Tipo de artículo: Artículo original

Una solución para visualizar modelos de agrupamiento, correlación y regresión en Grafana

A solution for visualizing clustering, correlation and regression models in Grafana

Cristian Páez Olcha [1], https://orcid.org/0009-0001-8137-261X
Katherine Ramírez Hidalgo [1], https://orcid.org/0009-0008-1266-0918
Daniel Alejandro Deyne Rodríguez [1], https://orcid.org/0009-0001-6508-9603
Alejandro Rosete Suárez [1,2*], https://orcid.org/0000-0002-4579-3556

[1] Facultad de Ingeniería Informática, Universidad Tecnológica de La Habana "José Antonio Echeverría". Cuba.
[2] Avangenio S.R. L. Cuba.

*Autor para la correspondencia. rosete@ceis.cujae.edu.cu

RESUMEN

El proceso de ingeniería de datos se centra en proporcionar información valiosa que facilite la toma de decisiones basadas en datos, siendo la visualización un elemento clave en este contexto. Grafana es una de las herramientas de código abierto más poderosas disponibles para la visualización de datos. Sin embargo, aunque ofrece múltiples ventajas, no incluye gráficos específicos para representar modelos complejos de agrupamiento, correlación y regresión. El siguiente trabajo propone una solución que permite visualizar modelos de agrupamiento, correlación y regresión utilizando Grafana. Se representaron los modelos en una

base de datos relacional y se utilizaron componentes de Grafana para su visualización. Además de describir la solución propuesta, se analizó un caso de estudio que ilustra sus beneficios y demuestra su aplicabilidad en escenarios reales. Como resultado, se lograron diversas visualizaciones de cada algoritmo con elementos interactivos y dinámicos, que permitieron ilustrar los modelos obtenidos y comprender las relaciones entre los datos.

Palabras clave: agrupamiento; correlación; Grafana; regresión; visualización de datos.

ABSTRACT

The data engineering process focuses on providing valuable information that facilitates data-driven decision making, with visualization being a key element in this context. Grafana is one of the most powerful open source tools available for data visualization. However, although it offers multiple advantages, it does not include specific graphics to represent complex clustering, correlation and regression models. The following work proposes a solution that allows visualizing clustering, correlation and regression models using Grafana. The models were represented in a relational database and Grafana components were used for their visualization. In addition to describing the proposed solution, a case study was analyzed that illustrates its benefits and demonstrates its applicability in real scenarios. As a result, various visualizations of each algorithm were achieved with interactive and dynamic elements, which allowed illustrating the obtained models and understanding the relationships between the data.

Keywords: clustering; correlation; Grafana; regression; data visualization.

Recibido: 04/01/2025
Aceptado: 24/03/2025
En línea: 01/04/2025

Introducción

La ingeniería de datos se encarga de preparar y transformar grandes volúmenes de información para que puedan ser procesados y analizados de manera efectiva (Reis and Housley, 2022). La ingeniería de datos se enfoca en proporcionar los datos que luego serán consumidos para la toma de decisión. Con datos bien

estructurados y accesibles, las herramientas de visualización cobran relevancia, ya que permiten interpretar y comunicar de manera clara los patrones, tendencias y relaciones subyacentes. El uso adecuado de gráficos interactivos permite explorar y descubrir patrones ocultos que podrían no ser evidentes en tablas o resúmenes numéricos (Skiena, 2017).

En el ámbito del aprendizaje automático, técnicas como el agrupamiento, la regresión y la correlación permiten extraer patrones y relaciones significativas dentro de los datos. El agrupamiento se utiliza para segmentar un conjunto de datos en grupos homogéneos, permitiendo identificar patrones y relaciones sin la necesidad de etiquetas previas. Los modelos de agrupamiento, como K-medias o K-medoides, ayudan a clasificar datos según características similares, proporcionando una visión más clara de la estructura interna de los datos. La regresión, por otro lado, se enfoca en modelar la relación entre variables dependientes e independientes, permitiendo predecir valores continuos basados en variables explicativas. Finalmente, el análisis de correlación mide la relación entre dos o más variables, identificando posibles dependencias lineales o no lineales que pueden ser clave para entender las dinámicas subyacentes en los datos (Hernández Orallo et al., 2004).

Existen una amplia variedad de herramientas y bibliotecas disponibles para el aprendizaje automático, que ofrecen diversas funcionalidades para facilitar el desarrollo, entrenamiento y visualización de modelos. Estas herramientas abarcan desde soluciones generales para la manipulación y análisis de datos hasta opciones más especializadas que permiten una interpretación detallada de los resultados generados por los modelos. La elección de una herramienta u otra depende de factores como la complejidad del modelo, la necesidad de interacción en las visualizaciones, el tipo de datos a analizar y el nivel de personalización que se requiera (Kadam and Akhade, 2024).

Para los modelos de regresión, las visualizaciones más comunes incluyen gráficos de dispersión con líneas de tendencia, que muestran la relación entre las variables independientes y dependientes, así como gráficos de residuos para evaluar la calidad del ajuste del modelo. En el caso de la correlación, los mapas de calor son una de las técnicas más populares para visualizar matrices de correlación, donde los valores de correlación entre pares de variables se representan mediante colores de intensidad variable. Esto facilita la identificación rápida de relaciones fuertes o débiles entre variables. Herramientas como Matplotlib y Seaborn en Python, o ggplot2 en R, son ampliamente utilizadas para generar estas representaciones. Además, los gráficos de pares

permiten visualizar distribuciones conjuntas y correlaciones entre múltiples variables, lo que es especialmente útil en el análisis exploratorio de datos (Müller and Guido, 2017; Hassan Sial et al., 2021; Wickham and Grolemund, 2016; Fox and Weisberg, 2019).

Para los modelos de agrupamiento, las técnicas de visualización se centran en representar la estructura y distribución de los grupos identificados. Los gráficos de dispersión con colores diferenciados para cada clúster son una opción común, especialmente cuando se trabaja con dos o tres dimensiones. Para conjuntos de datos de mayor dimensionalidad, es posible utilizar gráficos de coordenadas paralelas, que permiten visualizar múltiples dimensiones en un solo gráfico. Esto facilita la identificación de patrones y agrupamientos en espacios multidimensionales. Otra alternativa son los gráficos de radar, que representan múltiples variables en un formato circular. Herramientas como Matplotlib en Python, así como Tableau y R, son ampliamente utilizadas para implementar estas técnicas. Además, los dendrogramas son útiles en el agrupamiento jerárquico, ya que muestran la estructura de los clústeres en forma de árbol, permitiendo identificar relaciones y distancias entre grupos de manera intuitiva (Müller and Guido, 2017; Kassambara, 2017; Machairidou, 2018).

