




Meta-aprendizaje para AutoML heterogéneo

Meta-learning for heterogeneous AutoML

Lia de la Concepción Zerquera Ferrer¹ , Alberto Fernández Oliva^{*2} , Alejandro Piad Morffis³ ,
Suilan Estévez Velarde⁴ 

Resumen El aprendizaje de máquina automatizado (AutoML) es un área de la Inteligencia Artificial en auge, aunque enfrenta varios desafíos. Este proceso puede ser lento e ineficiente computacionalmente. El meta-aprendizaje, que consiste en aprender de experiencias pasadas mediante algoritmos aplicados a diversos tipos de datos, puede mejorar AutoML al identificar los mejores algoritmos para problemas específicos, acelerando así el proceso y mejorando los resultados. Esta investigación propone una estrategia de meta-aprendizaje para dominios genéricos de aprendizaje automático, capaz de abordar una variedad amplia de problemas mediante la selección de características adecuadas. Se utiliza AutoGOAL como complemento para AutoML, ya que ofrece soluciones efectivas en múltiples dominios y permite crear flujos de algoritmos que generan una base de conocimientos útil para el meta-aprendizaje. El enfoque facilita la adquisición de conocimiento a partir de la ejecución de distintos problemas en AutoGOAL y evalúa el rendimiento de los flujos de algoritmos correspondientes. Con esta información, se desarrolla un modelo que ayuda a descartar flujos inadecuados para futuros problemas. Los resultados experimentales indican que esta estrategia puede reducir significativamente el tiempo de ejecución en AutoGOAL, permitiendo identificar rápidamente flujos erróneos.

Palabras Clave: aprendizaje de máquina, AutoML, meta-aprendizaje.

Abstract Automatic machine learning (AutoML) is a field of Artificial Intelligence that has gained a lot of popularity recently, however, it still faces many challenges. It is time consuming and can be computationally inefficient. Meta-learning, the process of learning from past experiences by using learning algorithms to learn different types of data, can support the AutoML process from the best algorithms to solve a particular type of problem, which speeds up the process and achieves better results in the same amount of time. In this research, a meta-learning strategy for generic domains of machine learning is designed. The approach can address a wide variety of automatic learning problems by selecting a set of characteristics to represent each problem. AutoGOAL is used as a complementary system to AutoML, due to its ability to provide efficient solutions in a wide range of domains. It allows the establishment of algorithm flows from which a knowledge base is generated to perform meta-learning. This approach facilitates the acquisition of knowledge from the execution of various machine learning problems in AutoGOAL and the performance of the algorithm flows tested by AutoGOAL for these problems. With this knowledge, a model is designed to discard algorithm flows, whose application is not suitable for future automatic learning problems. Experimental evaluation shows that the meta-learning strategy can reduce the execution time of AutoGOAL, which allows to quickly detect erroneous flows.

Keywords: machine learning, AutoML, meta-learning.

Mathematics Subject Classification: 68, 68T05, 68T30.

¹Departamento de Computación, Facultad de Matemática y Computación, Universidad de La Habana, Cuba. Email: liazerquera@gmail.com.

²Departamento de Computación, Facultad de Matemática y Computación, Universidad de La Habana, Cuba. Email: afoliva55@gmail.com.

³Departamento de Computación, Facultad de Matemática y Computación, Universidad de La Habana, Cuba. Email: apiad@apiad.net.

⁴Departamento de Computación, Facultad de Matemática y Computación, Universidad de La Habana, Cuba. Email: sestevez@matcom.uh.cu.

*Autor para Correspondencia (Corresponding Author)

Editado por (Edited by): Damian Valdés Santiago, Facultad de Matemática y Computación, Universidad de La Habana, Cuba.

Citar como: Zerquera Ferrer, L.C., Fernández Oliva, A., Piad Morffis, A., & Estévez Velarde, S. (2024). Meta-aprendizaje para AutoML heterogéneo. *Ciencias Matemáticas*, 36(Único), 79–85. DOI: <https://doi.org/10.5281/zenodo.14164788>. Recuperado a partir de <https://revistas.uh.cu/rcm/article/view/9130>.

