Artículo Original

Una solución para visualizar modelos de árboles de decisión y reglas de asociación en Grafana

A solution for visualizing decision tree models and association rules in Grafana

Daniel Alejandro Deyne Rodríguez[1], Katherine Ramírez Hidalgo[1], Cristian Páez Olcha[1], Alejandro Rosete Suárez[2*]

[1]Universidad Tecnológica de La Habana José Antonio Echeverría Cujae, Marianao 19390, La Habana.
[2]Universidad Tecnológica de La Habana José Antonio Echeverría Cujae, Marianao 19390, La Habana. Avangenio S.R.L.

*Autor de correspondencia: alejandro.rosete.suarez@gmail.com, rosete@ceis.cujae.edu.cu

Resumen

Como elemento final de un proceso de Ingeniería de Datos, la visualización de la información de manera comprensible es un aspecto esencial. Entre las herramientas libres que pueden emplearse para la visualización se encuentra Grafana. A pesar de las numerosas ventajas que tiene esta herramienta no cuenta con gráficos que permitan visualizar árboles de decisión, ni reglas de asociación, dos de los modelos más populares para representar el conocimiento obtenido de un proceso de minería de datos. En este trabajo se presenta una solución a este problema que incluye como paso intermedio la representación de los modelos obtenidos en una base de datos relacional, así como la adaptación y extensión de componentes de Grafana para mostrar estos modelos. Se explican los pasos incluidos y se brinda un caso de estudio que permite evaluar las ventajas de la propuesta. En consecuencia, se obtienen visualizaciones de cada algoritmo con elementos dinámicos y personalizables, lo que posibilitó no solo la representación gráfica de los modelos generados sino también la interpretación de las relaciones entre los datos.

Palabras clave: Aprendizaje Automático, caso de estudio, Ingeniería de Datos, paneles, visualización.

Abstract

As a final element of a Data Engineering process, the comprehensible visualization of information is essential. Among the free tools that can be used for visualization is Grafana. Despite its numerous advantages, this tool does not offer graphics that allow the visualization of decision trees or association rules, two of the most popular models for representing the knowledge obtained from a data mining process. This paper presents a solution to this problem that includes, as an intermediate step, the representation of the obtained models in a relational database, as well as the adaptation and extension of Grafana components to display these models. The steps involved are explained, and a case study is provided to evaluate the advantages of the proposal. Consequently, visualizations of each algorithm are obtained with dynamic and customizable elements, which enabled not only the graphical representation of the generated models but also the interpretation of the relationships between the data.

Keywords: Machine Learning, case study, Data Engineering, dashboards, visualization.

1. Introducción

La ingeniería de datos y su visualización se han convertido en componentes fundamentales en el panorama actual, caracterizado por la generación masiva de datos. Estas disciplinas no solo permiten gestionar y procesar grandes volúmenes de información, sino también transformarla en representaciones gráficas comprensibles, facilitando así la toma de decisiones estratégicas. Su impacto, se puede observar en diversas áreas como por ejemplo en la del comercio electrónico, donde la integración con herramientas de inteligencia artificial permite anticipar las necesidades del usuario y optimizar de esta forma su experiencia. [1-9].

En este contexto, los modelos de aprendizaje automático, como los árboles de decisión y las reglas de asociación, desempeñan un papel crucial al descubrir patrones y relaciones significativas. Los árboles de decisión, son utilizados con el fin de clasificar y predecir resultados a partir de características específicas, representándose mediante diagramas jerárquicos que simplifican la interpretación de los criterios de decisión y los flujos lógicos [10, 11]. Este tipo de modelo ha sido aplicado en algunos estudios como, el análisis del dolor lumbar en estudiantes universitarios, mediante el cual se logró determinar factores relevantes asociados a esta afección [12]. Por otro lado, las reglas de asociación permiten identificar relaciones frecuentes entre elementos, lo que resulta especialmente útil en el análisis de transacciones y recomendaciones personalizadas [13, 14].

Existen diversas herramientas que permiten visualizar los modelos obtenidos con estas técnicas. Los árboles de decisión se suelen representar mediante diagramas jerárquicos utilizando funciones propias de la biblioteca scikit-learn [15], como plot_tree [16]. Además, scikit-learn permite exportar árboles en formato DOT, que puede ser procesado con herramientas como Graphviz [17] y Pydotplus [17] para generar visualizaciones personalizadas de alta calidad en diversos formatos, como PNG o PDF. Orange [18], una plataforma de análisis de datos visual, ofrece módulos interactivos para construir y visualizar árboles de decisión de manera sencilla. Estas herramientas destacan no solo nodos y métricas clave, como los umbrales de división y la distribución de las clases en las hojas, sino que también proporcionan interfaces intuitivas para explorar el comportamiento del modelo. Estudios recientes, como el análisis de cambios de momento en partidos de tenis [19] o la predicción de canciones exitosas mediante recursos audiovisuales [20] muestran las ventajas de utilizar la visualización de este tipo de modelo, contribuyendo así a interpretar mejor los factores determinantes en cada escenario.

Por otro lado, las reglas de asociación se visualizan, principalmente, mediante grafos que facilitan la comprensión de las relaciones entre diferentes elementos en un conjunto de datos transaccionales. En Python, la biblioteca networkx [21] se destaca por su capacidad para crear grafos personalizables que representan estas relaciones de manera visualmente atractiva. Además, permite evaluar métricas como soporte, confianza y lift directamente en los nodos y aristas del grafo, lo que facilita el análisis interactivo de las reglas generadas. Gephi [22], una plataforma de análisis de redes, también se utiliza para visualizar y explorar reglas de asociación, permitiendo crear representaciones gráficas, complejas y dinámicas, que destacan las relaciones entre los elementos. En el entorno de R, arulesViz [13] es una herramienta ampliamente utilizada que complementa este enfoque, ofreciendo diagramas de dispersión, matrices y grafos optimizados para el análisis de reglas de asociación. De manera similar, el enfoque que se propone en [23] introduce una matriz interactiva que permite explorar y comparar reglas de asociación, facilitando la identificación de patrones dentro de grandes volúmenes de datos.

Plataformas más completas, como Knime [24], destacan por ofrecer soluciones integradas que abarcan desde el procesamiento de datos hasta la visualización de modelos, incluyendo tanto árboles de decisión como reglas de asociación. Knime permite a los usuarios construir flujos de trabajo visuales de manera intuitiva, eliminando la necesidad de programar cada etapa del proceso. Su entorno incluye módulos preconfigurados que facilitan la generación de diagramas jerárquicos para árboles de decisión [25], así como grafos para las reglas de asociación [26].