Grafana es reconocida como una de las herramientas más avanzadas y versátiles para la visualización de datos, destacándose por ser gratuita y de código abierto y se alinea muy bien con procesos de ingeniería de datos. Esta plataforma permite la creación de paneles altamente interactivos y personalizables, utilizados principalmente para el monitoreo de sistemas, el análisis de métricas de rendimiento y la supervisión de datos en tiempo real. Su arquitectura, flexible y extensible, la convierte en una herramienta adaptable a diversos contextos. Gracias a sus capacidades para integrar múltiples fuentes de datos y aplicar filtros avanzados, Grafana facilita la visualización dinámica e interactiva, lo que la posiciona como una excelente opción para la exploración y análisis de datos complejos (Grafana Labs, 2025).

No obstante, a pesar de sus muchas capacidades, Grafana presenta limitaciones en cuanto a la visualización de modelos de agrupamiento, regresión y correlación. La extensión Business Charts, proporciona algunas pocas posibilidades. Por una parte aporta un diagrama de dispersión que puede ser útil para visualizar los resultados de agrupamiento, pero presenta la importante limitación de solo ser aplicable a datos bidimensionales, sin incluir detalles como los centroides. De manera similar, ofrece un gráfico de regresión lineal, cuyo uso se restringe a gráficos bidimensionales, lo que limita su capacidad para representar de forma

adecuada modelos de regresión más complejos o análisis multidimensionales (Grafana Labs, 2025; Apache ECharts, 2025; Volkov Labs, 2025).

Dada la situación problemática planteada, se define como objetivo: desarrollar paneles en Grafana para representar modelos de agrupamiento, específicamente K-medias, K-medoides y agrupamiento jerárquico, así como regresión lineal y logística, y análisis de correlación utilizando los métodos de Pearson y Spearman. Se tomaron como referencia los gráficos que ofrecen las herramientas mencionadas anteriormente, con el fin de diseñar y crear los paneles en Grafana.

A continuación, se presenta la metodología empleada, que describe el proceso seguido para desarrollar la solución, desde el preprocesamiento de datos hasta su visualización en Grafana. Se presenta el modelo lógico de la base de datos diseñado para almacenar los modelos, así como el aprovechamiento de las funcionalidades de Grafana para la creación de gráficos interactivos y personalizables. Posteriormente, se ilustran los resultados obtenidos a través de un caso de estudio sobre el análisis de comentarios textuales relacionados con hoteles, destacando las ventajas y desventajas de la solución. Finalmente, se incluyen las conclusiones.

Métodos o Metodología Computacional

Descripción de la solución

El flujo de trabajo consta de varias etapas, desde el preprocesamiento de los datos hasta la visualización de los resultados. A continuación, se describe cada fase del proceso:

1. **Preprocesamiento de datos**: el primer paso consiste en el preprocesamiento de los datos. Este proceso implica limpiar, transformar y preparar los datos para que estén listos para el entrenamiento del modelo. Las tareas comunes en esta fase incluyen la eliminación de valores nulos, la normalización de características, la codificación de variables categóricas y la conversión de los datos en un formato adecuado para el análisis. Para el preprocesamiento, se utilizan bibliotecas como Pandas para manipular datos tabulares, NumPy para operaciones matemáticas y Scikit-learn para escalar y transformar características (Stancin and Jovic, 2019).
2. **Entrenamiento del modelo:** con los datos listos, se procede al entrenamiento del modelo**.** Dependiendo del tipo de problema a resolver (por ejemplo, clasificación, regresión, agrupamiento), se selecciona el

algoritmo adecuado y se ajustan los parámetros para optimizar su rendimiento. En esta etapa, se emplean bibliotecas como Scikit-learn, Statsmodels y SciPy (Scikit-Learn, 2025; SciPy, 2025; Statmodels, 2025).

3. **Conexión con PostgreSQL y almacenamiento del modelo:** una vez entrenado el modelo, es necesario almacenarlo para su uso posterior. El modelo se guarda en una base de datos PostgreSQL para su fácil recuperación y uso futuro. Esto se realiza mediante la conexión a la base de datos utilizando herramientas como Psycopg2. El modelo es almacenado en un formato específico para cada algoritmo que es explicado en secciones posteriores (Psycopg, 2021).
4. **Visualización en Grafana:** una vez almacenado el modelo, el siguiente paso es visualizar los resultados en un panel interactivo de Grafana. En este punto, se diseña un panel personalizado que permita representar de manera comprensible los datos generados por el modelo. A continuación, se describen los aspectos principales para implementar la visualización en Grafana:
    - **Conexión de Grafana con PostgreSQL:** para realizar la visualización, se configura Grafana para conectarse a la base de datos PostgreSQL, donde se almacenan los resultados generados por el modelo. La fuente de datos de Grafana permite extraer la información necesaria para las visualizaciones, mediante consultas SQL.
    - **Diseño del panel personalizado:** el diseño del panel debe ser adaptado a las características específicas de cada modelo entrenado y los tipos de resultados que se desean visualizar. Dependiendo del modelo, se eligen los gráficos más adecuados para facilitar la comprensión de los resultados y la interpretación del modelo.
    - **Interactividad y filtros:** Grafana permite agregar elementos interactivos al panel. Por ejemplo, se pueden agregar filtros para seleccionar entre diferentes colores, símbolos, tamaños y otros atributos visuales, mejorando la personalización y la comprensión de los datos. Esto ofrece una experiencia de visualización más flexible y personalizada, adaptándose a las necesidades específicas de análisis.

La Figura 1 describe el diagrama de secuencia que modela las interacciones entre los distintos componentes del sistema. La herramienta Grafana, implementa un enfoque Modelo-Vista-Controlador (MVC) que se ve reflejado en el diagrama.