Introducción

El aprendizaje de máquina automatizado (AutoML) se ha convertido en un tema de tendencia en el ámbito de la Inteligencia Artificial (IA). AutoML abarca un conjunto de técnicas para automatizar y facilitar el proceso de implementación, ex-

perimentación y despliegue de algoritmos de aprendizaje automático. Uno de los principales desafíos de muchos sistemas de AutoML es su incapacidad para aprovechar la experiencia adquirida en la resolución de problemas al enfrentarse a nuevas tareas. [1]. Para intentar solucionar esto, se han comenzado a

utilizar técnicas de meta-aprendizaje (*meta-learning*) con el objetivo de que las herramientas de AutoML sean capaces de encontrar buenas soluciones a nuevos problemas presentados, de forma más rápida y basándose en la experiencia.

El meta-aprendizaje ha sido aplicado con gran éxito en varios campos de la IA, donde los volúmenes de datos eran muy grandes y se hacía necesario buscar una relación entre ellos. Algunos de estos campos son la robótica [3], el aprendizaje no supervisado [4] y la medicina inteligente [8]. En el campo del AutoML también se suele trabajar con grandes *datasets* y grandes espacios de búsqueda, razón por la cual parece atractivo incorporar el meta-aprendizaje en esta área.

De las herramientas más conocidas dentro del campo de AutoML se encuentran Auto-Weka [11], Auto-Sklearn Hyperopt-Sklearn [5], TPOT [10], ML-Plan [7], H2O AutoML [6], AutoGluon [9] y AutoGOAL [2]. Entre estas, Auto-Sklearn es la única que aplica técnicas de meta-aprendizaje, aunque no se descarta que el resto de ellas lo hagan en próximas versiones. Sin embargo, esta herramienta usa un enfoque denominado *warm-starting*, que ha obtenido buenos resultados cuando los datos a procesar están en forma tabular, lo cual impone una restricción significativa para su uso.

Por otro lado, las herramientas de AutoML requieren procesar datos de diferente naturaleza, como imágenes y textos. Como antecedentes a esta investigación, en el grupo de IA de la Facultad de Matemática y Computación de la Universidad de La Habana, se creó AutoGOAL [2], que es una herramienta de AutoML, muy versátil en cuanto a la gran variedad de tipos de problemas que puede resolver y a la diversidad de formatos en los que es capaz de recibir los *datasets*.

AutoGOAL es una herramienta competitiva dentro del campo del AutoML, ya que devuelve soluciones que brindan buen rendimiento a los problemas que se le presentan. Sin embargo, está sujeta a mejoras en cuanto al tiempo y el consumo de recursos a la hora de encontrar soluciones, precisamente, esa es una de las razones por la cual se le quiere integrar enfoques de meta-aprendizaje. Se conoce que la técnica *warm-starting* [12] tiene como deficiencia que solo se encarga de empezar la búsqueda de soluciones a problemas de aprendizaje automático en un punto o “lugar” prometedor y luego no vuelve a intervenir en dicho proceso de búsqueda. Por ello, en el presente trabajo se utilizará un enfoque de meta-aprendizaje basado en características de los *datasets*, que intervenga activamente en el proceso de optimización de AutoGOAL.

Relevancia del estudio

Esta investigación constituye un paso para la mejora de las herramientas de AutoML, lo cual es crucial tanto tecnológica como ambientalmente. Se utiliza un enfoque de meta-aprendizaje basado en características de los *datasets*, que intervenga activamente en el proceso de optimización de AutoGOAL. Al automatizar y optimizar la creación de modelos de aprendizaje automático se reduce el tiempo y los recursos necesarios. Además, al reutilizar experiencias previas, se evita

la redundancia y el desperdicio de recursos.

1. Propuesta y experimentación

Para incorporar meta-aprendizaje a AutoGOAL se llevaron a cabo dos etapas principales:

- Etapa de extracción del meta-conocimiento: se utilizó un conjunto de *datasets* heterogéneo y como herramienta complementaria AutoGOAL, donde se tuvo en cuenta su capacidad para dar solución a una amplia gama de problemas.
- Etapa de aplicación del meta-conocimiento al proceso de optimización de AutoGOAL: un modelo de regresión lineal, entrenado con los datos extraídos en la etapa anterior, permite identificar y descartar flujos “malos” en el momento de hacer la evaluación, sin que esto implique análisis computacionales costosos.