Estas soluciones, sin embargo, suelen ser específicas y, en algunos casos, carecen de integración directa con herramientas libres avanzadas de visualización como Grafana. Esta herramienta, a través de su extensión "Business Chart" [27], integra las capacidades de la biblioteca Apache Echarts y ofrece un amplio conjunto de opciones para crear visualizaciones interactivas y personalizables. A pesar de que ECharts [28] permite generar visualizaciones para estructuras como árboles y grafos, estas no están diseñadas originalmente para representar de manera directa modelos de aprendizaje automático, lo cual limita su utilidad para este propósito [29]. En una reciente investigación [30], se logra la visualización de otros modelos de aprendizaje automático dentro de este entorno, no obstante, este esfuerzo no aborda la representación de algoritmos como árboles de decisión y reglas de asociación, aunque constituye un referente para la presente investigación.
Por tanto, cuando en un proyecto de Ingeniería de Datos que aproveche las ventajas de Grafana y ECharts se obtienen modelos de árboles de decisión y de reglas de asociación que pueden obtenerse de sus datos, no se cuenta con gráficos adecuados para poder visualizarlos. Este trabajo tiene como objetivo crear paneles en la herramienta Grafana para la visualización de modelos de aprendizaje automático que representen árboles de decisión y reglas de asociación.

2. Materiales y Métodos

Esta investigación se basó en el diseño de una propuesta para el almacenamiento y la visualización de modelos de árboles de decisión y reglas de asociación, basada en la integración de Python [31, 32] (como entorno para ejecutar algoritmos de aprendizaje automático) con un gestor de bases de datos relacionales como PostgreSQL [33], con el objetivo de facilitar su consulta y visualización en Grafana. Además, se realizó un caso de estudio a partir de comentarios emitidos por clientes de varios hoteles en relación con su experiencia.
En un primer momento se realizó un proceso de carga inicial de los datos, los cuales se utilizan para crear y entrenar los modelos a partir de los algoritmos correspondientes en cada caso (biblioteca scikit-learn para árboles de decisión y mlxtend para reglas de asociación), siendo almacenados posteriormente en PostgreSQL (ver interacción de las clases objetos involucrados en el procesamiento en el Diagrama UML de Secuencia mostrado en la **Fig.1**). El usuario inicia el proceso de creación y entrenamiento del modelo solicitándolo al *Controler*, el cual solicita a *DecisionTreeModel/AssociationRulesModel* que cree y entrene un modelo de árbol de decisión o de reglas de asociación. Este entonces entrena el modelo y devuelve la estructura del árbol o la regla de asociación al *Controler*. Posteriormente, *Controler* guarda la estructura del modelo entrenado en la base de datos. Finalmente, la base de datos confirma que la estructura del modelo ha sido almacenada y *Controler* envía notificación al usuario de que el modelo ha sido almacenado.

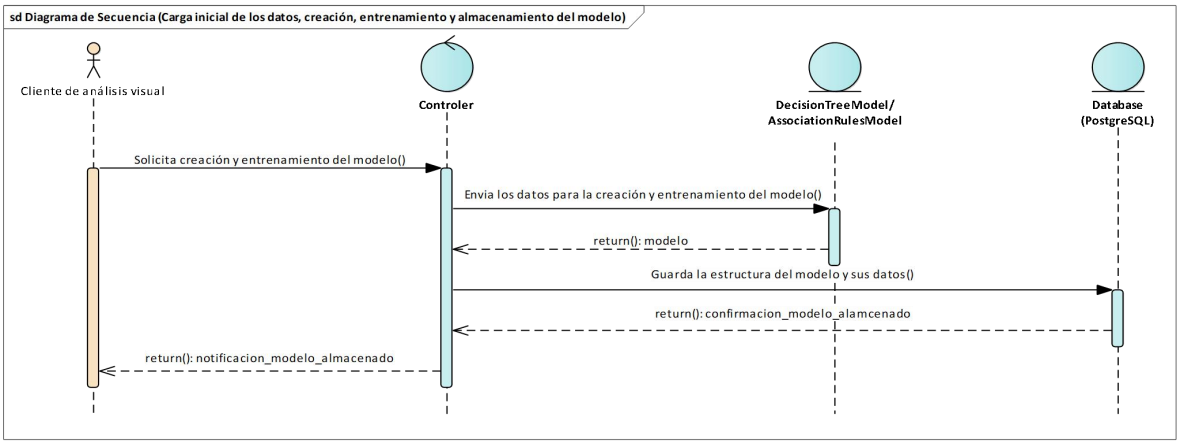

Fig.1 Diagrama de secuencia del proceso: Carga inicial de los datos, creación, entrenamiento y almacenamiento del modelo

Propuesta para el almacenamiento de los modelos

Para almacenar el modelo obtenido (árbol de decisión o reglas de asociación) se lleva a cabo lo siguiente:

1. Conectar el entorno Python de desarrollo con la base de datos, en este caso PostgreSQL, a través de la biblioteca *pg8000* (v1.31.2) [34, 35].
2. Crear una tabla en la base de datos para almacenar todos los modelos (ver **Fig.2**). Las filas serán cada uno de los modelos almacenados con su información correspondiente y las columnas son:
   - ✓ id: Es clave primaria y permite referenciar cada modelo de forma individual.
   - ✓ nombre: Permite guardar una sola vez el nombre del modelo y de esta manera se asegura la duplicidad de los modelos.
   - ✓ descripcion: Permite describir el conjunto de datos usado para la creación del modelo, así como el/los modelos almacenados.
   - ✓ creador: Permite identificar al creador del modelo, lo cual facilita el control con respecto a cada responsable de un modelo en específico.

	id [PK] integer	nombre character varying (255)	descripcion character varying (255)	creador character varying (255)
1	1	Iris	Se usa el dataset iris para modelo de árboles de decisión	Daniel
2	2	Transacciones de productos	Se usa el dataset de transacciones de productos para modelo de árboles de decisión y reglas de asociac...	Daniel
3	3	Coches	Se usa el dataset de coches para modelo de árboles de decisión	Daniel
4	4	Transacciones de prendas	Se usa el dataset de transacciones de prendas para modelo de reglas de asociación	Daniel

Fig.2 Tabla *grafana_ml_model_index* para almacenar los modelos

3. Crear una o varias tablas en la base de datos para almacenar un modelo en específico (árboles de decisión o reglas de asociación).

Árboles de decisión

1ra tabla (grafana_ml_model_arbol_decision)

- Las filas serán cada uno de los nodos del árbol con sus atributos correspondientes.
- Las columnas son:
  - ✓ modelo_id: Permite que cada nodo tenga la referencia del modelo al que pertenece.
  - ✓ nodo_id: Es clave primaria y permite referenciar cada nodo de forma individual.
  - ✓ nodo_padre: Permite referenciar el padre en cada uno de los nodos. En el caso de que el nodo sea raíz este campo será NULL.
  - ✓ característica: Indica qué característica se utiliza para realizar la división en este nodo. Si el nodo es una hoja (no tiene hijos y predice un valor), este campo puede ser NULL porque no hay más divisiones en las hojas.
  - ✓ umbral: Definir el valor que se compara con la característica para determinar la dirección de la rama que debe seguirse en el árbol (nodo izquierdo o derecho).
  - ✓ nodo_izquierdo y nodo_derecho: Estas columnas permiten seguir el camino en el árbol hacia el siguiente nodo. Si el nodo es una hoja (nodo terminal), ambos valores pueden ser NULL porque no hay más nodos hijos que seguir.
  - ✓ es_hoja: Si es TRUE, significa que el nodo es una hoja (nodo terminal); si es FALSE, el nodo sigue dividiendo los datos en más ramas.
  - ✓ valor predicción: Se usa cuando se llega a una hoja durante el recorrido del árbol. Este valor se devuelve como la predicción final.

