UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Básicas e Ingenierías

Licenciatura en Ciencias Computacionales

Aplicación web para la ejecución de algoritmos descriptivos de Minería de Datos para usuarios no expertos

T E S I S

PARA OBTENER EL GRADO DE:
LICENCIADO EN CIENCIAS COMPUTACIONALES

Presenta:

Guzman Vera Oscar Adair

Asesora: Dra. Anilú Franco Arcega
CoAsesor: Dr. Luis Heriberto García Islas

Mineral de la Reforma, Hgo. 2024

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 10 de junio de 2024

Número de control: ICBI-D/764/2024
Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio le comunico que el Jurado asignado al Egresado de la Licenciatura en Ciencias Computacionales **Oscar Adair Guzmán Vera,** quien presenta el trabajo de titulación **"Aplicación web para la ejecución de algoritmos descriptivos de minería de datos para usuarios no expertos",** después de revisar el trabajo en reunión de Sinodales ha decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Dr. Virgilio López Morales

Secretario: Dr. Manuel Alejandro Misses Ojeda

Vocal: Dra. Anilú Franco Arcega

Suplente: Dr. Luis Heriberto García Islas

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
"Amor, Orden y Progreso"

Dr. Otilio Arturo Acevedo Sandoval
Director del ICBI

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO ICBI DIRECCIÓN

OAAS/YCC

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 Ext. 40001
direccion_icbi@uaeh.edu.mx,
acevedo@uaeh.edu.mx

Agradecimientos

Agradezco a mis padres por su apoyo constante durante mi investigación y redacción de esta tesis. Sus palabras de ánimo y comprensión han sido fundamentales en este proceso. Gracias por estar siempre ahí para mí.

Le agradezco a mis queridos hermanos por su constante apoyo y comprensión durante la realización de esta tesis. Su presencia ha sido una fuente de motivación en cada etapa de este viaje académico.

Quiero expresar mi más profundo agradecimiento a mi asesora de tesis, Anilú Franco Arceaga, por su invaluable orientación, apoyo y dedicación a lo largo de este proyecto. Su experiencia, conocimientos y perspicacia han sido fundamentales en cada etapa de este proceso de investigación.

Finalmente agradezco a mis amigos, quienes han sido una parte fundamental de mi viaje durante toda la universidad. Su amistad ha sido un pilar de fortaleza en los momentos más difíciles y una fuente constante de alegría.

Resumen

La Minería de Datos es el proceso de descubrir patrones y relaciones significativas en grandes conjuntos de datos, utilizando técnicas de análisis estadístico y algoritmos de aprendizaje automático. Esta disciplina se utiliza para extraer información útil de grandes conjuntos de datos. Además, la Minería de Datos puede ayudar a identificar tendencias ocultas en los datos, lo que puede mejorar la toma de decisiones en áreas como la predicción de tendencias del mercado, la detección de fraudes y la evaluación del rendimiento de los empleados.

La evolución de la Minería va desde el análisis manual de los datos, donde se ocupaba un especialista experimentado para tomar las decisiones, luego los sistemas operacionales, posteriormente los sistemas de consultas de bases de datos que permiten a los usuarios realizar consultas en una base de datos para recuperar información, y por último los sistemas informativos que son sistemas de software diseñados para integrar y gestionar datos de diferentes fuentes, permitiendo la generación de informes y análisis de datos a través de disciplinas como la Minería de Datos.

La aplicación de la Minería de Datos es interdisciplinaria lo que permite involucrarla a varias áreas de estudio. Un elemento importante para esta disciplina son los almacenes de datos, los cuales son una colección de datos orientada a un tema, integrada, no volátil y variable en el tiempo, que se utiliza para respaldar el proceso de toma de decisiones de una organización.

La Minería de Datos cuenta con distintas técnicas, cada una para realizar una tarea en específico y con sus propias características, estas se dividen en predictivos y descriptivos. Para hacer uso de estas técnicas se tiene software dedicado a realizar estas tareas, algunos ejemplos son Python, KNIME y Weka. La elección de la herramienta adecuada depende de la tarea específica y las necesidades del usuario, lo que subraya la importancia de desarrollar una herramienta integral que simplifique el proceso de Minería de Datos para usuarios no expertos.

Este trabajo presenta el desarrollo de una herramienta que permite aplicar técnicas descriptivas, con las cuales se pueden generar diferentes particiones de un conjunto de datos y a su vez elegir cuál partición, entre las creadas, es la mejor para un problema específico. Se pretende que con el diseño de esta herramienta, usuarios que no tengan un conocimiento amplio de los conceptos de Minería de Datos, puedan utilizarla en la toma de decisiones.

Abstract

Data Mining is the process of discovering meaningful patterns and relationships in large data sets, using statistical analysis techniques and machine learning algorithms. This discipline is used to extract useful information from large data sets. In addition, Data Mining can help identify hidden trends in data, which can improve decision making in areas such as predicting market trends, detecting fraud and evaluating employee performance.

The evolution of Mining goes from manual data analysis, where an experienced specialist was needed to make decisions, then operational systems, then database query systems that allow users to perform queries on a database to retrieve information, and finally information systems that are software systems designed to integrate and manage data from different sources, allowing the generation of reports and data analysis through disciplines such as Data Mining.

The application of Data Mining is interdisciplinary, which allows it to involve several areas of study. An important element for this discipline is data warehouses, which are a collection of topic-oriented, integrated, non-volatile and time-varying data used to support an organization's decision making process.

Data mining has different techniques, each one to perform a specific task and with its own characteristics, these are divided into predictive and descriptive. To make use of these techniques there is software dedicated to perform these tasks, some examples are Python, KNIME and Weka. The choice of the appropriate tool depends on the specific task and the user's needs, which underlines the importance of developing a comprehensive tool that simplifies the data mining process for non-expert users.

This document presents the development of a tool that allows the application of descriptive techniques, with which different partitions of a data set can be generated and in turn choose which partition, among those created, is the best for a specific problem. It is intended that with the design of this tool, users who do not have a broad knowledge of the concepts of Data Mining, can use it in decision making.