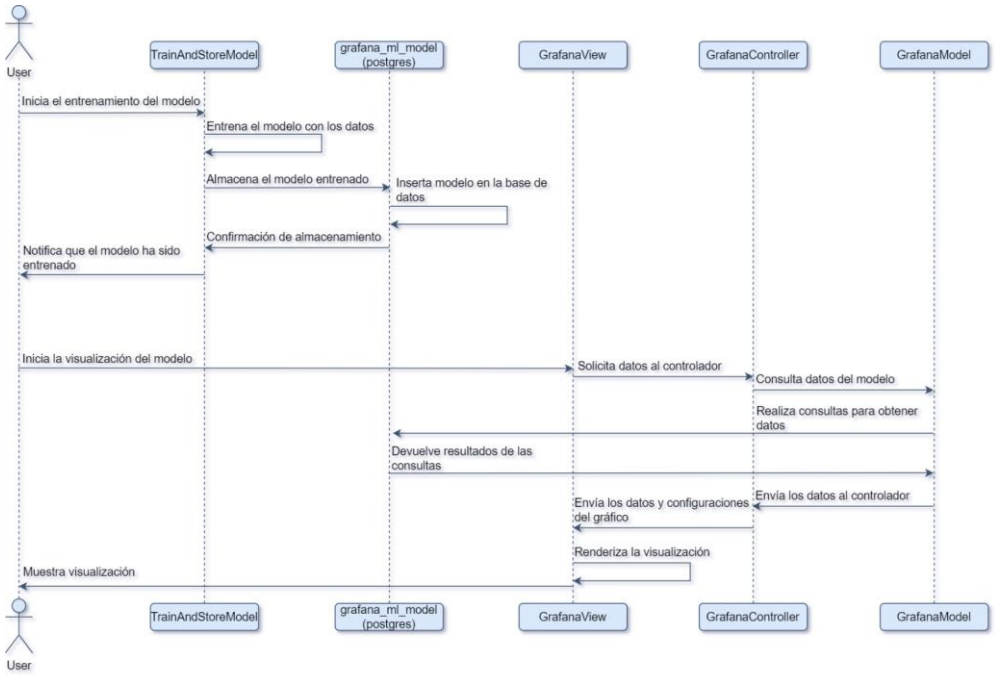

Fig. 1 – Diagrama de secuencia.

El proceso se divide en dos partes fundamentales, la fase de entrenamiento y almacenamiento realizada en Python y PostgreSQL, y posteriormente la fase de visualización en Grafana.

Proceso de entrenamiento

1. El usuario proporciona los datos necesarios e inicia el proceso de entrenamiento del modelo.
2. Se realiza el entrenamiento del modelo en Python utilizando los datos proporcionados por el usuario.
3. Una vez completado el entrenamiento, el modelo se envía al sistema de gestión de base de datos para su almacenamiento.
4. El sistema de gestión de base de datos inserta los datos del modelo en las tablas correspondientes.
5. El sistema de gestión de base de datos confirma que el modelo ha sido almacenado correctamente.
6. Finalmente, el usuario es notificado de que el modelo ha sido entrenado y almacenado exitosamente.

Proceso de visualización

1. El usuario inicia la visualización del modelo a través de la vista en Grafana.
2. La vista solicita al controlador los datos necesarios para la visualización.
3. El controlador se comunica con el modelo para obtener la información del modelo entrenado.

4. El modelo realiza las consultas necesarias a PostgreSQL para obtener los datos almacenados.
5. PostgreSQL devuelve los datos solicitados al modelo.
6. El modelo transfiere los datos obtenidos al controlador.
7. El controlador envía los datos y configuraciones necesarias para renderizar el gráfico a la vista.
8. La vista renderiza el gráfico utilizando los datos y la configuración proporcionada por el controlador.
9. Finalmente, la vista muestra al usuario la visualización del modelo.

Diseño para la persistencia

La Figura 2 presenta el esquema de base de datos diseñado para almacenar y gestionar los datos relacionados con los conjuntos de datos y los modelos generados mediante algoritmos de aprendizaje automático. Su estructura permite registrar tanto los datos de entrada como los resultados de los modelos, facilitando la organización, consulta y evaluación de los mismos.

Esencialmente, existe una tabla principal, *grafana_ml_model_index*, que define el contexto general de cada caso de estudio y se relaciona con el resto de las tablas. Para cada caso, se registran las instancias y sus características mediante las tablas *grafana_ml_model_point*, *grafana_ml_model_feature* y *grafana_ml_model_point_value*, lo que permite representar la información en diagramas de dispersión, gráficos de coordenadas paralelas y otras visualizaciones. Cada modelo cuenta con una o varias tablas diseñadas para su posterior recuperación. Por ejemplo, *grafana_ml_model_correlation* almacena referencias a dos características, indicando la correlación entre ellas y su tipo (Pearson o Spearman). Todos los detalles de implementación están disponibles en Git Hub en el código que se ofrece como parte de esta propuesta.

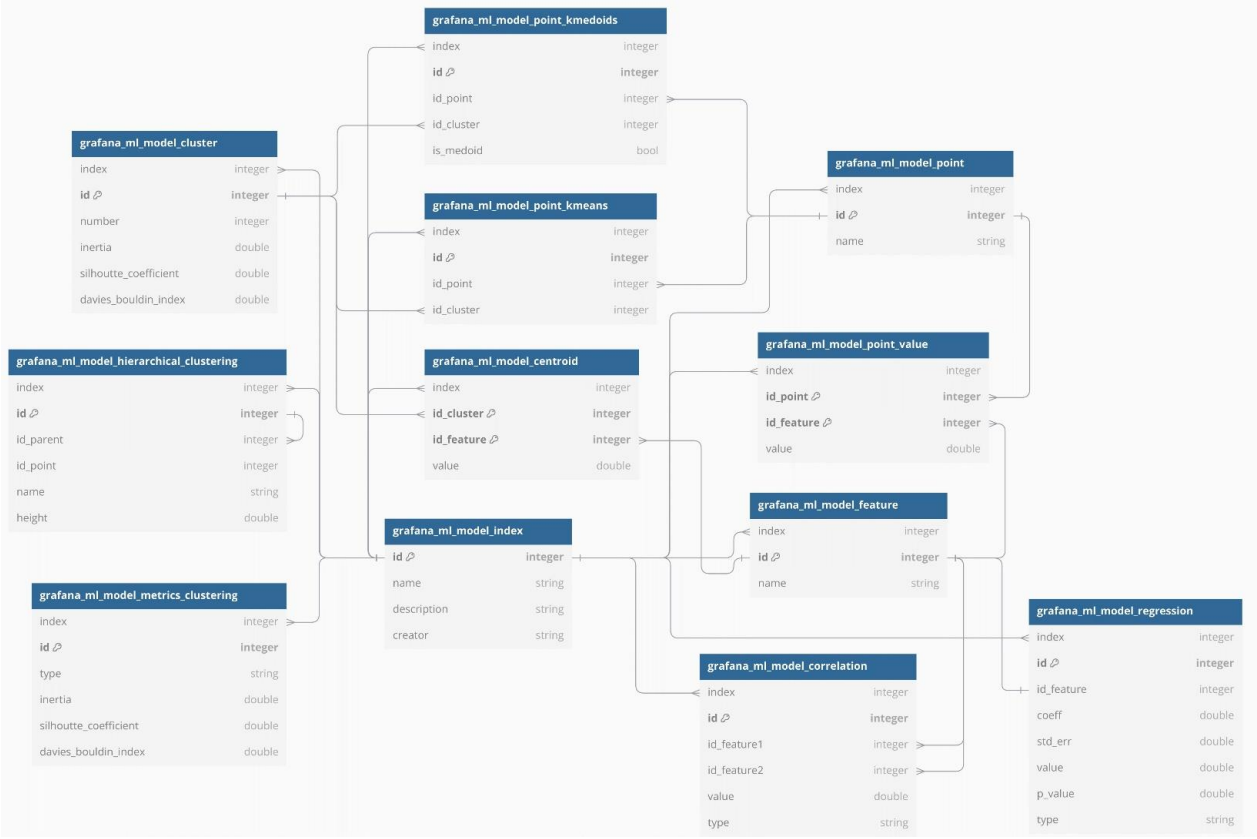

Fig. 2 – Diseño lógico de la base de datos.