1.1 Extracción del meta-conocimiento

La adquisición del meta-conocimiento se realiza mediante la extracción de características de un conjunto de *datasets*, mientras que la extracción de información se realizó con los flujos de algoritmos que probó AutoGOAL en dichos *datasets*. Un flujo de algoritmos (*pipeline*) es un conjunto de algoritmos que se deben ejecutar, uno a continuación de otro, para resolver un problema representado en un determinado *dataset*. De cada flujo, se extrae el nombre de cada algoritmo junto con sus hiperparámetros, respetando el orden en que aparecen en el mismo. Además, se extrae la evaluación del flujo de acuerdo a la métrica determinada por el usuario. En ocasiones surgen errores a la hora de evaluar el flujo, en este caso, se reporta dicho error para posteriormente ser guardado y asociado al problema ejecutado junto a dicho flujo (Figura 1).

Para la extracción de meta-características se ejecutó en AutoGOAL un conjunto de 100 *datasets* tabulares y 100 de imágenes tomados de *OpenML*¹ y 100 *datasets* de texto tomados de *Hugging Face*² (Figura 2).

1.2 Aplicación del meta-conocimiento

Una vez construido el conjunto de datos, las meta-características de los *datasets* y de los *pipelines* ejecutados, se construye un vector con todas ellas para adaptar dicha información a un formato que pueda entender un algoritmo de *machine learning*. Para crear dicho vector se concatenan las meta-características de los *datasets* con la vectorización de la representación en *string* de la función objetivo y de un *pipeline*. Por último, se concatena a lo anterior la evaluación de la función objetivo, de donde se obtiene un vector donde todos sus elementos son numéricos.

Una vez que se tienen todos los vectores, se separan en tres conjuntos según el tipo de *dataset* (tabular, imágenes y texto).

¹OpenML es una plataforma abierta para compartir conjuntos de datos, algoritmos y experimentos.

²<https://huggingface.co/datasets>.

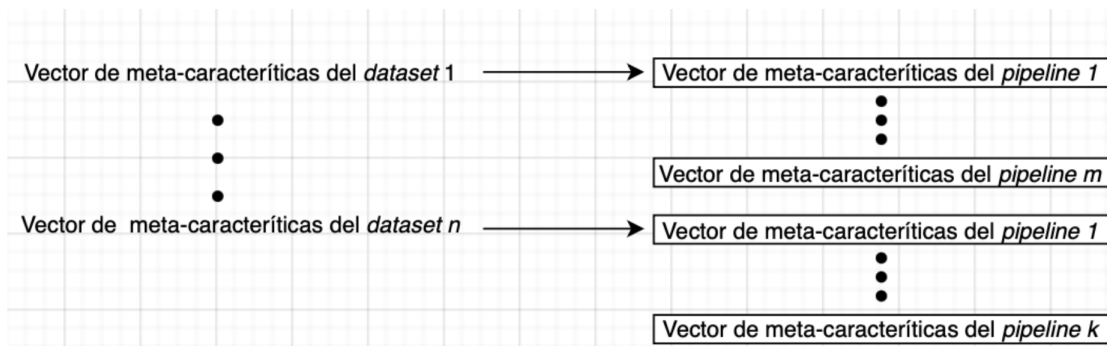


Figura 1. Representación del *dataset* y del *pipeline* en la base de datos [Representation of the dataset and the pipeline in the database].