Índice general

Índice de Figuras

Índice de Tablas

Introducción

La Minería de Datos desde sus inicios ha sido un proceso muy importante en cualquier área donde se trabaje con grandes volúmenes de información, su valor principal radica en el procesamiento y análisis de dichas cantidades de información de manera automatizada o semi automatizada que, mediante diversas técnicas y algoritmos enfocados a problemas específicos, permite obtener y extraer información valiosa para brindar un conocimiento útil a la hora de tomar decisiones dentro del área en donde se utiliza, por ejemplo, empresas o instituciones que tienen la capacidad de generar históricos o almacenar grandes cantidades de información.

Antes de la aparición de la Minería de Datos, formalmente como se conoce hoy en día, los sistemas encargados de almacenar la información de las empresas, y las técnicas de análisis que se aplicaban, eran manuales, haciendo estos procesos tardados, costosos y deficientes, además de ser necesaria una persona con mucha experiencia en el área para poder realizar y tomar la decisión basada en los registros y la experiencia de sí misma dentro de esa área. Por esta razón, era más difícil y tardado acertar una buena decisión sea del tipo que sea, por ejemplo, acreditar un préstamo o crédito bancario a una persona sin tener un histórico de la misma o de personas similares, esto complicaba predecir si la persona sería un buen candidato o no para otorgarle ese beneficio.

Toda esta información es relevante, como por ejemplo en el caso de las tiendas en línea, aplicando técnicas de Minería de Datos se haría una predicción de lo que se le puede ofrecer a un cliente en particular, basado en el histórico de la empresa, donde se encuentran los registros de los productos que compró el año pasado, ayudando así a tomar la decisión y predecir qué productos son probables que cierto cliente adquiera. De esta forma se podría aplicar para cualquier área, seleccionando y usando de manera adecuada el mejor algoritmo y técnica de Minería de Datos para tomar la mejor decisión.

El modelo predictivo o clasificación supervisada se utiliza para clasificar los datos en categorías predefinidas. Este tipo de técnicas implican el uso de un conjunto de datos de entrenamiento que ya ha sido etiquetado. El algoritmo de clasificación utiliza este conjunto de datos para aprender a clasificar nuevos datos. Por ejemplo, se puede utilizar la clasificación supervisada para identificar correos electrónicos en correos deseados y no deseados.

Por otro lado, el modelo descriptivo o clasificación no supervisada no utiliza un conjunto de datos de entrenamiento etiquetado, por lo que se basa en la identificación de patrones y relaciones en los datos sin conocer las categorías correctas a priori. En la Minería de Datos no supervisada, el objetivo es encontrar grupos naturales dentro de los datos, lo que se conoce como agrupamiento o clustering. El algoritmo de clustering agrupa los datos en función de su similitud. Por ejemplo, se puede utilizar la Minería de Datos descriptiva para identificar grupos de clientes con comportamientos similares.

La clasificación y el agrupamiento (clustering) son dos técnicas de Minería de Datos importantes que se utilizan para analizar conjuntos de datos. Ambas técnicas tienen sus propias fortalezas y debilidades, y la elección de una técnica en particular dependerá del problema específico que se esté tratando de resolver.

La Minería de Datos puede ayudar a mejorar la toma de decisiones, optimizar los procesos, aumentar la rentabilidad, descubrir nuevas oportunidades, entre otros beneficios. Sin embargo, para realizar esta tarea se requiere de una herramienta especializada que ofrezca la aplicación de este tipo de algoritmos. Estas herramientas suelen ser complejas y requerir un conocimiento previo de todas las variables y características de los algoritmos, como por ejemplo, los parámetros, las métricas, las ventajas y desventajas, las limitaciones, etc. Por lo tanto, no están al alcance de todos los usuarios que podrían beneficiarse de la Minería de Datos, como son los gerentes, los analistas, los investigadores, los docentes, los estudiantes, etc.

Problemática

En la actual era digital, la cantidad de datos generados y almacenados ha alcanzado proporciones gigantescas en diversos campos, desde el comercio electrónico hasta la atención médica. La Minería de Datos emerge como una disciplina crucial para descubrir patrones, tendencias y conocimientos ocultos en estos vastos conjuntos de datos. Sin embargo, a pesar de su importancia, la adopción generalizada de técnicas de Minería de Datos se ha visto limitada por una barrera fundamental: la falta de herramientas accesibles y amigables para usuarios no expertos en el campo.

Los métodos tradicionales de Minería de Datos a menudo requieren conocimientos técnicos y habilidades de programación, lo que excluye a una gran cantidad de profesionales de diversas disciplinas que podrían beneficiarse enormemente de la capacidad de extraer información valiosa de sus datos. La carencia de soluciones intuitivas y asequibles ha obstaculizado el potencial de estas personas para aprovechar el poder de la Minería de Datos y tomar decisiones informadas basadas en evidencia.

Hasta la fecha, no se ha desarrollado una herramienta integral que combine tanto los algoritmos de agrupamiento como los índices de validación en una sola plataforma de manera accesible para usuarios no expertos. La ausencia de esta combinación en una única herramienta representa una brecha en el panorama actual de la Minería de Datos. La importancia de integrar ambos aspectos en una misma aplicación radica en la necesidad de brindar a los usuarios una experiencia completa y eficiente en el análisis descriptivo de datos. Al proporcionar una solución que englobe tanto la aplicación de algoritmos de agrupamiento como la evaluación confiable de los resultados a través de índices de validación, lo que facilita a usuarios no expertos la toma de decisiones informada basada en los patrones identificados en los datos. Esta integración no sólo simplifica el proceso de análisis, sino que también permite una comprensión más profunda de los resultados, lo que lleva a una toma de decisiones más precisa y fundamentada.

Justificación

Surge la necesidad de desarrollar herramientas que reduzcan las barreras de entrada a la Minería de Datos, permitiendo que usuarios no expertos tengan acceso a técnicas avanzadas de análisis sin requerir un conocimiento extenso en programación o estadística. Actualmente, no existe una aplicación que permita esto. En este sentido, la creación de una aplicación web, impulsada por Python, representa una solución potencialmente innovadora. Esta herramienta permitirá a los usuarios cargar conjuntos de datos y aplicar algoritmos de agrupamiento y validación en una interfaz amigable, visualizando los resultados de manera comprensible. Al presentar los índices de validación de manera comprensible y contextual, permitirá que los usuarios no expertos puedan informarse sobre la técnica más efectiva para su conjunto de datos.