2da tabla (características)

- Las filas serán cada uno de las características del conjunto de datos.
- Las columnas son:
  - ✓ modelo_id: Permite que cada característica tenga la referencia del modelo al que pertenece.
  - ✓ caracteristica_id: Es clave primaria y permite referenciar cada característica de forma individual.
  - ✓ nombre: Permite guardar una sola vez el nombre de la característica y de esta manera se evita la redundancia de los datos.

3ra tabla (valores_prediccion)

- Las filas serán cada una de las clases o valores de predicción del conjunto de datos.
- Las columnas son:
  - ✓ modelo_id: Permite que cada clase tenga la referencia del modelo al que pertenece.
  - ✓ prediccion_id: Es clave primaria y permite referenciar cada una de las clases de forma individual.
  - ✓ nombre_clase: Permite guardar una sola vez el nombre de la clase y de esta manera se evita la redundancia de los datos.

Reglas de asociación

En el caso de las Reglas de Asociación, se optó por una sola tabla. A continuación, se describe la misma:

- Las filas serán cada una de las reglas generadas con sus respectivos atributos.
- Las columnas son:
  - ✓ modelo_id: Permite que cada regla tenga la referencia del modelo al que pertenece.
  - ✓ id: Sirve como clave primaria para identificar cada regla de forma individual.
  - ✓ antecedente: Definir la condición inicial de la regla, es decir, los ítems cuya presencia se analiza para determinar su relación con el consecuente.
  - ✓ consecuente: Representa el resultado probable de la regla, indicando los elementos que tienden a ocurrir junto con el antecedente.
  - ✓ soporte: Calcula la proporción de transacciones en las que el antecedente y el consecuente ocurren juntos.
  - ✓ confianza: Evalúa la certeza de la regla calculando la proporción de transacciones con el antecedente y el consecuente, en relación con las transacciones que contienen solo el antecedente.
  - ✓ lift: Calcula la fuerza de la asociación comparando el soporte conjunto de A y B con la confianza de B.

Propuesta para la visualización de los modelos

Una vez realizado los procesos de crear, entrenar y almacenar los modelos, se pasa a la visualización de los mismos. Para ello se tuvieron en cuenta los siguientes pasos:

1. Ejecutar una consulta que permita obtener los datos del modelo que se desea visualizar.
2. Modificar el código de la visualización seleccionada del sitio de Echarts, con el fin de ajustar la entrada de datos y la estructura del modelo que se desea mostrar.
3. Finalmente, se muestra la visualización del modelo.

Árboles de decisión

En el caso de los árboles de decisión, se parte de insertar en el panel de consulta de Grafana el código que recupera el modelo a visualizar, almacenado en la estructura previamente explicada. Luego, en el panel de programación de Grafana se toma el código base de ejemplos de árboles de

Echart y se modifica para conectar la entrada con la consulta que obtiene los modelos con el fin de que pueda cambiar según lo que se haya obtenido del algoritmo que induce los modelos. Otro ajuste es con respecto a la forma de visualizar el modelo, donde se tiene en cuenta, por ejemplo, la comparación de la característica con el umbral, lo cual representa una decisión clave en cada nodo del árbol y define cómo se separan los datos. Dichas etiquetas son fundamentales para entender de forma clara y sencilla el flujo lógico del modelo, pues indican el camino que sigue cada decisión, simplificando la interpretación de cómo las características y los valores conducen a una predicción final. Todo el código que respalda esta propuesta se brinda de manera libre en Git Hub para permitir la experimentación y reproducción de los resultados, así como su uso por la comunidad.

Reglas de asociación

En el caso de las reglas de asociación, opera de manera similar. Se ajusta la consulta para recuperar las reglas a mostrar, usando el panel de consulta de Grafana y se ajusta el panel de código, refinando de una manera diferente el gráfico que usa Echart para mostrar grafos. En este caso se tienen en cuenta las etiquetas para mostrar la confianza, el soporte, el lift, las cuales permiten interpretar la calidad y relevancia de las reglas generadas. Además, la dirección de las relaciones mediante flechas es otro aspecto importante a tener en cuenta a la hora de visualizar este tipo de modelo, pues de esta forma el usuario logra entender cuál es el antecedente y el consecuente, así como también las relaciones de conjunción, que surgen cuando un elemento compuesto, como por ejemplo "**pan, leche**," mantiene conexiones individuales con los elementos que lo conforman, como "**pan**" y "**leche**". Se usan colores en los nodos para diferenciar las partes de las reglas (antecedente o consecuente).

Opciones de personalización para la visualización de los modelos

Árboles de decisión

En el caso de los árboles de decisión, incluyen variables que permiten adaptar las visualizaciones según desee el usuario:

- Horizontal, donde las ramas del árbol se despliegan de izquierda a derecha, ofreciendo una perspectiva clara de las jerarquías y decisiones en un formato lineal;
- Vertical, que organiza los nodos de arriba hacia abajo, facilitando la interpretación desde la raíz hasta las hojas como un flujo descendente;
- Circular, que dispone los nodos en un diseño radial a partir del nodo raíz en el centro; y
- Doble, que permite comparar dos estructuras arbóreas en un solo esquema visual.

También es ajustable el color de fondo (modo Claro u Oscuro), el grosor de los nodos, el tamaño de las letras, las dimensiones de los nodos (para poder hacer los gráficos más o menos compactos), etc. Todo esto es muy importante para que puedan ajustarse los paneles creados al entorno visual del proyecto de Ingeniería de Datos en que se quieran mostrar los modelos. En la siguiente sección se muestran ejemplos que ilustran la flexibilidad de la propuesta.

Reglas de asociación

En cuanto a las reglas de asociación, también hay varias opciones para configurar la visualización. En cuanto al diagrama en sí, se incluyen 4 variantes:

- Grafo, donde se utilizan nodos y aristas para mostrar las relaciones entre los elementos, representando los antecedentes y consecuentes como nodos conectados por flechas que indican la dirección de la regla, así como también nodos conectados por líneas discontinuas para representar relaciones de conjunción;

- Matricial, donde el eje "x" (abscisas) corresponde a los antecedentes y el eje "y" (ordenadas) a los consecuentes. Los nodos dentro de la matriz simbolizan reglas específicas e incluyen métricas clave, como la confianza y el soporte;
- Separadas, que distribuye las reglas en grupos individuales, organizando los antecedentes y consecuentes de manera independiente;
- 3D (ver **Figura.10**), que permite introducir una dimensión adicional como es el Lift, además de los antecedentes (eje x) y los consecuentes (eje y), y al igual que en la matricial, los nodos representan reglas específicas e incluyen métricas como la confianza y el soporte.

En la siguiente sección se muestran ejemplos que ilustran estas variantes y otros aspectos de la flexibilidad de la propuesta como el modo Claro u Oscuro y la paleta de colores. También se incluye la posibilidad de mostrar u ocultar las relaciones de conjunción (que puede haber entre varios conceptos en el antecedente) o implicación (entre antecedente y consecuente) que conforman las reglas de asociación para facilitar el análisis en detalle de las mismas en la variante de Grafo.

