate incomplete / incorrect code, which may incur a large number of the revise and retry steps.
- Although we evaluate AutoML-Agent across multiple machine learning tasks, specialized tasks (e.g., reinforcement learning and recommendation systems) may require further modifications to the framework.

Thank you!

GitHub: github.com/deepauto-ai/automl-agent
Email: patara@deepauto.ai