LLM-Based Agent for Supervised Classifier Lifecycle Automation: Shifting from Stochastic Search to Heuristic Reasoning

Abstract. Algorithm selection and hyperparameter optimization are critical stages in the lifecycle of supervised classifiers, traditionally requiring a significant investment of time, computational capacity, and expert supervision. This work evaluates the potential of Large Language Models (LLMs) as decision-support assistants for automating this process. Preliminary results are reported in which an LLM-based agent assumes the role of a human specialist during the training and fine-tuning phase, operating iteratively on the dataset. The experimental design compares two deployment modes, one based on closed weights accessed through an API and another based on open weights executed locally, under a common optimization protocol. The results show that the agent, in both deployment modes (via API and local execution), achieves performance comparable to the reference values reported by human experts. These findings support the feasibility of using LLM-driven agents to reduce manual effort while preserving traceability, iterative feedback, and competitive predictive performance.

Keywords: large language models; hyperparameter optimization; supervised learning; automated machine learning; intelligent agents.

Automatización asistida por LLMs en el ciclo de vida de clasificadores supervisados: De la exploración estocástica al razonamiento heurístico

Resumen. La selección de algoritmos de aprendizaje automático y la optimización de sus hiperparámetros constituyen fases críticas del ciclo de vida de los clasificadores supervisados, demandando tradicionalmente una inversión significativa de tiempo, capacidad de cómputo y supervisión experta. El presente trabajo evalúa el potencial de los Modelos Grandes de Lenguaje (LLMs) como asistentes de decisión en la automatización de dicho ciclo. Se reportan resultados preliminares, en los cuales un agente LLM asume el rol de especialista humano en la fase de entrenamiento y ajuste fino, operando de forma iterativa sobre el conjunto de datos. El diseño experimental contrasta dos modalidades de despliegue, una basada en pesos cerrados gestionados vía API y otra basada en pesos abiertos ejecutados localmente, bajo un protocolo común de optimización. Los resultados muestran que el agente, en sus dos modalidades de despliegue (vía API y local), presenta rendimientos comparables a los de referencia reportados por expertos humanos. Estos resultados demuestran la viabilidad de una automatización más eficiente, explicable y soberana en el desarrollo de soluciones de Machine Learning, con capacidad para reducir trabajo manual sin perder trazabilidad ni consistencia en el proceso iterativo.

Palabras clave: Modelos grandes de lenguaje; Optimización de hiperparámetros; Aprendizaje supervisado; Aprendizaje automático automatizado; Agentes inteligentes.

1 Introducción

El desarrollo de modelos predictivos basados en aprendizaje automático supervisado se ha consolidado como una práctica extendida en dominios académicos e industriales. Sin embargo, el proceso de selección del algoritmo más adecuado y el ajuste de sus hiperparámetros (aun aplicando estrategias como búsqueda aleatoria u optimización bayesiana) demanda una inversión significativa de tiempo, recursos computacionales y profesionales especializados (Martínez-Plumed et al., 2022; Ali et al., 2023).

Los enfoques de automatización existentes, como AutoML y MLOps, han logrado reducir parcialmente esta carga mediante la optimización matemática del espacio de búsqueda, no obstante, operan de manera mecánica, no comprenden el dominio del problema ni pueden razonar causalmente sobre los resultados obtenidos (Huang et al., 2026; Azevedo et al., 2024). Esta limitación ha motivado la exploración de los Modelos Grandes de Lenguaje (LLMs) como componentes de razonamiento dentro del ciclo de vida del aprendizaje automático, gracias a su capacidad para interpretar contexto semántico, analizar métricas y proponer configuraciones de manera iterativa (Xu et al., 2025; Luo et al., 2025).

Este trabajo propone un enfoque basado en la integración de un agente LLM dentro de un entorno experimental controlado, en el cual interactúa con un conjunto de 15 clasificadores supervisados representativos del estado del arte. El agente asume funciones críticas tradicionalmente ejecutadas por un especialista humano, tales como la selección de modelos, el análisis de métricas de desempeño y el ajuste iterativo de hiperparámetros. A traves de este diseño se busca cuantificar el potencial de los LLMs como motores de optimización y apoyo en la automatización del ciclo de vida de los modelos.