Para garantizar tanto la reproducibilidad de los resultados que aquí se presentan, como para facilitar su uso, todo el código de la solución se puede encontrar en el siguiente sitio de **GitHub**: https://github.com/DanielDeyne/Codigos-Arboles-de-decision-y-Reglas-de-asociacion.git

3. Resultados y Discusión

Para mostrar las ventajas de la propuesta se desarrolló el siguiente caso de estudio que se apoya en un conjunto de datos que corresponde a comentarios emitidos por clientes en relación con su experiencia en hoteles. Este cuenta con 29 columnas y un total de 43 443 filas, las cuales representan cada comentario. Luego de realizar un preprocesamiento, donde no se tuvo en cuenta las filas con valores faltantes, quedaron finalmente un total de 41 492 filas.
Las columnas se dividen en varias categorías:

- Palabras clave en comentarios (Binarias): Estas indican la presencia (**1**) o ausencia (**0**) de palabras específicas en los comentarios. Cada palabra representa un aspecto relevante de la experiencia del cliente.
  Columnas: **hotel**, **resort**, **staff**, **beach**, **cuba**, …..., **buffet**, **excelente**.
- Valoraciones del comentario: Describen la evaluación del comentario en base a percepciones humanas o modelos automáticos.
  Columnas:
  - ✓ **val_h**: Valoración realizada por humanos sobre el comentario (1: Positiva, **0**: Neutra, -1: Negativa).
  - ✓ **val_llm**: Valoración realizada por un modelo de lenguaje (LLM) (1: Positiva, **0**: Neutra, -1: Negativa).
- Atributos descriptivos adicionales: Describen información adicional sobre el contexto del comentario y el perfil del hotel.
  Columnas: **l_comentario**, **polo**, **modality**, **segment**, **dia_de_semana**, **mes**, **dia_del_mes**.

Árboles de decisión

En el caso de los árboles de decisión se crearon dos modelos (ver **Fig.3**). Para el 1er modelo, se consideraron las columnas relacionadas con las palabras claves en comentarios, así como el contenido textual del comentario (l_comentario). La variable objetivo seleccionada fue "val_h", la cual representa la valoración realizada por humanos (1: Positiva, 0: Neutra, -1: Negativa). En el 2do modelo, se consideraron las mismas columnas utilizadas en el modelo anterior, cambiando solamente la variable objetivo, que para este caso fue "val_llm**"**, la cual representa la valoración realizada por un modelo de lenguaje (1: Positiva, 0: Neutra, -1: Negativa). En ambos casos se

ajustó la profundidad máxima a 3 niveles y se aplicó un peso a las clases para mejorar el balance en la distribución de las clases de respuestas. En la figura se empleó la variante Doble para mostrar 2 árboles lo cual facilita la comparación de los modelos que en este caso reflejan los aspectos detectados como atributos más importantes para la clasificación hecha por un humano y por un LLM.

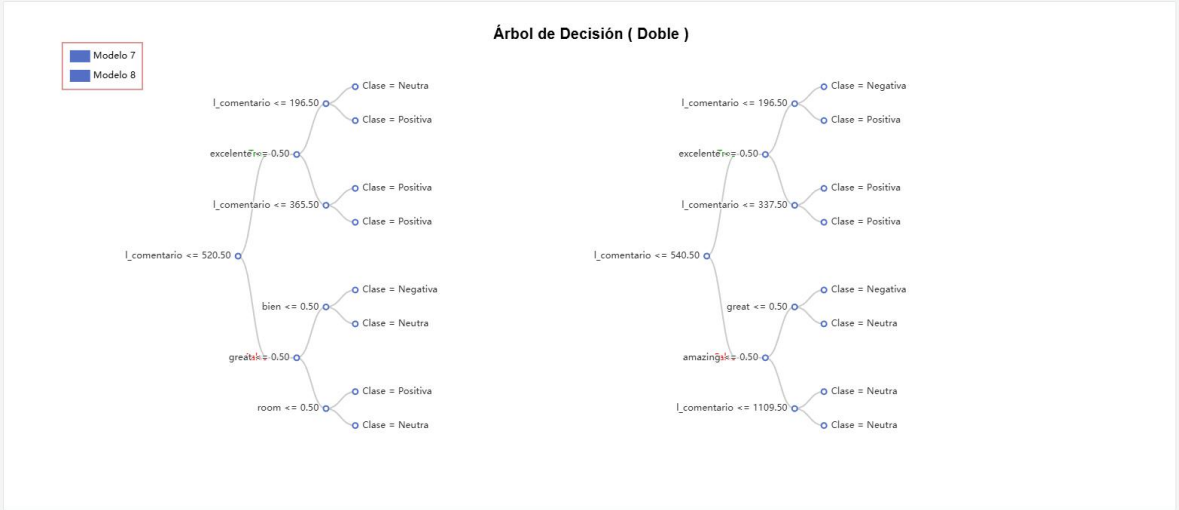

Fig.3 Visualización de modelo de árbol de decisión en forma doble modo claro: Modelo 7 - dataset de comentarios emitidos: “val_h” como variable objetivo, Modelo 8 - dataset de comentarios emitidos: “val_llm” como variable objetivo

Ventajas y flexibilidad de este modelo:

- Identificación de factores determinantes: Permite identificar palabras claves y características del contenido textual (como l_comentario) que tienen mayor impacto en las valoraciones, ya sean humanas (val_h) o automáticas (val_llm).
- Comprensión y visualización de patrones: Ofrece una representación clara y estructurada de cómo las características seleccionadas influyen en la percepción de los comentarios, mostrando de manera intuitiva cómo se clasifican en positivos, neutros o negativos.
- Análisis comparativo del modelo: Facilita la comparación entre valoraciones humanas y automáticas, ayudando a comprender cómo cada modelo procesa la información y qué características tienen mayor peso en las decisiones finales.
- Apoyo en la gestión de opiniones y toma de decisiones: Proporciona información práctica para identificar palabras claves críticas en las valoraciones (positivas o negativas), apoyando decisiones que mejoren la experiencia del cliente.

En las **Fig.4**, **Fig.5**, **Fig.6** se muestran ejemplos de la visualización de estos mismos árboles empleando las otras variantes (Horizontal, Vertical, Circular). La **Fig.4** ilustra la disposición horizontal, donde las ramas del árbol se despliegan de izquierda a derecha, facilitando la lectura secuencial de las decisiones. En la **Fig.5**, se presenta la variante vertical, con los nodos organizados jerárquicamente desde la raíz hasta las hojas, permitiendo una visualización estructurada del modelo. Finalmente, la **Fig.6** muestra la representación circular, en la que los nodos se organizan radialmente en torno al nodo raíz, destacando las relaciones entre los elementos. Se incluyen también variaciones de otros aspectos (Modo, Tamaño de etiqueta, Grosor arista) para ilustrar la flexibilidad de la propuesta.