Adaptación de los gráficos de Business Charts

La mayoría de las visualizaciones fueron creadas utilizando la popular extensión de Grafana, Business Charts. Esta permite integrar gráficos creados con la biblioteca Apache ECharts en los paneles de Grafana, proporcionando una solución poderosa y flexible para la visualización de datos. Apache ECharts es una biblioteca de código abierto, que destaca por su accesibilidad y capacidad para crear visualizaciones avanzadas (Apache ECharts, 2025).

Entre sus principales características están la amplia variedad de gráficos disponibles, como de líneas, barras, dispersión y mapas de calor. Ofrece una personalización avanzada, permitiendo a los usuarios ajustar desde elementos visuales básicos hasta interacciones complejas como zoom y animaciones. También facilita la exploración interactiva de datos, con herramientas como leyendas dinámicas y consejos emergentes,

ayudando a profundizar en los detalles sin perder la visión general (Apache ECharts, 2025; Volkov Labs, 2025).

Ajustes en la visualización

Para hacer más flexibles los modelos y permitir que estos paneles se puedan integrar de mejor manera a la visualidad que requiera un proyecto concreto de Ingeniería de Datos, se han aprovechado las potencialidades de las variables de Grafana que permiten cambiar valores en los paneles sin necesidad de editar manualmente las consultas cada vez. Estas variables se definen a nivel del panel y permiten a los usuarios cambiar el contenido dinámicamente (McCollam, 2024). Se han implementado un conjunto de variables entre las cuales destacan:

- **Variable para cambiar el caso:** permite cambiar el conjunto de datos que se está analizando sin necesidad de modificar manualmente las consultas en cada panel. Esto es útil cuando se trabaja con múltiples casos de estudio.
- **Variable para cambiar el algoritmo:** se utiliza para cambiar el algoritmo de análisis utilizado en las consultas. Permite variar entre K-medias y K-medoides, entre regresión lineal y regresión logística, entre correlación de Pearson y correlación de Spearman.
- **Variable para colores:** permite cambiar los colores utilizados en los gráficos para mejorar la interpretación y la estética. Puede ser una selección de colores fijos o paletas de colores.
- **Variable para símbolos en diagramas de dispersión:** permite cambiar los símbolos utilizados en los diagramas de dispersión, como círculos, rectángulos, triángulos, entre otros.
- **Variable para el tamaño de los símbolos:** permite ajustar el tamaño de los símbolos en los diagramas de dispersión, ayudando a destacar ciertos datos o mejorar la visualización.
- **Variable para elegir el número de clústeres en el agrupamiento jerárquico:** ofrece una visualización dinámica que ajusta automáticamente la agrupación de los datos según el número de clústeres especificado, facilitando la exploración y análisis de diferentes configuraciones.

Código fuente

La solución implementada está disponible en GitHub en el siguiente enlace: https://github.com/KatherineRamirezH/grafana-machine-learning. Este repositorio contiene las clases Python

utilizadas para el entrenamiento y almacenamiento de los modelos, el *script* para la creación de la base de datos y los paneles de Grafana.

Resultados y discusión

Para la validación de la solución, se visualizaron varios modelos obtenidos a partir de conjuntos de datos reales relacionados con análisis de comentarios textuales sobre hoteles. A continuación, se describe uno de los casos de estudio analizados. Luego de un preprocesamiento que incluyó eliminación de valores faltantes y normalización, se obtuvo un conjunto de datos con 13347 instancias y 26 características, entre estas se incluyen:

- Indicadores binarios: 20 campos que reflejan la presencia (1) o ausencia (0) de palabras clave específicas en los comentarios, tales como: hotel, resort, staff, beach, cuba, bar, food, comida, playa, great, bien, servicio, personal, always, room, service, todos, amazing, buffet, excelente.
- l_comentario: longitud del comentario en caracteres.
- dia_de_semana: día de la semana en el cual se emitió el comentario.
- mes: mes en el cual se emitió el comentario.
- dia_del_mes: día del mes en el cual se emitió el comentario.
- val_h: indica la polaridad del comentario en relación al hotel, basada en una valoración humana. Tiene 3 clases posibles: 1 (positiva), 0 (neutral) o -1 (negativa).
- val_llm: polaridad general del comentario tras un análisis de sentimiento usando modelos lingüísticos. También tiene 3 clases posibles: 1 (positiva), 0 (neutral) o -1 (negativa).

Para este conjunto de datos, se define *val_h* y *val_llm* como las variables objetivo para aquellos algoritmos que lo requieren como la regresión logística. En el caso de la regresión cada variable objetivo fue analizada de manera independiente, adaptándolas a 2 clases posibles: 1 (positiva) o 0 (negativa o neutra). A continuación, se presentan las visualizaciones correspondientes a los análisis de correlación, agrupamiento y regresión.

La Figura 3 muestra las correlaciones entre las variables en una matriz de correlación, donde se puede observar que la gran mayoría de las correlaciones son débiles o moderadas. La Figura 4 muestra las correlaciones

ordenadas, entre una característica seleccionada por el usuario y las demás variables del conjunto de datos. En este caso, se ha seleccionado la variable *val_h*. Se puede observar que las correlaciones positivas más fuertes están asociadas con *val_llm*, *excelente* y *servicio*, mientras que las correlaciones negativas más destacadas se encuentran con *l_comentario*, *room* y *buffet*.

Por ejemplo, esto puede permitir a un responsable del hotel detectar que un posible problema con las habitaciones y el buffet es que se relaciona con comentarios negativos, los cuales suelen ser más largos, mientras que el servicio parece ser una fortaleza del hotel. Es importante resaltar la fuerte correlación de aproximadamente 0.86 entre las dos variables objetivo, *val_h* (evaluación humana) y *val_llm* (análisis de sentimiento automático). Esta alta correlación sugiere que las valoraciones generadas por modelos lingüísticos coinciden en gran medida con las evaluaciones humanas, validando así la consistencia y fiabilidad del análisis de sentimiento.