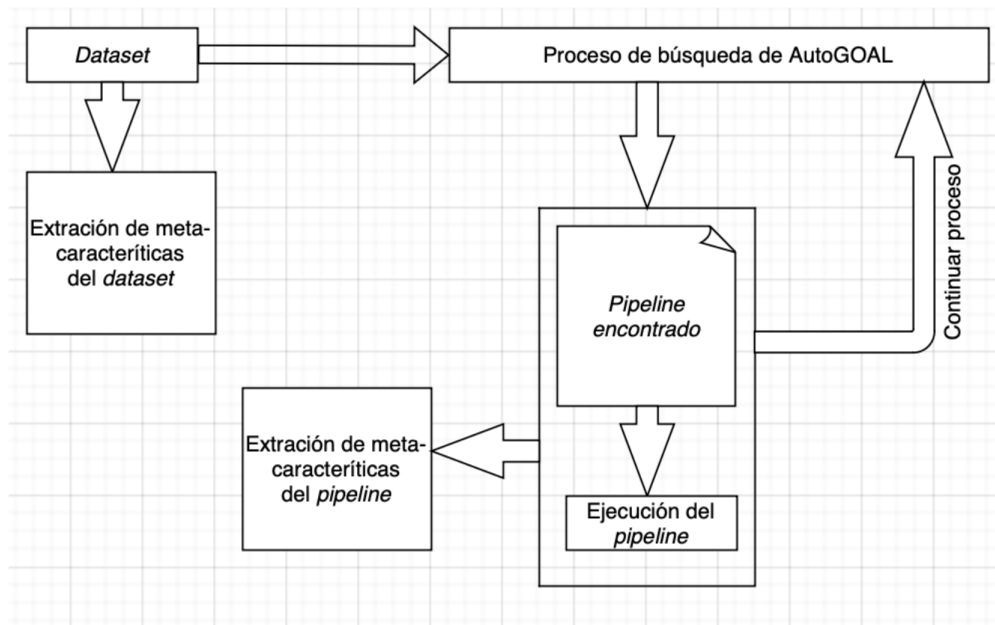


Figura 2. Extractor de características [Feature extractor].

Luego cada uno de estos conjuntos se separa en dos subconjuntos, uno de entrenamiento y otro de prueba. Finalmente, se entrena un modelo de *machine learning* con ayuda de AutoGluon. Para ello, se le proporcionan a dicha herramienta, de uno en uno, todos los conjuntos de entrenamiento que fueron determinados. Por cada conjunto de entrenamiento, se obtiene un modelo de *machine learning*, que es evaluado por el conjunto de prueba correspondiente y determinado anteriormente. De esta manera se obtienen los modelos de *machine learning* para predecir el rendimiento de un *pipeline*, dado un *dataset* en forma tabular, o en forma de texto.

Los modelos anteriormente mencionados son incorporados al proceso de búsqueda de AutoGOAL. Cada vez que la herramienta encuentra un *pipeline* válido para dar solución al problema que se está tratando de resolver en ese momento, las meta-características asociadas a dicho *pipeline* y al problema, son suministradas al conjunto de modelos que fueron incorporados al proceso para que se active, convenientemente,

uno de ellos, en dependencia del tipo de *dataset* sobre el cual esté representado el problema (Figura 3).

El modelo activado devuelve un número que representa la predicción del rendimiento del *pipeline*. Si el rendimiento de este es menor que un umbral determinado por el usuario (el umbral por defecto es la mitad de la mejor evaluación encontrada hasta el momento), entonces no pasará a la fase de evaluación y, en su lugar, se utiliza la predicción del modelo como su rendimiento real. Sin embargo, si el rendimiento del *pipeline* es mayor que el umbral, entonces, AutoGOAL realiza el proceso de evaluación original, sin modificación alguna, y sesga el espacio de búsqueda según los resultados de dicha evaluación.

2. Experimentación

En esta sección se evalúa el comportamiento de AutoGOAL con la integración del meta-aprendizaje. Para ello,