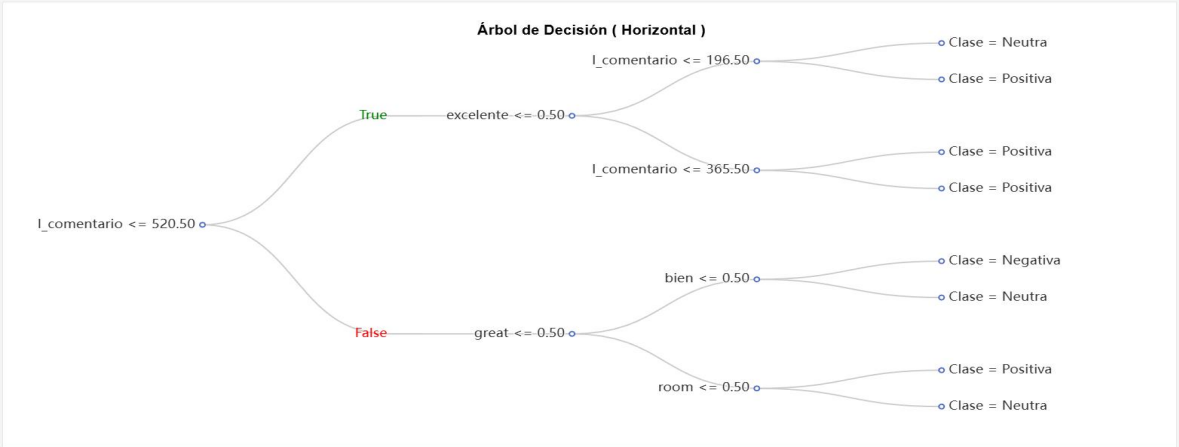

Fig.4 Visualización de modelo de árbol de decisión en forma horizontal, modo claro, etiquetas grandes y aristas finas: Modelo 7 - dataset de comentarios emitidos: “val_h” como variable objetivo

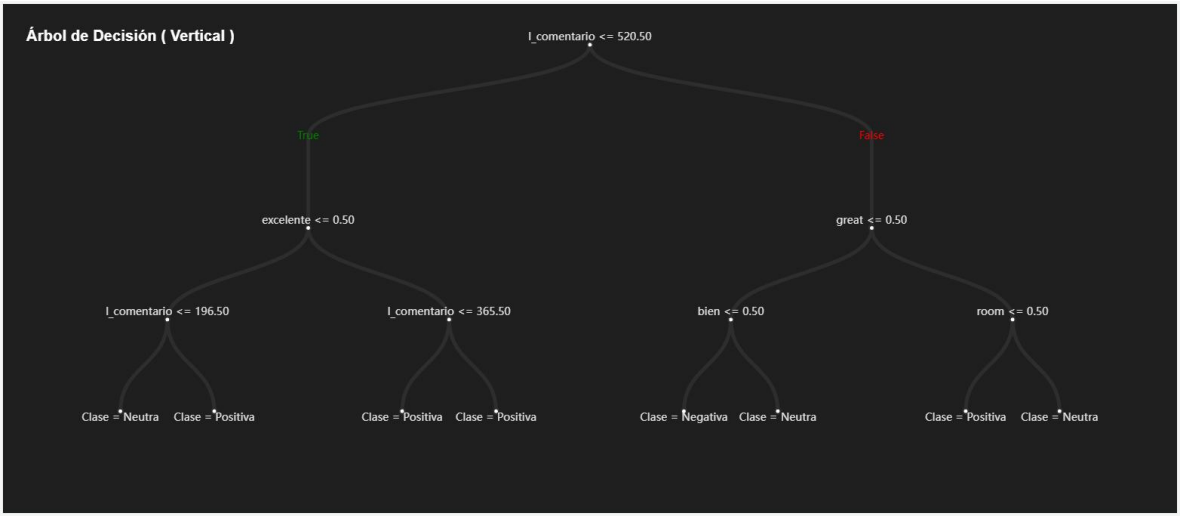

Fig.5 Visualización de modelo de árbol de decisión en forma vertical, modo oscuro, etiquetas medianas y aristas gruesas: Modelo 7 - dataset de comentarios emitidos: “val_h” como variable objetivo

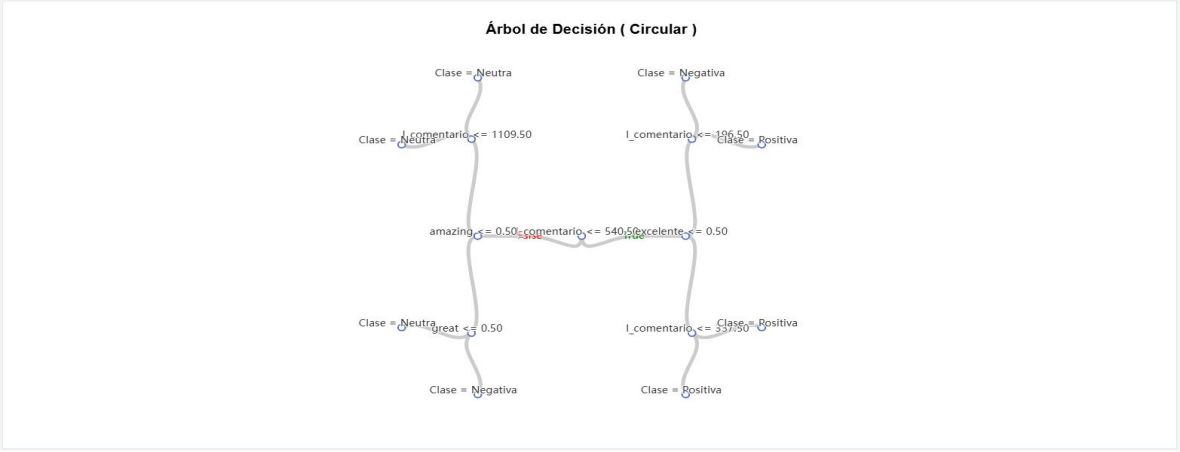

Fig.6 Visualización de modelo de árbol de decisión en forma circular, modo claro, etiquetas medianas y aristas gruesas: Modelo 8 - dataset de comentarios emitidos: “val_llm” como variable objetivo

Reglas de asociación

En el caso de las reglas de asociación se crearon dos modelos para ilustrar la flexibilidad de la propuesta. Para el 1er modelo (ver **Fig.7**), se seleccionaron las columnas correspondientes a las palabras clave en comentarios, además de las variables de valoración val_h (humana) y val_llm (modelo de lenguaje). La inclusión de estas variables adicionales tuvo como propósito analizar no solo las co-ocurrencias entre las palabras claves, sino también descubrir patrones frecuentes asociados a las valoraciones, tanto humanas como del modelo. Con el fin de garantizar una representación clara y comprensible, se limitó la consulta a las 20 primeras reglas, mediante un LIMIT en la configuración del panel, con el fin de evitar la sobrecarga y facilitar la interpretación de las mismas.