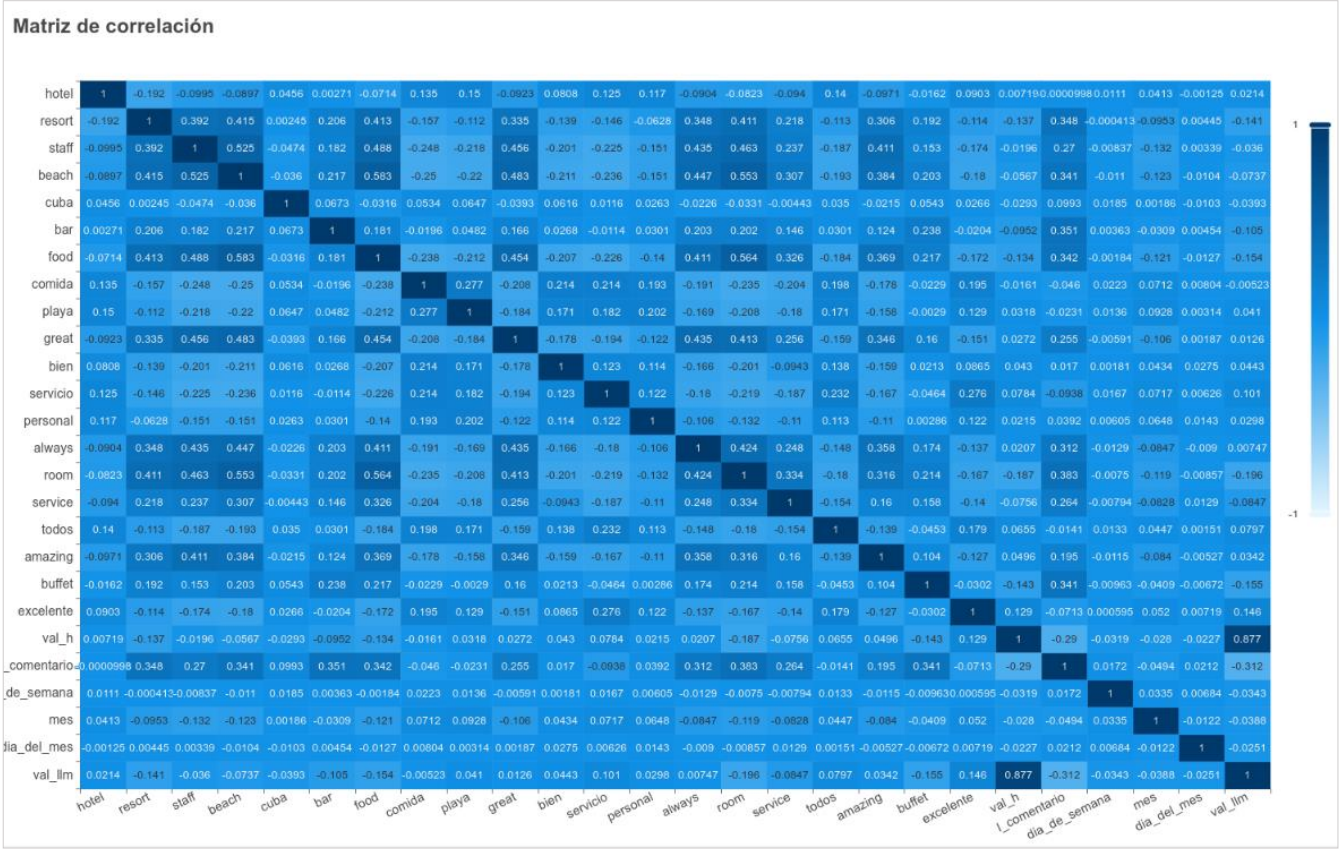

Fig. 3 – Correlación de Pearson en una matriz de correlación.

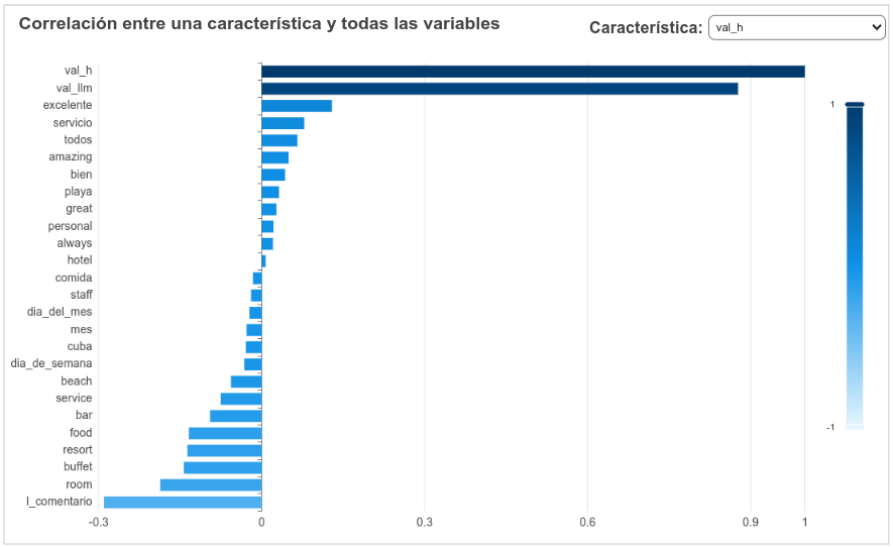

Fig. 4 – Correlación entre *val_h* y el resto de las variables.

La Figura 5 muestra los coeficientes y la desviación estándar del modelo de regresión logística utilizando como variable objetivo *val_h*. Las variables que se encuentran más alejadas de 0 (eje vertical rojo) son las más significativas, mientras que aquellas más cercanas a 0 son las menos significativas. Teniendo en cuenta esto, se puede concluir que las palabras que aumentan significativamente la probabilidad de clasificar un comentario como positivo son: *excelente, always, amazing* y *great*. Por otro lado, las palabras que reducen significativamente la probabilidad de que un comentario sea clasificado como positivo incluyen *room, food, comida,* y, además, la longitud del comentario.