Se desarrollará una nueva herramienta que permite a usuarios no expertos usar técnicas de Minería de Datos sin la necesidad de una herramienta especializada con muchas técnicas. Esta herramienta se basa en un sistema intuitivo y guiado que ayuda al usuario a identificar la técnica más adecuada para cada tipo de dato y objetivo, y presenta los resultados de forma sencilla y comprensible.

Así, los usuarios pueden obtener información valiosa de sus datos sin tener que aprender a usar una herramienta compleja o depender de un experto en Minería de Datos. Específicamente, orientar esta herramienta a usuarios no expertos, no solo facilitará el acceso a la Minería de Datos, sino que también abrirá nuevas posibilidades para la toma de decisiones basada en datos a una gran cantidad de sectores. Al romper con esas barreras técnicas y de conocimiento, esta herramienta tiene el potencial de brindar apoyo a personas y organizaciones para aprovechar al máximo el valor de sus datos.

La nueva herramienta propuesta tendrá las siguientes características: (i) una interfaz que guía al usuario paso a paso en el proceso de Minería de Datos. (ii) un sistema que ayude al usuario a medir la calidad de los resultados de una forma fácil y comprensible. (iii) una interfaz gráfica que muestra los resultados del análisis con gráficos, tablas y textos explicativos. (iv) una opción para exportar los resultados a diferentes formatos.

Objetivos

Objetivo General

- Desarrollar una aplicación con el uso de lenguaje para entornos web, que permita a los usuarios no expertos en términos de Minería de Datos aplicar algoritmos de agrupamiento y validación, para visualizar los resultados de manera interactiva.

Objetivos Específicos

- Analizar las tecnologías y herramientas de procesamiento de datos que ofrece Python, para determinar cuáles son adecuadas integrar en la aplicación propuesta, revisando las librerías existentes en dicho lenguaje.
- Comparar los frameworks web existentes para detectar cuál de estos es el óptimo para la realización del proyecto, considerando las herramientas que proporcionan.

- Sintetizar los algoritmos de agrupamiento y de validación en la aplicación propuesta, con el fin de generar resultados comprensibles para el usuario, utilizando la herramienta Python para su programación.
- Validar la aplicación propuesta aplicando instrumentos de medición a un grupo de usuarios para detectar mejoras en el desarrollo.

Estructura del documento

En este documento se encontrarán varios capítulos que abarcan diferentes aspectos de la tesis. En el primer capítulo, se encuentra el marco teórico, donde se explora en profundidad el concepto de Minería de Datos, sus diversas aplicaciones, así como los algoritmos fundamentales utilizados en este campo. Este capítulo proporciona una base sólida para comprender el contexto y la importancia de la Minería de Datos en la actualidad.

En el segundo capítulo, se presenta el estado del arte, donde se realiza una comparación de las aplicaciones existentes en el campo de la Minería de Datos.Se analizan las diferentes herramientas y tecnologías utilizadas en proyectos similares, destacando sus fortalezas y debilidades.

El tercer capítulo se centra en el desarrollo del proyecto, aplicando la metodología de programación extrema (XP) como enfoque principal. Se aborda detalladamente el proceso de desarrollo de la aplicación, desde la planificación inicial hasta la implementación y las pruebas.

En el cuarto capítulo, se presentan los resultados obtenidos. Este incluye capturas de pantalla de la aplicación desarrollada y los formularios utilizados para medir la usabilidad por parte de los usuarios.Se analizan los datos recopilados.

Finalmente, se encuentran las conclusiones del trabajo realizado, donde se resumen los hallazgos más importantes de la investigación y se discute posibles direcciones para futuros estudios en el campo. Además, se incluye una lista de referencias bibliográficas que respaldan el trabajo realizado a lo largo de la tesis.

Capítulo 1

Marco Teórico

Se abordarán algunos conceptos básicos de la Minería de Datos para proporcionar una visión general y para ayudar a comprender las ideas fundamentales detrás de esta disciplina. Así mismo, se presenta un conjunto de metodologías de desarrollo de software revisadas, con el fin de identificar aquella que pueda servir de guía en la propuesta de este trabajo.

PARTE I

Minería de Datos

La Minería de Datos es el proceso de descubrir patrones y relaciones significativas en grandes conjuntos de datos, utilizando técnicas de análisis estadístico y algoritmos de aprendizaje automático. Esta técnica se utiliza para extraer información útil de grandes conjuntos de datos, lo que puede ser útil en áreas como el marketing, la investigación de mercado, la medicina, la ciencia y la tecnología [8].

La justificación de su uso radica en su capacidad para ayudar a las empresas y organizaciones a tomar decisiones informadas basadas en datos históricos, lo que puede mejorar la eficiencia y la productividad, y al mismo tiempo reducir los costos y el riesgo. Además, la Minería de Datos puede ayudar a identificar patrones y tendencias ocultas en los datos, lo que puede mejorar la toma de decisiones en áreas como la predicción de tendencias del mercado, la detección de fraudes y la evaluación del rendimiento de los empleados.

El uso de esta idea de procesamiento inteligente de datos para la efectiva toma de decisiones, ha ido evolucionando con el pasar de los años a través del desarrollo de sistemas operacionales, sistemas de consultas de bases de datos y finalmente, sistemas informativos [8]. Dicha evolución se puede describir de acuerdo a las siguientes secciones.

I

ANÁLISIS MANUAL

Se refiere al proceso de explorar y analizar datos de forma manual, utilizando herramientas como hojas de cálculo y gráficos. Este enfoque puede ser lento y propenso a errores, especialmente al trabajar con grandes conjuntos de datos. Además, puede ser difícil identificar patrones y tendencias ocultos en los datos utilizando sólo herramientas manuales.

II

SISTEMAS OPERACIONALES

OLTP: Es un sistema que se utiliza para registrar y procesar transacciones de negocios en tiempo real. OLTP es una base de datos de producción que se utiliza para recopilar y actualizar datos transaccionales en tiempo real. Estos sistemas se utilizan comúnmente en entornos empresariales para realizar actividades cotidianas como procesamiento de pedidos, facturación y contabilidad.