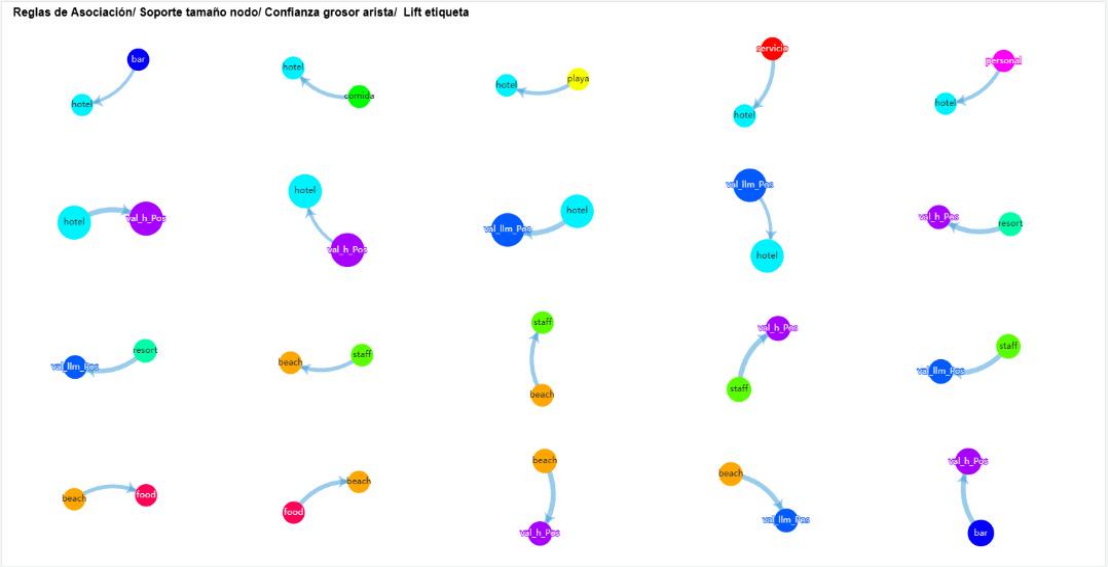

Fig.7 Visualización de modelo de reglas de asociación de forma separadas modo claro con dataset de comentarios emitidos (con “val_h” y “val_llm”)

Ventajas y flexibilidad de este modelo:

- Descubrir patrones frecuentes: Identificar combinaciones de palabras claves que aparecen de manera recurrente en los comentarios y analizar cómo estas relaciones se reflejan en las valoraciones humanas (val_h) y automáticas (val_llm), y viceversa.
- Comparar perspectivas de valoración**:** Evaluar si las asociaciones entre palabras claves son consistentes al comparar las valoraciones humanas y automáticas, detectando posibles diferencias o similitudes en la interpretación de los comentarios.
- Revelar tendencias implícitas**:** Descubrir relaciones complejas entre las palabras claves y cómo estas co-ocurrencias pueden destacar preferencias o aspectos específicos percibidos en los comentarios, incluso cuando las valoraciones varían.

Para la creación del 2do modelo (ver **Fig.8**), se consideraron las mismas columnas utilizadas en el modelo anterior, excluyendo las variables de valoración val_h y val_llm. El objetivo principal de este modelo es descubrir relaciones frecuentes entre las palabras clave sin la influencia directa de las valoraciones. En este caso no fue necesario aplicar un LIMIT en la configuración del panel,

como en el caso del 1er modelo debido a que no se generaron tantas reglas y por ende no afecta la visualización.

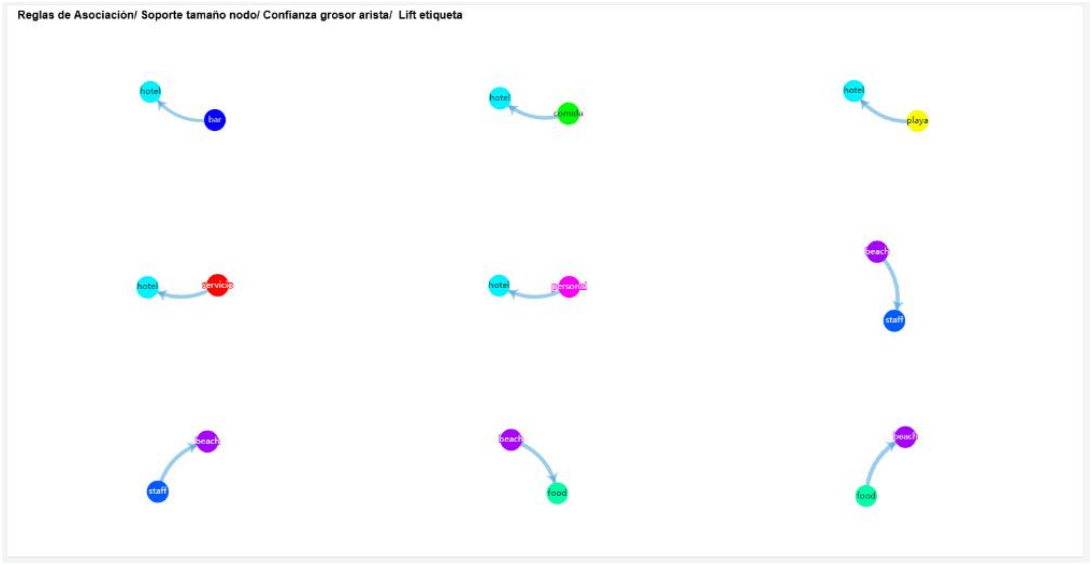

Fig.8 Visualización de modelo de reglas de asociación de forma separadas modo claro con dataset de comentarios emitidos (sin "val_h" y "val_llm")

Ventajas y flexibilidad de este modelo:

- Descubrir patrones frecuentes**:** Identificar combinaciones de palabras claves que se mencionan de manera recurrente en los comentarios, lo cual permite detectar tendencias en los servicios o aspectos más mencionados por los clientes.
- Revelar relaciones entre servicios: Analizar cómo ciertos términos claves tienden a aparecer juntos en las encuestas, destacando posibles vínculos implícitos entre los ítems evaluados.
- Analizar contenido sin sesgos: Explorar las relaciones directas entre los términos claves del contenido textual, sin la influencia de las valoraciones humanas o automáticas, permitiendo una visión más neutral de las preferencias o menciones recurrentes.

En la **Fig.9** y **Fig.10** se muestra el mismo conjunto de reglas de la **Fig.7**, pero utilizando la variante matricial y la variante 3D. La **Fig.9** presenta la visualización matricial, donde los antecedentes y consecuentes se organizan en un sistema de coordenadas, permitiendo identificar patrones de asociación de manera estructurada. Por otro lado, la **Fig.10** emplea la variante 3D, incorporando una tercera dimensión que permite visualizar el lift junto con el soporte y la confianza, lo que facilita un análisis más detallado de la relevancia de las reglas generadas. Se puede ver cómo estas representaciones ofrecen una visión alternativa del modelo, destacando diferentes aspectos clave en la interpretación de las reglas de asociación.
Como puede verse en los ejemplos, la propuesta puede adaptarse a diferentes intereses de visualización, permitiendo integrar visualizaciones de árboles de decisión y reglas de asociación a paneles de Grafana usados en un proyecto de Ingeniería de Datos. Al disponerse de manera libre estos códigos, puede permitir un mayor uso de resultados de la aplicación de algoritmos de aprendizaje automático en contexto de toma de decisiones.