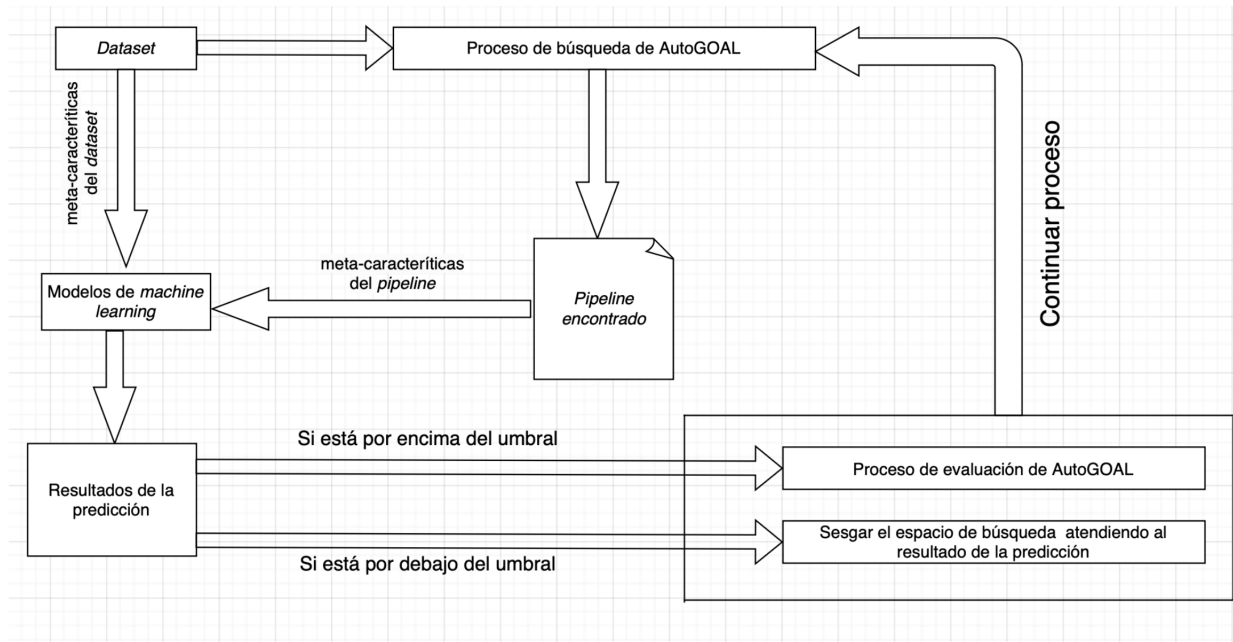


Figura 3. Incorporación de los modelos *Integration of the models*.

se examina la eficiencia de dicha herramienta haciendo uso de meta-aprendizaje en comparación con AutoGOAL, sin su incorporación. Se ejecutaron en AutoGOAL cinco *datasets* que representan problemas de aprendizaje automático clásicos (por ejemplo, Cars, Credit G, Abalone, Yeast, HAHA). Los *datasets* para estos experimentos fueron seleccionados porque ya habían sido usados en estudios previos con versiones anteriores de AutoGOAL [2].

En las figuras 4 y 5 se observa que Cars, Credit G y Yeast, sin meta-aprendizaje, obtienen el mejor rendimiento. Sin embargo, con la incorporación de *meta-learning* se consigue que todos los *pipelines* evaluados obtengan una precisión por encima del umbral (mayor que la mitad de la mejor precisión obtenida).

En el caso de Abalone (Figura 5), a diferencia de Cars y Credit G, AutoGOAL con meta-aprendizaje encuentra el flujo de algoritmos que tiene la mejor precisión, en comparación con todos los demás flujos que fueron evaluados con y sin meta-aprendizaje.

En la Figura 6 se observa el comportamiento de HAHA con meta-aprendizaje y sin él. Su comportamiento es similar al de Cars, ya que el meta-aprendizaje no consigue evaluar ningún *pipeline* por debajo del umbral. Sin embargo, no logra igualar ni mejorar la mejor precisión conseguida por AutoGOAL sin meta-aprendizaje.

2.1 Resultados

Luego de ver el comportamiento de AutoGOAL con meta-aprendizaje puede surgirle al lector la siguiente interrogante: ¿existe alguna diferencia significativa en cuanto a eficiencia entre AutoGOAL con meta-aprendizaje y sin meta-aprendizaje?

Ambos enfoques son capaces de encontrar modelos de

aprendizaje que se comportan de manera eficiente ante un problema de aprendizaje de máquina. Si embargo, si la comparación se hace teniendo en cuenta el tiempo de ejecución para dar solución a cada problema, en los ejemplos ejecutados se observa que AutoGOAL con meta-aprendizaje encuentra de forma más rápida modelos que resuelvan de forma eficiente problemas de aprendizaje de máquina.

Para evaluar la capacidad de los modelos de aprendizaje automático creados se realizaron varios experimentos que consistieron en guardar todos los flujos de algoritmos que son descartados y evaluados por los modelos. Se eliminaron los *pipelines* que no se pudieron evaluar en tiempo de ejecución por cualquiera de los posibles motivos (por ejemplo, exceden el espacio en memoria, el tiempo de ejecución o por errores propios de adaptaciones hechas por AutoGOAL a los algoritmos de algunas bibliotecas).

Los experimentos muestran que los modelos de aprendizaje automático creados tienen un comportamiento eficiente. En todos los problemas ejecutados se obtiene una precisión y un recobrado por encima de 0,90. Esto se corrobora con la evaluación de la métrica F1 en la que se aprecia que existe un balance entre el total de *pipelines* adecuados y los detectados como tal por el modelo (Figura 6).