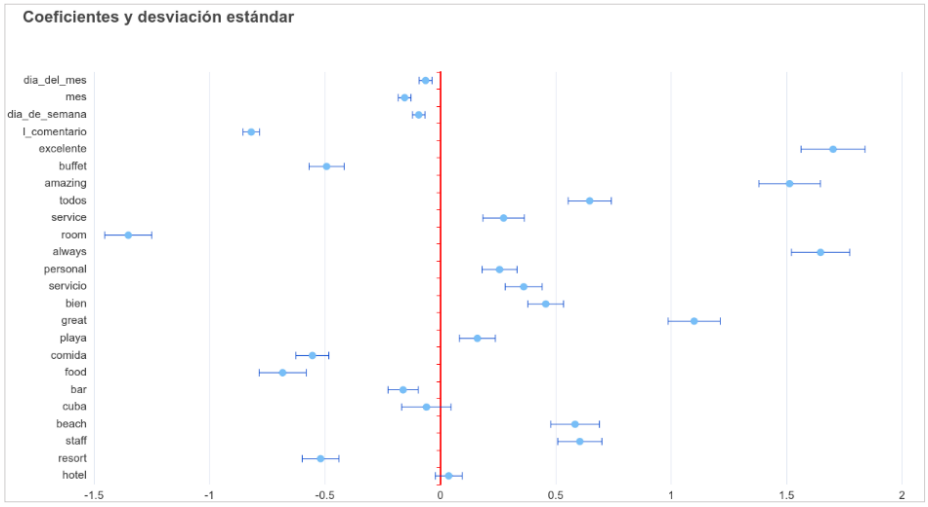

Fig. 5 – Modelo de regresión para predecir *val_h*.

De manera similar la Figura 6 muestra los coeficientes y la desviación estándar del modelo de regresión logística utilizando como variable objetivo *val_llm*. Los resultados son muy similares, se mantienen las mismas variables mencionadas anteriormente como las más significativas para el modelo.

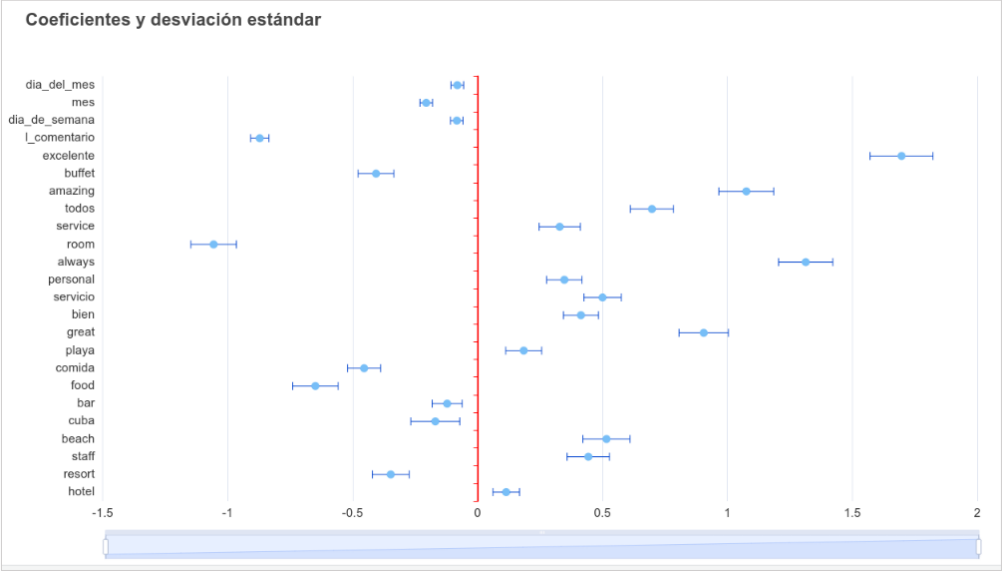

Fig. 6 – Modelo de regresión para predecir *val_llm*.

Otra manera de visualizar los coeficientes y la desviación estándar del modelo de regresión se representa en la Figura 7, en este caso el valor del coeficiente es representado mediante una barra y la desviación estándar mediante una línea. Por otro lado, en la Figura 8 se muestra la matriz de confusión que permite evaluar la eficiencia del modelo. En estas dos visualizaciones se tuvo en cuenta como variable objetivo *val_h*.

Fig. 7– Modelo de regresión en un gráfico de barras con líneas de error.

Fig. 8– Matriz de confusión.

La Figura 9 presenta la visualización del agrupamiento utilizando el algoritmo K-medias en un gráfico de coordenadas paralelas, lo que permite observar las características de los centroides de cada grupo. El clúster 0 tiende a agrupar comentarios predominantemente en inglés, ya que contiene palabras como *food, great, always, room, service,* con valores más altos en comparación con los otros clústeres. Además, los comentarios en este clúster tienden a ser más largos. Por otro lado, los clústeres 1 y 2 muestran una tendencia similar entre sí, representando comentarios en español. Estos comentarios suelen ser más cortos en comparación con el clúster 0. Además, ambas valoraciones (*val_llm* y *val_h*) se acercan más a 1 en estos clústeres, lo que sugiere que, en promedio, los comentarios en estos grupos tienden a ser más positivos.

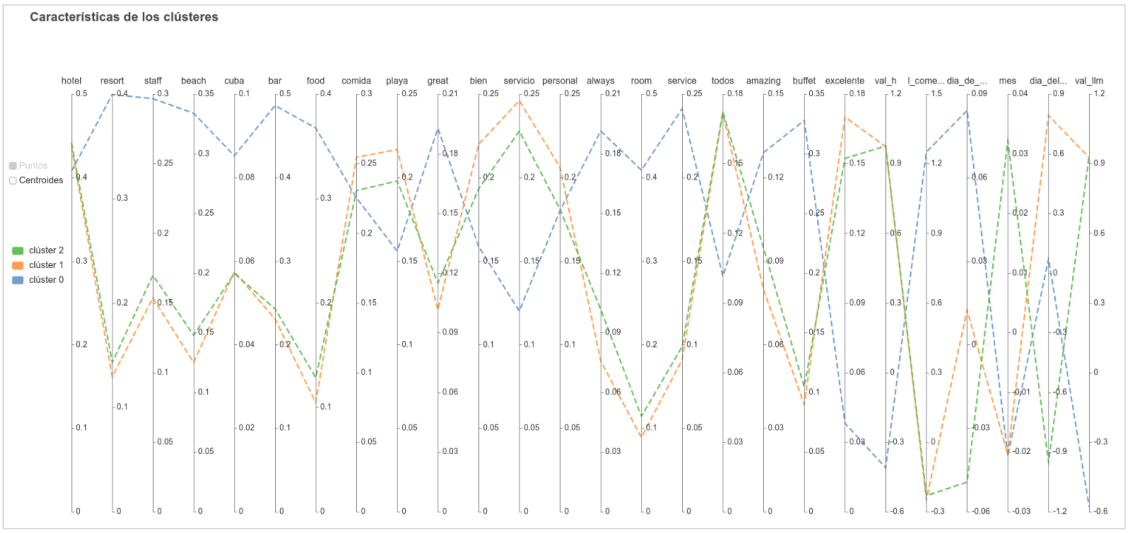

Fig. 9 – Características de los centroides en un gráfico de coordenadas paralelas (K-medias).