2 Metodología

2.1 Conjunto de datos

Se utilizó el conjunto de datos Bank Marketing (Moro et al., 2014), disponible en el repositorio UCI (https://archive.ics.uci.edu/). Este conjunto describe campañas de marketing telefónico de una entidad bancaria portuguesa cuya finalidad era conocer si un cliente suscribirá un depósito a plazo fijo. Como referencia humana se tomó el trabajo de Noviana and Sela (2024), quienes aplicaron un preprocesamiento replicable y obtuvieron un conjunto final de 73.074 instancias.

En el presente trabajo se replica exactamente dicho preprocesamiento con el propósito de garantizar una comparabilidad objetiva entre los resultados obtenidos por el agente basado en LLM y los informados por especialistas humanos. Particularmente, se reprodujeron las siguientes etapas sobre el conjunto de datos: verificación de valores faltantes, eliminación de registros duplicados, codificación de variables categóricas, normalización de los datos y balanceo de clases mediante SMOTE. El análisis se concentra en la capacidad del agente para intervenir en la selección, la parametrización y la evaluación de clasificadores dentro de un escenario experimental controlado.

2.2 Diseño del experimento

El agente LLM recibe como contexto inicial la descripción estructural y semántica del conjunto de datos, la variable objetivo y los parámetros del experimento: número de modelos a entrenar ($N = 3$), número máximo de iteraciones por modelo (30) y un umbral de parada anticipada por exactitud, fijado por encima del rendimiento del experto. A partir de este contexto se construyeron PROMPS estructurados con la metodología TCREI (Google Education, 2024), el agente propone N modelos del catálogo junto con una configuración inicial de hiperparámetros en formato JSON estructurado. Cada configuración es sometida a un ciclo iterativo de optimización. En cada iteración se realizan diez entrenamientos independientes y se reporta la media de las métricas como resultado representativo. El par $\{\text{hiperparámetros}, \text{métricas}\}$ se incorpora al historial, que es provisto al LLM en cada nueva iteración para guiar su siguiente propuesta, la cual devuelve acompañada de una justificación textual de los cambios realizados. El experimento se ejecutó en dos instancias independientes: un modelo de pesos cerrados gestionado vía API (GPT-4.5, OpenAI) y un modelo de pesos abiertos en despliegue local (DeepSeek-R1-Distill-Qwen-14B). Ambos operaron con temperatura 0,0 y un límite de 1,000 tokens por sugerencia de hiperparámetros.

2.3 Criterios de evaluación

La evaluación se estructuró en tres dimensiones. En primer lugar, se consideró el rendimiento predictivo, comparando las métricas alcanzadas por el prototipo con las reportadas por el experto en el trabajo de referencia. En segundo lugar, se analizó la calidad del proceso de optimización, observando la tendencia de mejora iterativa y la tasa de validez de las propuestas generadas por el agente. Finalmente, se examinó la coherencia del razonamiento mediante una evaluación cualitativa de las justificaciones textuales del agente, con el fin de determinar si resultan consistentes con la evolución del historial de iteraciones.

3 Resultados preliminares

El experimento generó en total seis propuestas de modelos, 180 sugerencias de hiperparámetros y el ajuste de 1800 clasificadores distribuidos entre ambas instancias.

3.1 Rendimiento predictivo

La Tabla 1 presenta los mejores resultados obtenidos por cada modelo en comparación con el experto humano de referencia. Ambas instancias igualaron o superaron el umbral del experto en sus mejores configuraciones. La diferencia entre el mejor resultado del prototipo (95.49 %, Extra Trees con modelo local) y el del especialista (95.30 % con Random Forest) es de 0.19 puntos porcentuales. Es relevante destacar que ninguno de los agentes recibió orientación explícita sobre qué modelos seleccionar; la convergencia hacia métodos de ensamble de árboles (LightGBM, XGBoost, Extra Trees, Random Forest) emergió del análisis autónomo del agente sobre los datos proporcionados, lo cual es consistente con el estado del arte para datos tabulares (Fernandez, 2021).

Tabla 1: Mejores resultados de desempeño predictivo por instancia y modelo.