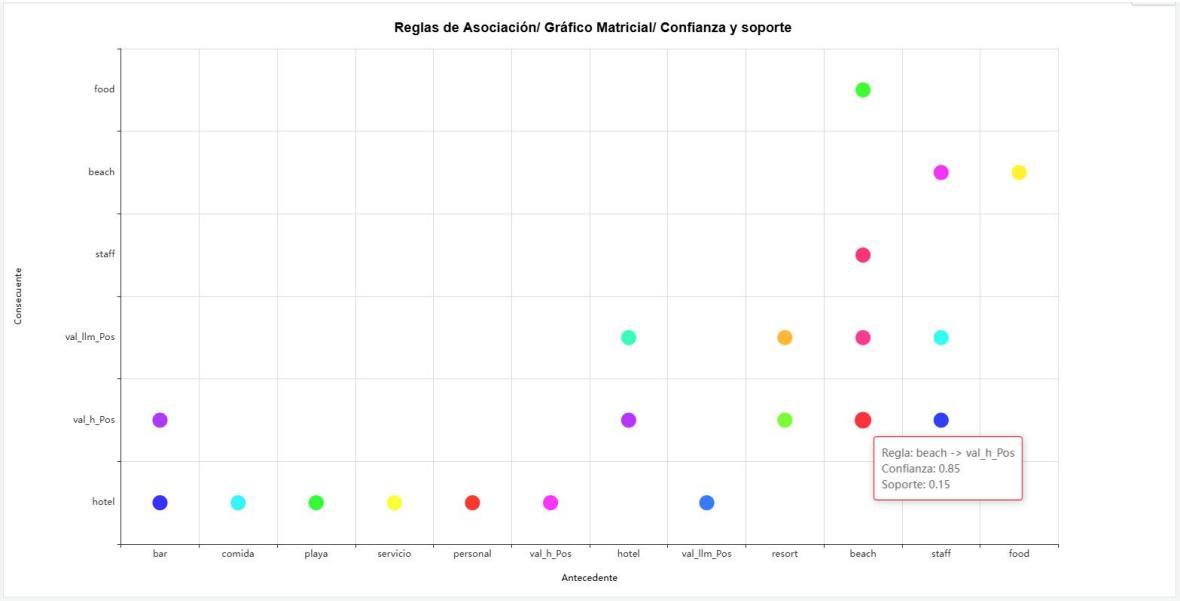

Fig.9 Visualización de modelo de reglas de asociación de forma matricial modo claro con dataset de comentarios emitidos (con "val_h" y "val_llm")

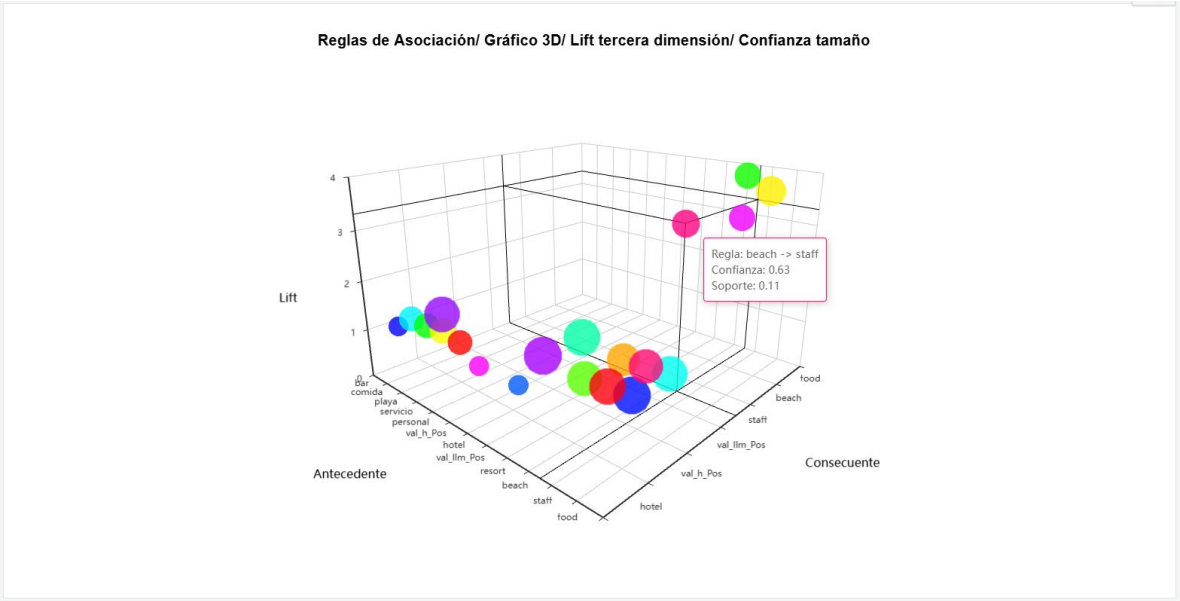

Fig.10 Visualización de modelo de reglas de asociación de forma 3D modo claro con dataset de comentarios emitidos (con "val_h" y "val_llm")

4. Conclusiones

En esta investigación se diseña una solución para el almacenamiento y la persistencia de los modelos en PostgreSQL, desarrollando una estructura que permite almacenar tanto árboles de decisión como reglas de asociación. Esta solución facilita la recuperación de los modelos almacenados mediante consultas SQL, integrándose con Grafana para su posterior visualización. Los paneles creados en Grafana extendieron los códigos de Echarts, lo que facilita tanto el procesamiento de datos desde consultas SQL como la representación de los modelos. Estos ajustes posibilitan la personalización de elementos claves de las visualizaciones, como el tamaño de las etiquetas, el grosor de las aristas, las paletas de colores, entre otros, garantizando flexibilidad y

adaptabilidad a las necesidades específicas del usuario. El caso de estudio realizado permite validar la funcionalidad de los paneles creados en Grafana. Este demuestra que, dichos paneles son capaces de representar correctamente los modelos de aprendizaje automático, de un modo integrable a un proyecto de Ingeniería de Datos usando Grafana. Además, los casos de estudios muestran cómo estos podrían ayudar a identificar patrones claves, resaltar relaciones significativas entre los datos y ofrecer una mayor claridad en la visualización.