OLAP: Es un sistema que se utiliza para analizar datos de negocios y generar informes basados en datos históricos. OLAP se utiliza para la toma de decisiones basada en datos y se enfoca en proporcionar información analítica sobre los datos de la empresa. Estos sistemas suelen tener un conjunto de datos más grande que OLTP y se utilizan para análisis de datos complejos, pronósticos y generación de informes.

III

SISTEMAS DE CONSULTAS DE BASES DE DATOS

Son sistemas de software que permiten a los usuarios realizar consultas en una base de datos para recuperar información específica. Estos sistemas pueden procesar consultas complejas que incluyen múltiples tablas y restricciones. Los sistemas de consultas de bases de datos también pueden incluir herramientas de visualización para ayudar a los usuarios a comprender los resultados de sus consultas.

IV

SISTEMAS INFORMATIVOS

Son sistemas de software diseñados para integrar y gestionar datos de diferentes fuentes, permitiendo la generación de informes y análisis de datos. Estos sistemas suelen utilizar bases de datos relacionales y permiten la creación de informes ad hoc para satisfacer las necesidades específicas de los usuarios. Los sistemas informativos también pueden incluir herramientas de análisis de datos para permitir la exploración de grandes conjuntos de datos.

La Minería de Datos es una disciplina que se apoya de diversas áreas de estudio, entre las que se incluyen estadística, inteligencia artificial, aprendizaje automático, bases de datos, visualización de datos y optimización. Esta disciplina aprovecha técnicas y algoritmos provenientes de estos campos para explorar y analizar conjuntos de datos extensos con el propósito de descubrir patrones y tendencias valiosas. Estas disciplinas en conjunto son las que se pueden incluir en los sistemas informativos.

En cuanto a la estadística, la Minería de Datos hace uso de sus técnicas para el análisis de datos y la validación de modelos. Esto implica la aplicación de pruebas de hipótesis, regresión, análisis de varianza, análisis multivariante y técnicas de muestreo estadístico en la exploración de datos.

Por otro lado, la inteligencia artificial y el aprendizaje automático desempeñan un papel esencial en la Minería de Datos al contribuir a la creación de modelos predictivos y descriptivos. Estas disciplinas incorporan herramientas como redes neuronales, árboles de decisión, algoritmos de agrupamiento y algoritmos de clasificación para llevar a cabo esta tarea.

La gestión y el almacenamiento de datos son aspectos cruciales en la Minería de Datos, y aquí es donde la base de datos entra en juego. Dado que esta disciplina se enfoca en la exploración y análisis de grandes conjuntos de datos, la gestión eficiente y el almacenamiento adecuado de los datos son fundamentales. Las bases de datos relacionales y no relacionales proporcionan las herramientas necesarias para esta gestión de datos.

Por último, la visualización de datos también desempeña un papel esencial en la Minería de Datos. Se utilizan técnicas de visualización para representar y comunicar de manera efectiva los patrones y tendencias descubiertos en los datos, lo que facilita su comprensión y su uso en la toma de decisiones.

La Minería de Datos, una técnica poderosa en el análisis de grandes conjuntos de datos, encuentra una amplia gama de aplicaciones prácticas en diversas áreas. Desde la investigación de mercado hasta la atención médica y la detección de fraudes, su utilidad es innegable.

En el ámbito del análisis de mercado, la Minería de Datos se convierte en una aliada fundamental. Permite la identificación de patrones y tendencias en el comportamiento de los consumidores, proporcionando a las empresas la información necesaria para tomar decisiones informadas sobre su estrategia de marketing y la introducción de nuevos productos al mercado.

Las redes sociales, por otro lado, son un campo propicio para la aplicación de la Minería de Datos. Esta herramienta permite analizar las interacciones en las redes sociales y descubrir patrones en el comportamiento de los usuarios. Estos insights ayudan a las empresas a comprender mejor a su audiencia y a adaptar su estrategia de marketing de manera más efectiva.

En la lucha contra el fraude, la Minería de Datos desempeña un papel crucial. Su capacidad para detectar patrones sospechosos en transacciones financieras y otras actividades permite prevenir el fraude y proteger los activos de las organizaciones.

En el sector de la salud, la Minería de Datos se utiliza para identificar patrones y tendencias en la salud de la población. Esto facilita a los profesionales de la salud el mejoramiento de los diagnósticos y tratamientos de enfermedades, contribuyendo así a una atención médica más efectiva y personalizada.

Estas aplicaciones de la Minería de Datos ilustran su versatilidad y su potencial para brindar soluciones valiosas en una amplia variedad de campos, mejorando la toma de decisiones y permitiendo un mayor entendimiento de los datos.

PARTE II

Técnicas de Minería de Datos

Como resultado de la Minería de Datos, los métodos de análisis de datos orientados al descubrimiento de conocimiento están reemplazando gradualmente el análisis de datos con fines de verificación. La principal diferencia entre los dos es que la última información se revela sin la formulación previa de supuestos. La aplicación automatizada de algoritmos de Minería de Datos facilita el descubrimiento de patrones de datos, lo que hace que este método sea mejor que el análisis de validación cuando se trata de explorar datos de repositorios grandes y altamente complejos. Estas nuevas tecnologías están en continua evolución gracias a la colaboración entre campos de investigación como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadísticas, visualización, recuperación de información y computación de alto rendimiento[13].

Existen dos tipos principales de técnicas: supervisadas y no supervisadas. En las técnicas supervisadas, el modelo se entrena utilizando un conjunto de datos etiquetado, donde se conocen las entradas y las salidas deseadas. El algoritmo aprende a relacionar las entradas con las salidas mediante la identificación de patrones y la construcción de un modelo predictivo. Por otro lado, en las técnicas no supervisadas, el modelo se enfrenta a datos sin etiquetar y se le pide que encuentre patrones o estructuras interesantes en los datos por sí mismo, sin ninguna guía externa. Estos métodos son útiles para descubrir la estructura inherente de los datos y pueden ser utilizados para segmentar los datos en grupos o clústeres, reducir la dimensionalidad o detectar anomalías. En la tabla 1.1 se muestran algunas de las técnicas de Minería de Datos en ambas categorías:

Supervisados	No supervisados
Clasificación	Reglas de asociación
Regresión	Agrupamiento

Tabla 1.1 Ejemplos de técnicas de Minería de Datos

La clasificación es un proceso de asignar etiquetas o categorías a datos basado en sus características. Se utiliza para predecir a qué clase o grupo pertenece un nuevo dato, mediante la construcción de modelos a partir de datos etiquetados previamente.