La Figura 10 presenta un diagrama de dispersión que ilustra la relación entre dos características seleccionadas y el clúster, representado mediante distintos colores. En este caso se han elegido la longitud del comentario y el día del mes en el cual fue emitido. Ambas características fueron estandarizadas. La longitud del comentario se observa bastante similar en los clústeres 1 y 2, y más largos en el clúster 0. Con respecto al día del mes se observa que el clúster 0 no sigue un patrón de distribución definido, mientras que el clúster 1 tiende a agrupar comentarios emitidos en la segunda quincena del mes y el clúster 2, en la primera quincena. Esta misma distribución del día del mes se puede visualizar en la Figura 11 a través de un diagrama de caja.

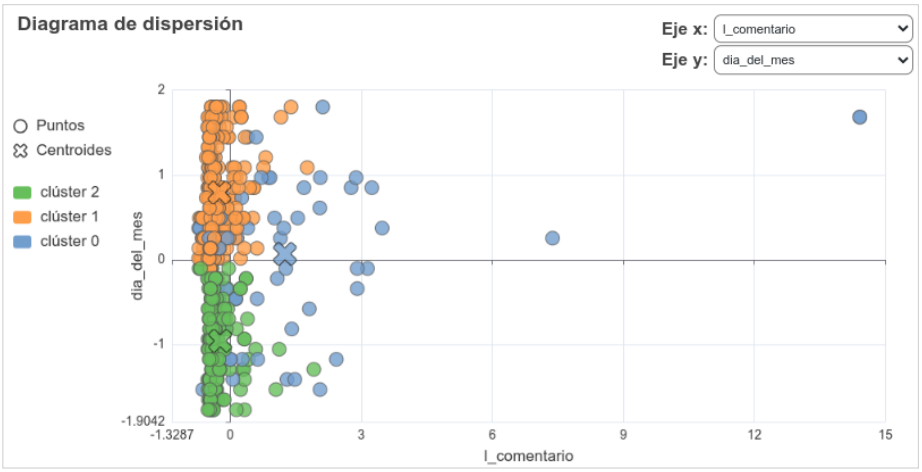

Fig. 10 – Agrupamiento K-medias en un diagrama de dispersión.

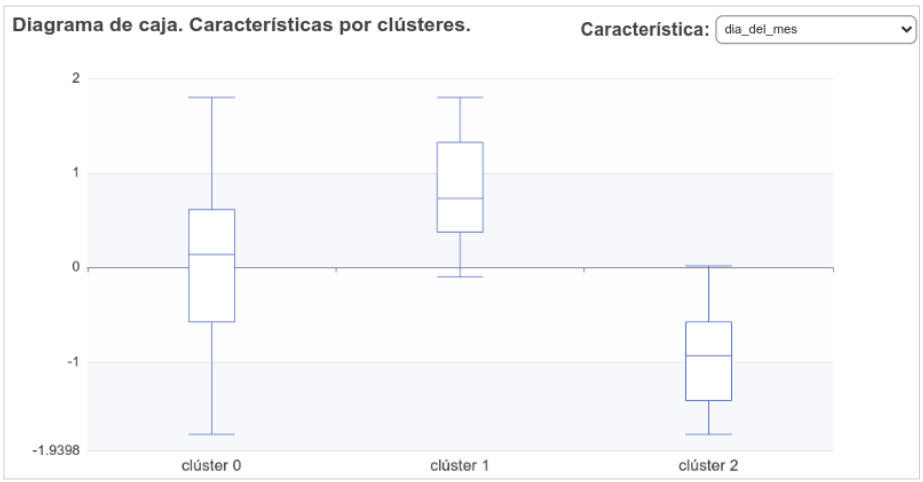

Fig. 11 – Distribución de la característica *dia_del_mes* para cada grupo en un diagrama de caja.

Por otro lado, el dendrograma resultante de agrupamiento jerárquico se muestra en la Figura 12. Este permite visualizar como se van agrupando todos los elementos del conjunto de datos en función de su similitud o altura. En la figura 13 se observan las características de los centroides de cada grupo. El clúster 2 tiene valores altos en palabras en inglés, sin embargo, a diferencia del resultado en K-medias, las valoraciones no son las más negativas. En este caso, el clúster 1 agrupa valoraciones más negativas en promedio, con valores más altos en *comida*, *dia_del_mes* y *mes*. El clúster 0 que abarca la mayoría de los datos es el que tiene las valoraciones más positivas en promedio.

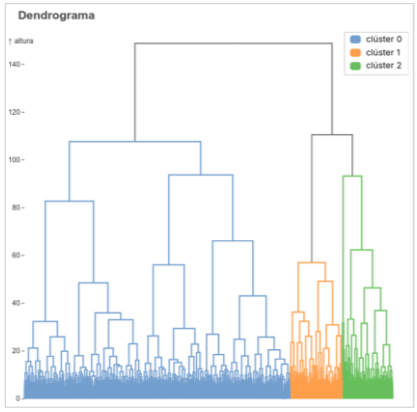

Fig. 12 – Dendrograma obtenido mediante el agrupamiento jerárquico.

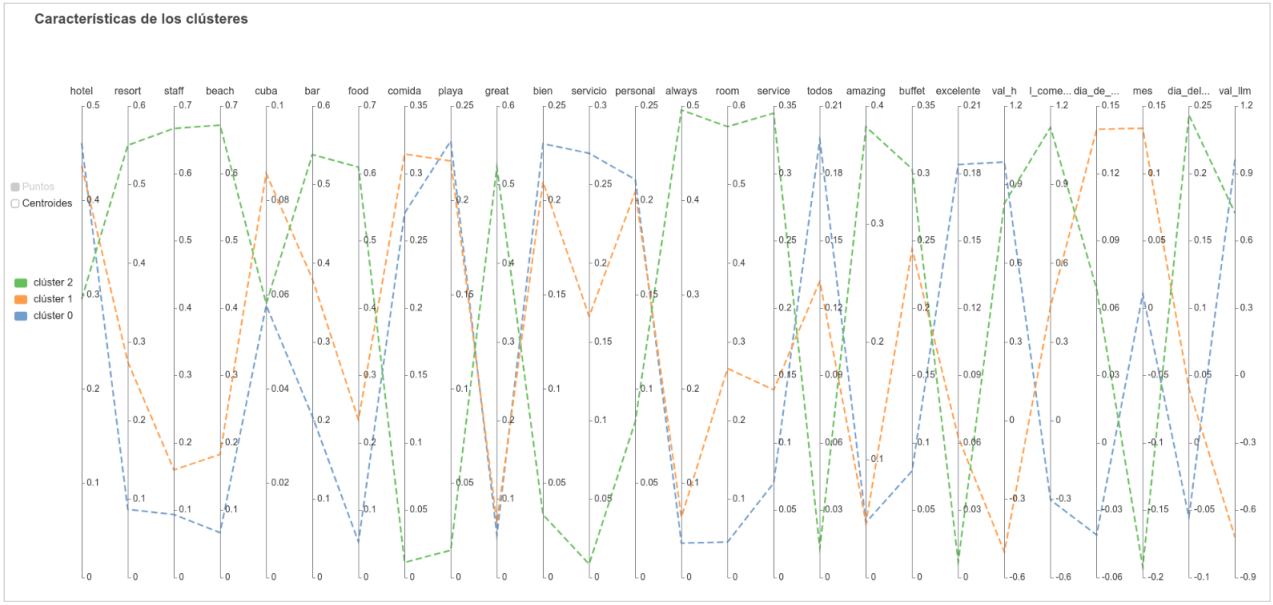

Fig. 13 – Características de los centroides en un gráfico de coordenadas paralelas (agrupamiento jerárquico).