Referencias

1. Reis, J., Housley, M., *Fundamentals of Data Engineering: Plan and Build Robust Data Systems*, 2022. 1ra ed. Boston: O'Reilly Media. ISBN: 978-1-098-10830-4
2. Gorodov, E.Y., Gubarev, V.V., *Analytical Review of Data Visualization Methods in Application to Big Data.* Journal of Electrical and Computer Engineering, 2013. vol. 2013: p. 1–7. DOI: http://dx.doi.org/10.1155/2013/969458
3. Han, J., Kamber, M., Pei, J., *Data Mining: Concepts and Techniques*, 2012. 3ra ed. Amsterdam: Morgan Kaufmann. ISBN: 978-0-12-381479-1
4. Ali, S.M., Gupta, N., Nayak, G.K., Lenka, R.K., *Big data visualization: Tools and challenges*. 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016. Greater Noida, India: p. 656–660.
5. Cuadrado-Gallego, J.J., Demchenko, Y., Losada, M.A., Ormandjieva, O., *Classification and Analysis of Techniques and Tools for Data Visualization Teaching*. 2021 IEEE Global Engineering Education Conference (EDUCON), 2021. Vienna, Austria: p. 1593–1599.
6. Taylor L., Gupta V., Jung K., *Leveraging Visualization and Machine Learning Techniques in Education: A Case Study of K-12 State Assessment Data*. Multimodal Technol Interact, 2024. 8(4):28. DOI: https://doi.org/10.3390/mti8040028
7. Fatih Ak M., *A Comparative Analysis of Breast Cancer Detection and Diagnosis Using Data Visualization and Machine Learning Applications*. Healthcare, 2020. 8(2):111. DOI: https://doi.org/10.3390/healthcare8020111
8. Liang Z., Liang Z., Zheng Y., Liang B., Zheng L., *Data Analysis and Visualization Platform Design for Batteries Using Flask-Based Python Web Service*. World Electr Veh J, 2021. 12(4):187. DOI: https://doi.org/10.3390/wevj12040187
9. Necula S.C., Strîmbei C., *Top 10 Differences between Machine Learning Engineers and Data Scientists*. Electronics, 2022. 11(19):3016. DOI: https://doi.org/10.3390/electronics11193016
10. Skiena, S.S., *The Data Science Design Manual*, 2017. 1ra ed. New York: Stony Brook University. ISBN: 978-3-319-55443-3
11. Quinlan, J.R., *Induction of Decision Trees*, 2007. Sydney: New South Wales Institute of Technology.
12. Abbas J., Yousef M., Hamoud K., Joubran K., *Low Back Pain Among Health Sciences Undergraduates: Results Obtained from a Machine-Learning Analysis*. J Clin Med, 2023. 14(6):2046. DOI: https://doi.org/10.3390/jcm14062046
13. Hahsler, M., *arulesViz: Interactive Visualization of Association Rules with R*. The R Journal, 2017. 9: p. 163–174.
14. Hahsler, M., Chelluboina, S., *Visualizing Association Rules in Hierarchical Groups*. Proceedings of the 42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface 2011), 2011. p. 1–11.
15. Pedregosa, F., Varoquaux, G., Michel, V., et al., *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 2011. 12: p. 2825–2830.
16. Scikit-learn, *plot_tree — scikit-learn 1.5.2 documentation*. [Consulta: 27 octubre 2024]. Disponible en: https://scikit-learn.org/1.5/modules/generated/sklearn.tree.plot_tree.html

17. Kattani, A., *Visualizing A Decision Tree Using GraphViz and Pydotplus*, 2022. [Consulta: 27 octubre 2024]. Disponible en: https://anantha-kattani.medium.com/visualizing-a-decision-tree-using-graphviz-and-pydotplus-24a046faac0b
18. Orange Data Mining, *Tree Viewer*. [Consulta: 27 octubre 2024]. Disponible en: https://orangedatamining.com/widget-catalog/visualize/treeviewer/
19. Xia Y., Li C., Zhang T., *Analyzing Momentum Shifts in Tennis: A Machine-Learning Approach to Predicting Match Outcomes*. Appl Sci, 2025. 15(4):2018. DOI: https://doi.org/10.3390/app15042018
20. Lee C.Y., Tu Y.N., *Predicting Hit Songs Using Audio and Visual Features*. Eng Proc, 2025. 8(1):9043. DOI: https://doi.org/10.3390/engproc2025089043
21. NetworkX Developers, *Tutorial: documentación de NetworkX 3.4.2*. [Consulta: 27 octubre 2024]. Disponible en: https://networkx.org/documentation/stable/tutorial.html
22. Pylak, K., Majerek, D., *Impact of the Service Sector on the Creation of Companies in Poland*. Procedia Economics and Finance, 2015. 24: p. 523–532.
23. Varu R., Christino L., Paulovich F.V., *ARMatrix: An Interactive Item-to-Rule Matrix for Association Rules Visual Analytics*. Electronics, 2022. 11(9):1344. DOI: https://doi.org/10.3390/electronics11091344
24. Knime, *How to Create Workflow & REST Services with KNIME*, 2023. [Consulta: 30 octubre 2024]. Disponible en: https://www.knime.com/blog/call-workflow-service-REST
25. Knime Community Hub, *Decision Tree to Image*. [Consulta: 27 octubre 2024]. Disponible en: https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.decisiontree2.image.DecTreeToImageNodeFactory
26. Knime Community Hub, *Rule Viewer (local) (Legacy)*. [Consulta: 27 octubre 2024]. Disponible en: https://hub.knime.com/knime/extensions/org.knime.features.ext.md/latest/org.knime.ext.md.RuleViewerNodeFactory
27. Volkov Labs, *Business Charts*. [Consulta: 27 octubre 2024]. Disponible en: https://volkovlabs.io/plugins/business-charts/
28. ECharts, *ECharts Examples*. [Consulta: 1 noviembre 2024]. Disponible en: https://echarts.apache.org/examples/en/index.html
29. Neptune.ai, *How to Do Model Visualization in Machine Learning?*, 2024. [Consulta: 26 octubre 2024]. Disponible en: https://neptune.ai/blog/visualization-in-machine-learning
30. P. Olcha K., Ramírez K, Deyne D.A., Rosete A., *Una solución para visualizar modelos de agrupamiento, correlación y regresión en Grafana*. Serie Científica de la Universidad de las Ciencias Informáticas, 2025. vol. 18: p. 28–48.
31. InfluxData, *MLOps: A Comprehensive Guide to Machine Learning Operations*. [Consulta: 27 octubre 2024]. Disponible en: https://www.influxdata.com/glossary/mlops/
32. Bytescrum, *How to Deploy Machine Learning Models in Production: Key Challenges and Fixes*, 2024. [Consulta: 27 octubre 2024]. Disponible en: https://blog.bytescrum.com/how-to-deploy-machine-learning-models-in-production
33. Viloria, A., Camargo Acuña, G., Alcázar Franco, D.J., et al., *Integration of Data Mining Techniques to PostgreSQL Database Manager System*. Procedia Computer Science, 2019. 155: p. 575–580.
34. PyPI, *pg8000 1.31.2*, 2024. [Consulta: 26 octubre 2024]. Disponible en: https://pypi.org/project/pg8000/1.31.2/
35. Ahmed, J., *Top PostgreSQL Drivers for Python*. [Consulta: 27 octubre 2024]. Disponible en: https://www.timescale.com/learn/top-postgresql-drivers-for-python

Conflicto de Intereses

Los autores declaran no tener conflicto de intereses.

Contribución de los autores

Daniel Alejandro Deyne Rodríguez. ORCID: https://orcid.org/0009-0001-6508-9603
Participó en la conceptualización, curación de datos, investigación, metodología, administración del proyecto, recursos y redacción.

Katherine Ramírez Hidalgo. ORCID: https://orcid.org/0009-0008-1266-0918
Participó en la conceptualización, investigación, metodología, validación y edición.

Cristian Páez Olcha. ORCID: https://orcid.org/0009-0001-8137-261X
Participó en la conceptualización, investigación, metodología, validación y edición.

Alejandro Rosete Suárez. ORCID: https://orcid.org/0000-0002-4579-3556
Participó en la conceptualización, curación de datos, investigación, metodología, administración del proyecto, recursos, supervisión, validación, redacción y edición.