Por su parte, la tarea de Regresión es un método que se utiliza para predecir valores numéricos en función de datos previos. Se establece una relación matemática entre variables, permitiendo estimar un valor desconocido a partir de otros datos conocidos. Se aplica en pronósticos financieros, análisis de tendencias, y más.

Del lado de las técnicas no supervisadas, las reglas de asociación identifican patrones y relaciones en conjuntos de datos, revelando conexiones entre elementos. Son ampliamente utilizadas en análisis de mercado y recomendaciones de productos. Estas reglas indican la probabilidad de que un evento ocurra dado otro, permitiendo tomar decisiones informadas.

Y por ultimo el agrupamiento o clustering es un ejercicio descriptivo que, como su nombre lo indica, tiene como objetivo agrupar a los individuos disponibles bien conocidos como objetos en grupos o clusters con comportamientos similares. A diferencia de las tareas de clasificación, los grupos no se conocen de antemano, y eso es lo que se pretende crear. El objetivo es, por tanto, obtener grupos (clusters o conglomerados)con fines exploratorios[7].

Este trabajo se enfocará en aquellas técnicas destinadas al agrupamiento o segmentación de datos, a través de la propuesta de una herramienta que permita su aplicación y evaluación de una forma automatizada.

PARTE III

Agrupamiento

Es una técnica fundamental que implica la segmentación de un universo de objetos o datos en grupos o clusters, donde los miembros de un mismo grupo comparten características similares entre sí y difieren de los objetos en otros grupos. Este proceso se lleva a cabo con el objetivo de descubrir patrones,

estructuras o relaciones subyacentes en conjuntos de datos no etiquetados. El agrupamiento es una herramienta valiosa para la segmentación de clientes, la categorización de contenido, la detección de anomalías y la simplificación de grandes volúmenes de datos, lo que facilita la toma de decisiones informadas

I

ALGORITMOS BASADOS EN PARTICIÓN

El clustering es el proceso de agrupar en grupos bien conocidos como clusters, de tal forma que los objetos que comparten el mismo cluster presentan una similitud entre ellos y una baja disimilitud. Estos grupos pueden ser exclusivos, traslapes, probabilísticos y jerárquicos [7].

I. Kmeans

El algoritmo K-means, tambien conocidos como k-medias, creado por MacQueen en 1967 es el algoritmo de clustering más conocido y utilizado ya que es de muy simple aplicación y eficaz.

Sigue un procedimiento simple de clasificación de un conjunto de objetos en un determinado número K de clusteres, K determinado a priori, es decir, se define previamente al inicio del algoritmo. La representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato. Cada cluster por tanto es caracterizado por su centro o centroide que se encuentra en el centro o el medio de los elementos que componen el cluster [7].

O es un conjunto de objetos $D_n = (x_1, x_2, \ldots, x_n)$, para todo i, x_i reales y k, ν_1, los centros de los K cluster.

El algoritmo del K-means se realiza en 4 etapas:

1. Elegir aleatoriamente K objetos que forman así los K clusters iniciales. Para cada cluster k, el valor inicial del centro es $= xi$, con los xi únicos objetos de Dn pertenecientes al cluster.

$$s = argmin||u_k - x||^2 \tag{1.1}$$

2. Reasigna los objetos del cluster. Para cada objeto x, el cluster provisional que se le asigna es el que es más próximo al objeto, según una medida de distancia, habitualmente la medida euclidiana que es calculada de usando la formula 1.2: Distancia euclidiana:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{1.2}$$

3. Una vez que todos los objetos son ubicados en un cluster, recalcular los centros de K cluster.

4. Repetir las etapas 2 y 3 hasta que no se hagan más reasignaciones. Aunque el algoritmo termina siempre con la repetición de los centroides en distintias iteraciones, no se garantiza el obtener la solución óptima. El algoritmo es muy sensible a la elección aleatoria de los K centros iniciales. Esta es la razón por la que, se utiliza el algoritmo del K-means numerosas veces sobre un mismo conjunto de datos para intentar minimizar este efecto, sabiendo que a centros iniciales lo más espaciados posibles dan mejores resultados.

La Figura 1 indica un clustering ideal, en el cual se observa el centroide ubicado justo al centro de la aglomeración de cada cluster.

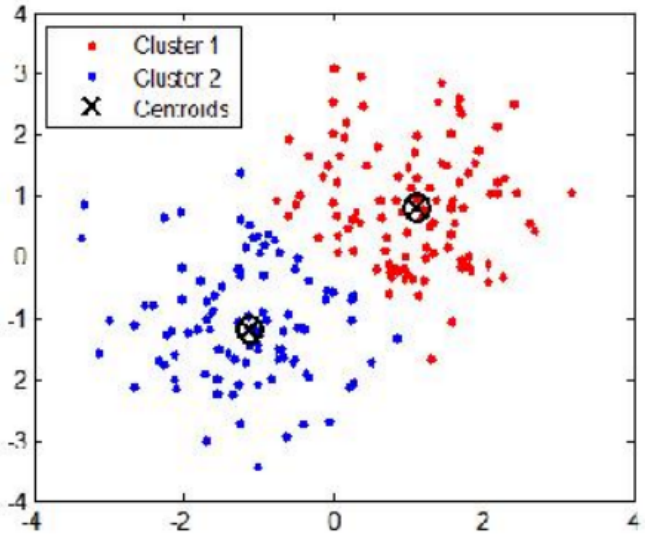

Figura 1 Agrupamiento ideal aplicando el algoritmo K-means. [4]

Ventajas del algoritmo K-means:

- K-Means es un algoritmo de agrupamiento simple y ampliamente utilizado.
- Es eficiente y escalable para grandes conjuntos de datos.
- Permite la identificación de grupos de datos de manera no supervisada.
- Es fácil de entender e implementar.
- Proporciona resultados interpretables, ya que los grupos son representados por centroides.