Conclusiones

En esta investigación se ha creado una nueva estrategia de meta-aprendizaje que se integró a AutoGOAL, la cual, al considerar el conocimiento alcanzado en ejecuciones previas de problemas de aprendizaje automático en AutoGOAL, el propio problema de aprendizaje automático que se analiza y los *pipelines* encontrados durante el proceso de búsqueda, des-

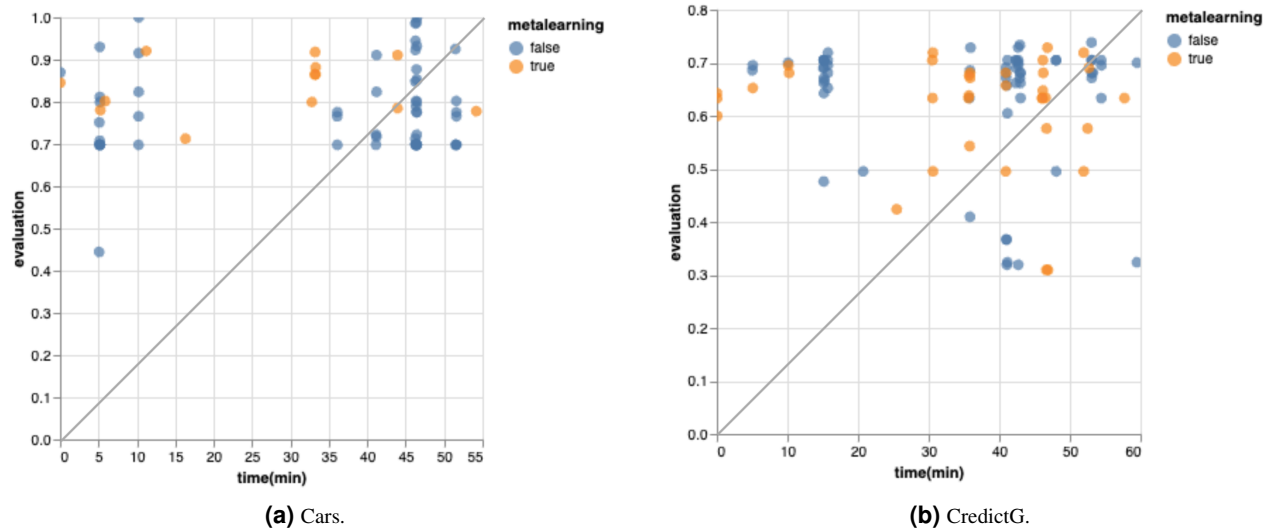


Figura 4. Resultados de ejecutar Cars y Credit G en AutoGOAL, con y sin meta-aprendizaje [Results of running Cars and Credit G in AutoGOAL with and without meta-learning].

carta *pipelines* que no son buenos. Esta estrategia se concibe a partir de algoritmos computacionalmente viables de extracción de meta-características y del entrenamiento de modelos de aprendizaje de máquina.

AutoGOAL se destaca por su capacidad para generar soluciones eficaces para una gran cantidad de dominios de problemas, pero dicha herramienta aún no contaba con un sistema de meta-aprendizaje. Por tanto, uno de los aportes de esta investigación fue la incorporación de un sistema de meta-aprendizaje a AutoGOAL. Para ello, fue necesario la implementación de un extractor de meta-características. Dicho extractor es capaz de analizar problemas de distinta configuración, o sea, problemas de texto, imágenes y tabulares. Un aspecto a destacar de dicho extractor es que el usuario puede añadir nuevas características para la representación vectorial de un problema de aprendizaje automático.

A partir del conocimiento obtenido con el extractor de meta-características se entrenaron modelos de aprendizaje automático, lo cual permitió descartar, en tiempo de ejecución, *pipelines* inadecuados. Con esto se logra reducir el tiempo de búsqueda que requiere AutoGOAL para encontrar soluciones factibles a problemas de aprendizaje automático.

La evaluación experimental de la propuesta se realizó utilizando un conjunto de problemas clásicos de aprendizaje automático, tomadas de investigaciones previas de AutoGOAL. Se tuvieron en cuenta los *pipelines* encontrados por AutoGOAL, con y sin la incorporación de meta-aprendizaje. Esto se hizo con el objetivo de ver qué versión encontraba *pipelines* con mejor rendimiento ante determinados problemas de aprendizaje automático. Se constató que, en la mayoría de los problemas analizados, AutoGOAL con meta-aprendizaje y sin meta-aprendizaje logran un comportamiento similar, aunque en algunos casos, AutoGOAL con meta-aprendizaje encuentra

mejores *pipelines*.

Recomendaciones

El enfoque de meta-aprendizaje presentado es capaz de abordar una gran variedad de problemas de aprendizaje automático. Sin embargo, aún se encuentra en una etapa de desarrollo inicial, por lo que es necesario seguir mejorando sus capacidades de precisión al momento de descartar *pipelines* malos.