Instancia	Modelo	Accuracy	F1-score	Recall
Experto humano	Random Forest	**95.30 %**	0.9500	0.9500
Experto humano	Regresión Logística	87.21 %	0.8700	0.8700
Pesos cerrados (API)	LightGBM	**95.35 %**	0.9537	0.9579
Pesos cerrados (API)	XGBoost	95.31 %	0.9533	0.9568
Pesos cerrados (API)	Random Forest	94.87 %	0.9498	0.9715
Pesos abiertos (local)	Extra Trees	**95.49 %**	0.9561	0.9815
Pesos abiertos (local)	XGBoost	95.14 %	0.9516	0.9543
Pesos abiertos (local)	LightGBM	95.01 %	0.9506	0.9599

3.2 Calidad del proceso de optimización

A lo largo de las 180 sugerencias de hiperparámetros generadas, no se registró ningún fallo de validación de esquema en ninguna de las dos instancias, lo que resultó en una tasa de validez del 100 %. En consecuencia, ambos modelos operaron de forma confiable dentro de las restricciones formales definidas por el catálogo.

La instancia vía API exhibió un proceso de optimización estable y coherente en los tres modelos evaluados, con métricas que se sostuvieron en niveles elevados de rendimiento a lo largo de las iteraciones. En cambio, la instancia local mostró un comportamiento más heterogéneo: XGBoost y LightGBM presentaron alta estabilidad, con diferencias de apenas 0.08 % y 0.09 % entre la mejor y la peor configuración, respectivamente, mientras que Extra Trees evidenció una degradación progresiva a partir de la iteración 15, con la exactitud descendiendo desde aproximadamente 95.5 % hasta 92.27 % en las iteraciones finales.

3.3 Coherencia del razonamiento

En la instancia vía API, las justificaciones textuales del agente reflejaron de manera consistente la evolución del historial, con argumentos explícitos y plausibles acerca de los ajustes propuestos. En la instancia local, este comportamiento también se observó en los casos de XGBoost y LightGBM, cuyas explicaciones acompañaron adecuadamente la dinámica de mejora y estabilización de las métricas.

No obstante, en el caso de Extra Trees se observó un deterioro de esta coherencia en etapas avanzadas del proceso. En particular, el agente continuó proponiendo modificaciones de carácter exploratorio sin evidencias claras de convergencia, aun cuando el historial mostraba una tendencia descendente sostenida.

4 Conclusiones y trabajo futuro

Los resultados preliminares validan el uso de LLMs como agentes autónomos para la optimización de clasificadores supervisados. En ambas modalidades de despliegue, el prototipo logró alcanzar y superar marginalmente el rendimiento reportado por un experto humano, sin conocimiento previo del dominio más allá de la descripción semántica del conjunto de datos y con plena conformidad estructural en todas las propuestas generadas.

Las diferencias observadas entre instancias sugieren que la calidad del modelo de lenguaje utilizado impacta directamente en la estabilidad del proceso iterativo y en la capacidad del agente para consolidar estrategias de explotación una vez identificadas regiones de alto rendimiento.

Como trabajo futuro inmediato, se planea ejecutar el experimento sobre conjuntos de datos adicionales para evaluar la generalización de los resultados, incorporar métricas de costo computacional y analizar la relación entre la longitud del contexto y la degradación del razonamiento en modelos locales.

Referencias

Ali, M., et al. (2023). A survey of AutoML methods and tools. *Soft Computing*.

Azevedo, P., et al. (2024). Limitations of classical AutoML: A critical review. *Machine Learning Journal*.

Fernandez, R. R. (2021). *Métodos de ensamblado en machine learning*. Universidad Santiago de Compostela.

Google Education. (2024). *Google Prompting Essentials: TCREI Framework*. Google LLC.

Huang, Z., et al. (2026). Semantic-aware AutoML: Beyond mathematical optimization. *Preprint*.

Luo, H., et al. (2025). AutoM3L: Multimodal machine learning via LLM orchestration. In *Proceedings of ICML*.

Martínez-Plumed, F., et al. (2022). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8).

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, 22–31.

Noviana, A., & Sela, R. (2024). Performance comparison of random forest and logistic regression in predicting time deposit customers with feature selection. In *Proceedings of ICAICTA 2024*.

Xu, Z., et al. (2025). LLMs as optimization agents for machine learning pipelines. In *NeurIPS Workshop*.