Desventajas del algoritmo Kmeans:

- Es necesario definir el número de clústeres y esta decisión puede afectar los resultados.
- Como la ubicación de los centroides iniciales es aleatoria, los resultados no pueden ser comparables y mostrar una falta de consistencia.
- K-means es susceptible a valores extremos porque distorsionan la distribución de los datos.
- Uno de los inconvenientes principales del K-means, además del hecho de que sea necesario realizar en sucesivas ocasiones el algoritmo para así tener el resultado más óptimo posible, es la necesidad de inicializar el número de prototipos al principio de la ejecución. Esto perjudica la eficacia del algoritmo ya que en la práctica, no se conoce a priori el número de cluster final.

II

ALGORITMOS JERÁRQUICOS

El propósito de los llamados métodos jerárquicos es agrupar grupos para formar un nuevo grupo o separar grupos existentes para formar otros dos grupos de tal manera que al realizar continuamente este proceso de agregación o división, se puede minimizar una cierta distancia o se puede reducir algo. maximizado. medida de la igualdad.
"Los métodos jerárquicos se subdividen en aglomerativos y disociativos. Cada una de estas categorías presenta una gran diversidad de variantes"[5].

- El enfoque aglomerativo, también conocido como bottom-up, inicia el análisis con tantos grupos como individuos. A partir de estas unidades iniciales se forman grupos que bien pueden ser llamados nodos hoja como nodos intermedios, estos van aumentando hasta incluir todos los casos tratados en un solo grupo al final del algoritmo, este nodo final puede ser conocido como raíz. En la figura 2 se observa un ejemplo de este algoritmo.

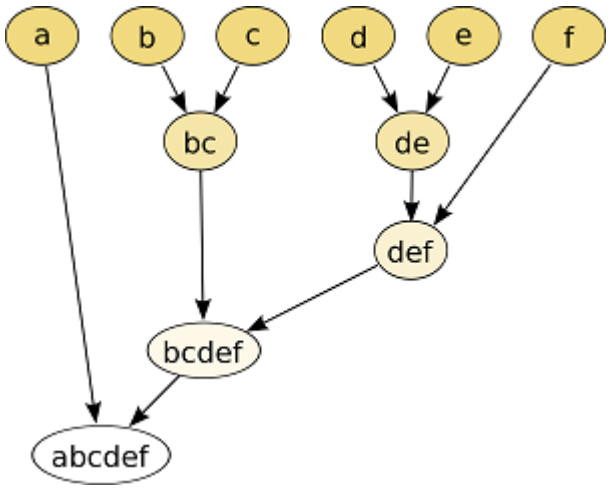

Figura 2 Representación del algoritmo aglomerativo. [6]

- Los métodos disociativos, también llamados descendentes, constituyen el proceso inverso al anterior.
  De un grupo que contiene todos los casos de tratamiento, ha creado un grupo cada vez más pequeño a partir de este grupo original a través de la división continua. Al terminar el algoritmo cada nodo hoja es interpretado como cada objeto procesado.

I. Algoritmo de aglomeración

Sea n el conjunto de individuos de la muestra. En el nivel inicial, se tiene $K = 0$ grupos, cada uno formado por un individuo. En el siguiente nivel, se agruparán los dos individuos que tengan la mayor similitud, calculada con la ecuación de la distancia euclidiana (1.2), resultando en $n - 1$ grupos. En consecuencia, en el nivel posterior se agruparán los dos individuos (o clusters ya formados) con menor distancia o mayor similitud. De esta forma, en el nivel L se tendrán $n - L$ grupos formados. Si se continúa agrupando de esta manera,

se llega al nivel $L = n - 1$, donde solo hay un grupo conocido como nodo raíz, formado por todos los individuos de la muestra.
Esta forma de crear nuevos grupos tiene la particularidad de que si se agrupan dos clústeres en un determinado nivel, ya están agrupados jerárquicamente para el resto de niveles.

Los métodos jerárquicos permiten la construcción de un árbol de clasificación, que recibe el nombre de dendrograma represenatado en la figura 3, en el cual se puede seguir de forma gráfica el procedimiento de unión seguido, mostrando que grupos se van uniendo, en que nivel concreto lo hacen, así como el valor de la medida de asociación entre los grupos cuando éstos se agrupan (valor que se llama nivel de fusión)[5].
El proceso de calcular distancias y unir nodos cercanos se repite hasta cumplir las siguientes metas:

- Se forma un solo grupo.
- Se alcanza el número de grupos prefijado.
- Se detecta, a través de un contraste de significación, que hay razones estadísticas para no continuar agrupando clusters, ya que los más similares no son lo suficientemente homogéneos como para determinar una misma agrupación.

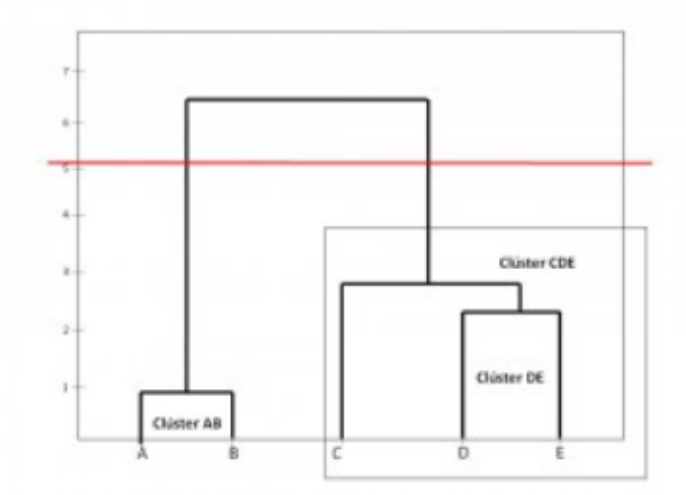

Figura 3 Dendograma con punto de fusión [3]

Ventajas del algoritmo aglomerativo:

- El número óptimo de clústeres se puede obtener por el mismo modelo, a través del dendrograma.

Desventajas del algoritmo aglomerativo:

- No es conveniente para grandes conjuntos de datos.

III

ALGORITMOS BASADOS EN DENSIDAD

Los algoritmos basados en densidad localizan zonas de alta densidad separadas por regiones de baja densidad [13].

I. DBScan

DBSCAN (Density Based Spatial Clustering of Aplications with Noise). Comienza seleccionando un punto t arbitrario, si t es un punto central, se empieza a construir un cluster alrededor de él por medio de calcular la distancia 1.2 que se tienen los objetos con respecto a los demas objetos, tratando de descubrir componentes denso-conectada, los cuales serán los que estén por debajo de el valor de epsilon, si no, se visita otro objeto del conjunto de datos.