El conocimiento adquirido fue añadido a AutoGOAL en el proceso de búsqueda. Sin embargo, existen otros momentos en que se puede añadir dicho conocimiento, por ejemplo, dicho conocimiento puede ser utilizado para inicializar el espacio de búsqueda y seguir una estrategia de *warm-starting*. De esta manera, se le proporciona a cada nuevo problema presentado al sistema de AutoML, un espacio de búsqueda donde inicialmente los pesos de las aristas del grafo que constituye dicho espacio de búsqueda estarán dados por la experiencia previa acumulada, a partir de problemas similares al que se analiza.

En esta investigación se entrenó un modelo a partir de los datos para descartar *pipelines* inadecuados. Sería interesante que en futuras investigaciones se entrenara un modelo para predecir los *pipelines* que no son posibles evaluar, según los recursos computacionales con los que se está ejecutando.

En esta investigación se diseñaron y se crearon extractores de meta-características y modelos para problemas de texto y tabulares. Por lo que es necesario que en futuras investigaciones se escale esta propuesta a otros dominios donde los problemas estén representados en *datasets* de otro tipo.

Suplementos

Este artículo no contiene información suplementaria.

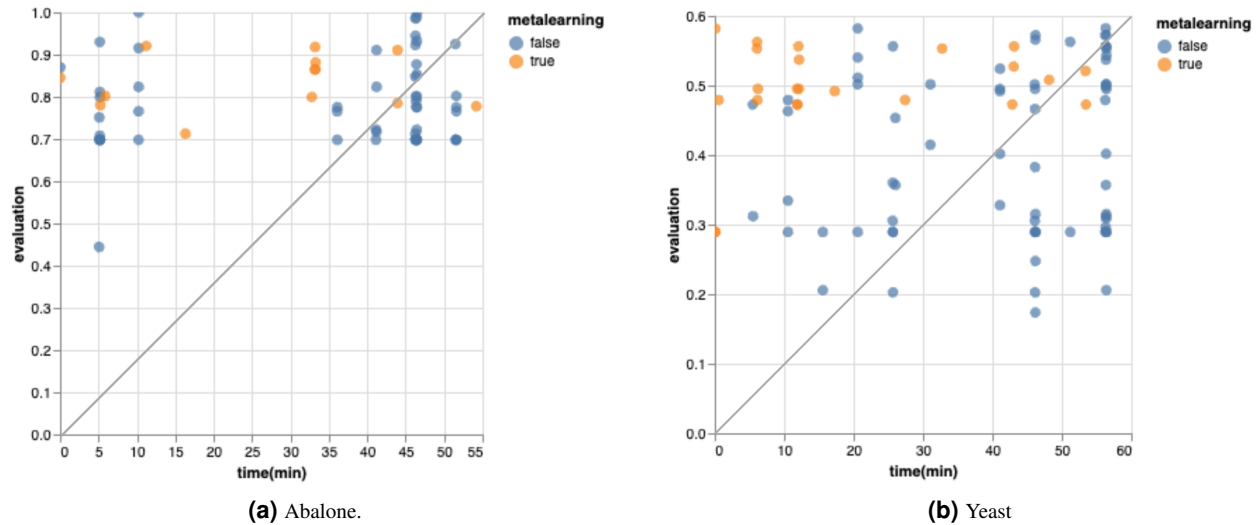


Figura 5. Resultados de ejecutar Abalone y Yeast en AutoGOAL, con y sin meta-aprendizaje [Results of running Abalone and Yeast in AutoGOAL, with and without meta-learning].

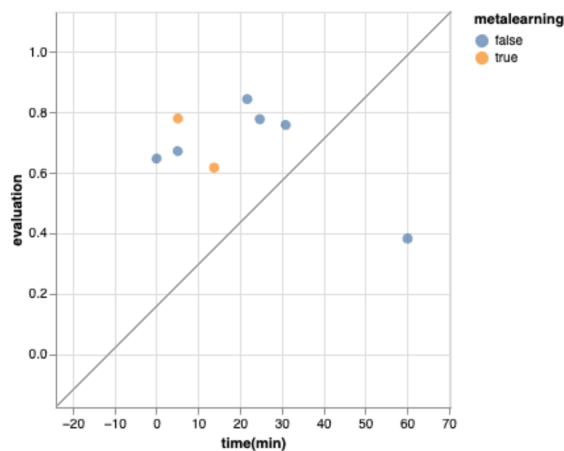


Figura 6. Resultados de ejecutar Haha con y sin meta-aprendizaje [Results of running Haha with and without meta-learning].