Entre las principales ventajas que ofrece la solución propuesta se encuentran:

- **Uso de tecnologías de código abierto:** la solución se fundamenta en herramientas de código abierto, lo que proporciona una alta flexibilidad y escalabilidad, además de facilitar su integración con otras plataformas y tecnologías. Este enfoque promueve una evolución continua de la solución, libre de restricciones comerciales o licencias costosas, favoreciendo la adaptabilidad y la colaboración en la comunidad de investigación.
- **Facilidad en la gestión de casos de estudio:** el diseño de la base de datos y la implementación de filtros permite cambiar entre distintos casos de estudio, sin la necesidad de configuraciones adicionales. Esta característica optimiza el tiempo requerido para realizar análisis comparativos entre diferentes conjuntos de datos.
- **Variedad en las visualizaciones:** la solución ofrece múltiples tipos de gráficos, tales como gráficos de dispersión, mapas de calor, diagramas de caja, gráficos de barras y gráficos de coordenadas paralelas. Esta diversidad de visualizaciones permite representar una amplia gama de patrones y relaciones de datos, lo que facilita la interpretación y comunicación de los resultados en diferentes contextos analíticos. Todos estos gráficos son ajustables para lograr una apariencia que se requiera.

Por otro lado, esta solución presenta la desventaja del rendimiento con grandes volúmenes de datos. Por ejemplo, un gráfico de agrupamiento que incluya la visualización de millones de puntos puede demorar mucho. Esta es una línea de trabajo futuro.

Conclusiones

El trabajo muestra cómo es posible extender Grafana para que también pueda ser empleada como una herramienta para la visualización de modelos de aprendizaje automático dentro del proceso de Ingeniería de Datos. La propuesta, ha permitido la incorporación de paneles personalizados y la inclusión de elementos interactivos que mejoran significativamente la experiencia del usuario, permitiendo un análisis profundo y una interpretación de los resultados para modelos de agrupamiento, regresión y correlación.

De este modo, se han adaptado las potencialidades de visualización de la extensión *Business Charts* para representar modelos de aprendizaje automático, cumpliendo los objetivos previstos.

Referencias

Reis, J.; Housley, M. Fundamentals of Data Engineering. 1 st ed. Sebastopol: O´Reilly; 2022. 403 p.

Skiena, S.S. The Data Science Design Manual. 1st ed. New York: Stony Brook University; 2017. 730 p.

Hernández Orallo, J.; Ramírez Quintana, J.; Ferri Ramírez, C. Introducción a la Minería de Datos. 1st ed. Madrid: Pearson Educación.S.A; 2004. 656 p.

Kadam, A.J.; Akhade, K. A Review on Comparative Study of Popular Data Visualization Tools. ALOCHANA JOURNAL. 2024;13(4): p. 532-538.

Müller, A.C.; Guido, S. Introduction to Machine Learning with Python. 1 st ed. Sebastopol: O´Reilly; 2017. 376 p.

Hassan Sial, A.; Shah Rashdi, S.Y.; Hafeez Khan, A. Comparative Analysis of Data Visualization Libraries Matplotib and Seaborn in Python. International Journal of Advanced Trends in Computer Science and Engineering. 2021;10(1): p. 277-281.

Wickham, H.; Grolemund, G. R for Data Science. 1 st ed. Sebastopol: O´Reilly; 2016. 492 p.

Fox, J.; Weisberg, S. An R Companion to Applied Regression. 3 rd ed. California: SAGE; 2019. 576 p.

Kassambara, A. Practical Guide To Cluster Analysis in R. 1st ed: STHDA; 2017.

Machairidou, S. Big Data and Tableau [Master´s Thesis]. [Thessaloniki]: Aristotle University of Thessaloniki 2018.

Grafana Labs. About Grafana; [cited 2024.11.05]. Available from: https://grafana.com/docs/grafana/latest/introduction/

Apache ECharts. Features.; [cited 2024.11.05]. Available from: https://echarts.apache.org/en/feature.html

Volkov Labs. Business Charts; [cited 2024.11.05].

Available from: https://volkovlabs.io/plugins/business-input/panels/business-charts/

Stancin; Jovic, A. An overview and comparison of free Python libraries for data mining and big data analysis. 42nd International Convention on Information and communication technology, electronics and microelectronics 2019. p. 977-982.

Scikit-Learn. scikit-learn: Machine Learning in Python; [cited 2024.11.03 2024.11.03]. Available from: https://scikit-learn.org/stable/

SciPy. SciPy Documentation; [cited 2024.11.03]. Available from: https://docs.scipy.org/doc/scipy/

Statmodels. Statistical models, hypothesis tests, and data exploration; [cited 2024.11.03]. Available from: https://www.statsmodels.org/stable/index.html

Psycopg. Psycopg - PostgreSQL database adapter for Python; [modified 20212025.01.10]. Available from: https://www.psycopg.org/docs/

McCollam, R. Grafana variables: what they are and how they create dynamic dashboards; [cited 2025.01.20]. Available from: https://grafana.com/blog/2024/10/30/grafana-variables-what-they-are-and-how-they-create-dynamic-dashboards/

Conflicto de interés

Los autores autorizan la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Alejandro Rosete Suárez.
2. Curación de datos: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.
3. Análisis formal: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.
4. Investigación: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.
5. Metodología: Alejandro Rosete Suárez.
6. Administración del proyecto: Alejandro Rosete Suárez.
7. Recursos: Alejandro Rosete Suárez.
8. Software: Cristian Páez Olcha, Katherine Ramírez Hidalgo y Daniel Alejandro Deyne Rodríguez
9. Supervisión: Alejandro Rosete Suárez.
10. Validación: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.

11. Redacción – borrador original: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.
12. Redacción – revisión y edición: Cristian Páez Olcha, Katherine Ramírez Hidalgo, Daniel Alejandro Deyne Rodríguez y Alejandro Rosete Suárez.

Financiación

La investigación no requirió fuente de financiamiento.