Puntos centrales (core points) son aquellos tales que en su vecindad de radio Eps (ϵ), hay una cantidad de puntos mayor o igual que un umbral conocido como $MinPts$ especificado. Un punto borde o frontera tiene menos puntos que $MinPts$ en su vecindad, pero pertenece a la vecindad de un punto central. Un punto ruido (noise) es aquel que no es ni central ni borde. La figura 4 ilustra cada uno de esos conceptos en donde $MinPts \geq 4 \leq 6$, A son puntos centrales, B es un punto borde y C es un punto ruido.

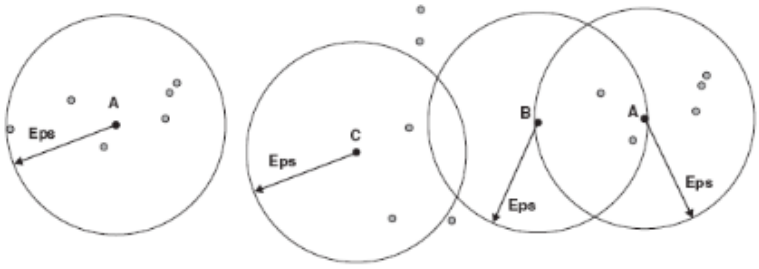

Figura 4 Definiciones de punto central, borde y ruido. [11]

Un punto q es directamente denso-alcanzable desde otro punto t (con relación a los parámetros MinPts y Eps) si t es un punto central y q pertenece a la vecindad de t.

Un punto q es denso-alcanzable desde un punto t si existe una cadena de puntos t_0, t_1, t_m, tales que $t_i - 1$ es directamente denso-alcanzable desde $t_i, 1 \leq i \leq m, t_0 = qyt_m = t$.

En consecuencia, los puntos centrales están en regiones de alta densidad, los puntos borde en la frontera de regiones densas y los puntos ruido en regiones de baja densidad. Este algoritmo busca clusters comprobando la vecindad de cada punto de la base de datos y va añadiendo puntos que son denso-alcanzables desde un punto central.

Dentro de su desarrollo se escuentran las siguientes etapas:

Entrada: X conjunto de datos

Eps: radio de la vecindad de cada punto

MinPts: número mínimo de puntos en una vecindad.

1. Seleccionar aleatoriamente un punto t
2. Si t es un punto central se forma un grupo alrededor de$t$ con todos los puntos denso-alcanzables desde t.
3. Si todos los puntos han sido visitados, terminar; si no, volver al paso 1.

En la imagen 5 se muestra una aplicación del algoritmo DBSCAN que contiene 3 clusters de diferentes colores y sus objetos de ruido están marcados de color morado [13].

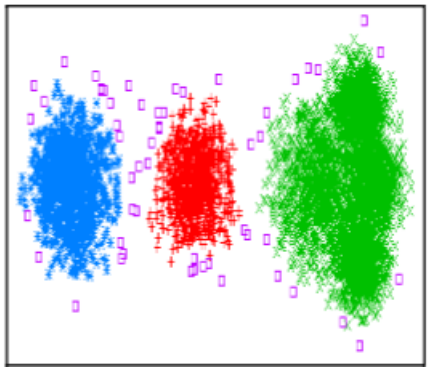

Figura 5 Ejemplo de algoritmo DBSCAN. [12]

Ventajas del algoritmo DBSCAN:

- El clúster no tiene que ser circular.
- Puede manejar muy bien el ruido y los valores atípicos.
- DBSCAN puede encontrar cualquier forma de clúster.

Desventajas del algoritmo DBSCAN:

- Si se tiene puntos de datos que forman clústeres de densidad variable, entonces DBSCAN no puede agrupar bien los puntos de datos, ya que el agrupamiento depende del parámetro épsilon y puntos mínimos, no se pueden seleccionar por separado para todos los clústeres.
- Es extremadamente sensible a los hiperparámetros. Un ligero cambio en los hiperparámetros puede llevar a un cambio drástico en el resultado.

PARTE IV

Índices de validación

Es de importancia evaluar el resultado de los algoritmos de clustering, sin embargo, es difícil definir cuando el resultado de un agrupamiento es bueno o medir la calidad de ese resultado de agrupamiento. Por esta razón existen técnicas e índices para la validación de un agrupamiento realizado.

De forma general existen dos clasificaciones la validación externa y la validación interna son las dos categorías más importantes para la validación de clustering. La principal diferencia es si se usa o no información externa para la validación, por ejemplo, información que no es producto de la técnica de agrupación utilizada, es principalmente usada para escoger un algoritmo de clustering óptimo sobre un data set específico. A diferencia de técnicas de validación externas, las de validación interna miden el clustering únicamente basadas en información de los datos, evaluando que tan buena es la estructura del clustering [10].

I

CALINSKI HARABASZ

El índice evalúa la cohesión(que tan unidos están) a través de la suma de las distancias de los elementos del grupo en relación con sus respectivos centroides. El criterio de separación es calculada a partir de la suma de las distancias entre los centroide de cada grupo y el centroide global del conjunto de datos. El costo computacional de esta función no es alta y supera, por lo general, a otras validaciones de conglomerados índices.

Cálculo del índice de Calinski-Harabasz: El índice CH para K número de conglomerados en un conjunto de datos $D = [d1, d2, d3, \ldots dN]$ se define como:

$$CH = [\frac{\sum_{k=1}^{k} n_k ||c_k - c||^2}{K-1}]/[\frac{\sum_{k=1}^{k} \sum_{k=1}^{n_k} ||d_i - c_k||^2}{N-K}] \tag{1.3}$$

donde, n_k y c_k son el número de puntos y centroide del grupo k respectivamente, c es el centroide global, N es el número total. de puntos de datos.

Un valor más alto del índice CH significa que los agrupamientos son densos y están bien separados, aunque no existe un valor de corte aceptable, pero si hay particiones mejores que otras en los casos de estudio. Es recomendable elegir aquella solución que dé un pico en el gráfico de líneas de los índices CH cuando se comparan diversas técnicas de agrupamiento. La figura 6 representa la comparación entre distintos agrupamientos y valores claros para elegir el mejor agrupamiento[15].