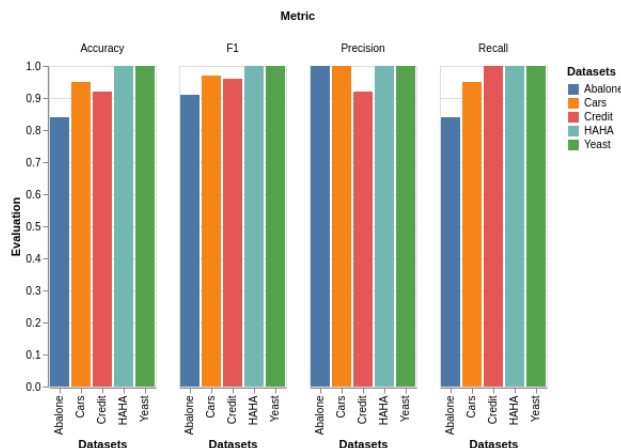


Figura 7. Rendimiento del modelo [Model performance].

Conflictos de interés

Se declara que no existen conflictos de interés. No hubo subvenciones involucradas en este trabajo.

Contribución de autoría

Conceptualización L.C.F.Z, S.E.V

Curación de datos L.C.F.Z, S.E.V., A.P.M.

Análisis formal L.C.F.Z, A.F.O, S.E.V., A.P.M.

Investigación L.C.F.Z, A.F.O, S.E.V., A.P.M.

Metodología L.C.F.Z, A.F.O.

Administración de proyecto L.C.F.Z, A.F.O.

Recursos L.C.F.Z, A.F.O, S.E.V.

Software L.C.F.Z, S.E.V.

Supervisión L.C.F.Z, A.F.O, A.P.M.

Validación L.C.F.Z, A.F.O, S.E.V., A.P.M.

Visualización L.C.F.Z, A.F.O

Redacción: preparación del borrador original L.C.F.Z, A.F.O.

Redacción: revisión y edición L.C.F.Z, A.F.O.

Referencias

- [1] Drori, I., Y. Krishnamurthy, R. Rampin, R. de Paula Lourenco, J. Piazzentin Ono, K. Cho, C. Silva, and J. Freire: *Alphad3m: Machine learning pipeline synthesis*, 2021. <https://arxiv.org/abs/2111.02508>.
- [2] Estevanell Valladares, E.L.: *AutoGOAL, un sistema de Auto-ML Heterogéneo*. Tesis en opción al grado de

- Licenciado en Ciencia de la Computación, Facultad de Matemática y Computación, Universidad de La Habana, 2020.
- [3] Finn, C., T. Yu, Zhang T., P. Abbeel, and S. Levine: *One-shot visual imitation learning via meta-learning*, 2017. <https://arxiv.org/abs/1709.04905>.
 - [4] Garg, V.K.: *Supervising Unsupervised Learning*. 2018. <https://proceedings.neurips.cc/paper/2018/file/72e6d3238361fe70f22fb0ac624a7072-Paper.pdf>.
 - [5] Komer, Brent: *Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn*. 2013. <https://conference.scipy.org/proceedings/scipy2014/komer.html>.
 - [6] LeDell, E.: *H2O AutoML: Scalable Automatic Machine Learning*. 2020. <https://api.semanticscholar.org/CorpusID:221338558>.
 - [7] Mohr, R.F.: *ML-Plan: Automated machine learning via hierarchical planning*. 2018. <https://link.springer.com/content/pdf/10.1007/s10994-018-5735-z.pdf>.
 - [8] Nguyen, B.D.: *Overcoming data limitation in medical visual question answering*. 2019. https://link.springer.com/chapter/10.1007/978-3-030-32251-9_57.
 - [9] Nick, E., J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola: *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data*. arXiv preprint arXiv:2003.06505, 2020. <https://arxiv.org/abs/2003.06505>.
 - [10] Olson, R.S.: *TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning*. 2016. https://link.springer.com/content/pdf/10.1007/978-3-030-05318-5_8.pdf.
 - [11] Thornton, C.: *Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms*. 2013. <https://dl.acm.org/doi/10.1145/2487575.2487629>.
 - [12] Truger, F., J. Barzen, M. Bechtold, M. Beisel, F. Leymann, A. Mandl, and V. Yussupov: *Warm-Starting and Quantum Computing: A Systematic Mapping Study*. ACM Computing Surveys, 56:1–31, 2023. <https://api.semanticscholar.org/CorpusID:257482369>.