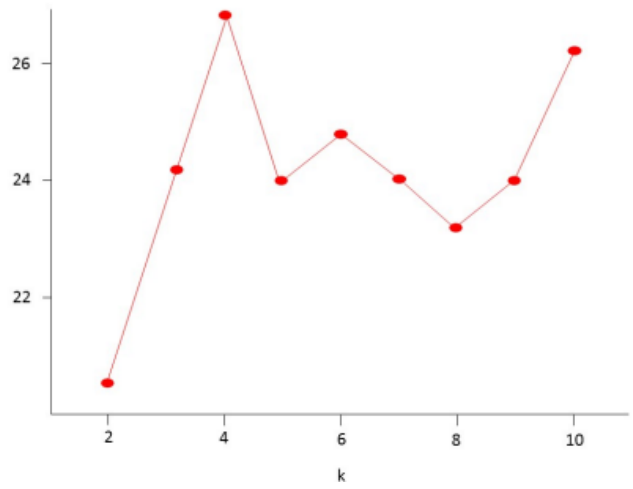

Figura 6 Gráfica de comparación de CH. [14]

El índice de validez de Davies-Bouldin (DB) es una medida utilizada en la evaluación de la calidad de agrupamiento (clustering) en análisis de datos. Este índice se basa en la idea de que un buen agrupamiento se caracteriza por tener grupos compactos y bien separados. El DB cuantifica la dispersión dentro de los grupos y la separación entre los grupos, proporcionando una evaluación global de la calidad del agrupamiento.

La fórmula del índice de Davies-Bouldin se calcula de la siguiente manera:

$$DB = \frac{1}{N} \sum_{i=1}^{N} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right) \tag{1.4}$$

Donde:

- N es el número de grupos en el agrupamiento.
- σ_i es la dispersión promedio dentro del grupo i, medida, por ejemplo, mediante la distancia media entre los puntos del grupo y su centroide.
- c_i es el centroide del grupo i.
- $d(c_i, c_j)$ es la distancia entre los centroides de los grupos i y j.

Un valor bajo de DB indica que los grupos son compactos y bien separados, lo que sugiere un agrupamiento de alta calidad. Por otro lado, un valor alto de DB indica que los grupos están dispersos o se superponen, lo que indica un agrupamiento deficiente.

En resumen, el índice de validez de Davies-Bouldin es una herramienta útil para evaluar la calidad de los agrupamientos en análisis de datos, proporcionando una medida cuantitativa de la separación y la dispersión de los grupos.

II

SILHOUETTE

El índice de validez de Silhouette es una métrica utilizada para evaluar la calidad de un agrupamiento (clustering) en análisis de datos. Este índice proporciona una medida de cuán similar es cada objeto de datos en su grupo (cluster) en comparación con otros grupos cercanos. El objetivo es cuantificar qué tan bien definidos y separados están los grupos en función de la distancia entre los objetos de datos.

La fórmula del índice de Silhouette se calcula para cada objeto de datos i en el conjunto de datos de la siguiente manera:

$$\text{Silhouette}(i) = \frac{b(i) - a(i)}{\text{máx}\{a(i), b(i)\}} \tag{1.5}$$

Donde:

- $a(i)$ es la distancia promedio entre el objeto de datos i y los demás objetos en el mismo grupo (cluster).
- $b(i)$ es la distancia promedio entre el objeto de datos i y los objetos en el grupo más cercano diferente al que pertenece.

El índice de Silhouette varía entre -1 y 1. Un valor alto indica que el objeto de datos está bien clasificado en su grupo y mal clasificado en otros grupos, lo que sugiere un buen agrupamiento. Por otro lado, un valor bajo o negativo indica que el objeto de datos podría estar mal clasificado en su grupo y/o que los grupos pueden estar superpuestos o mal separados.

Para obtener una medida global de la calidad del agrupamiento, se calcula el valor promedio de Silhouette para todos los objetos de datos en el conjunto de datos.

PARTE V

Metodologías para el desarrollo de software

En el ámbito del desarrollo de software existen diversas metodologías que guían el proceso de creación de aplicaciones. Estas metodologías varían en enfoque y en la forma en que gestionan el ciclo de vida del software. Algunas de las metodologías más conocidas incluyen Cascada, que sigue un enfoque secuencial y lineal; XP (Extreme Programming), que se centra en la colaboración y la adaptabilidad; el modelo en espiral, que incorpora iteraciones y evaluación constante; y SCRUM una metodología ágil que sirve para desarrollar, entregar y mantener productos complejos, priorizando la colaboración y adaptación constante. . Estas son solo algunas de las muchas metodologías disponibles, cada una con sus propias características y ventajas.

I

CASCADA

La metodología de cascada, también conocida como modelo en cascada, es un enfoque de desarrollo de software que se basa en una secuencia de etapas secuenciales y lineales para llevar a cabo un proyecto de desarrollo [9]. A continuación, se presenta una explicación detallada de las etapas de la metodología de cascada, las cuales se pueden observar de igual manera en la Figura 7:

1. Definición de Requisitos (Requerimientos): En esta fase inicial, se trabaja en la recopilación y documentación completa de los requisitos del proyecto. Esto implica la interacción con los clientes o usuarios finales para comprender sus necesidades y expectativas. El resultado es un documento detallado de especificaciones de requisitos que servirá como base para las fases posteriores.

2. Diseño (Design): Una vez que se han establecido los requisitos, se procede al diseño del sistema. En esta etapa, se crean diagramas, modelos y documentación técnica que describen cómo se estructurará y funcionará el software. Esto incluye el diseño de la arquitectura, la interfaz de usuario y otros aspectos importantes del sistema.

3. Implementación (Implementation): En esta fase, los desarrolladores comienzan a escribir el código del software en función de los diseños y especificaciones previamente establecidos. Se trata de la creación real del producto, donde se traducen los conceptos teóricos en código ejecutable.

4. Pruebas (Testing): Después de la implementación, se realizan pruebas exhaustivas para garantizar que el software funcione según lo previsto y cumpla con los requisitos definidos en la primera fase. Esto implica la identificación y corrección de errores o defectos.

5. Integración y Pruebas del Sistema (System Integration and Testing): En esta etapa, se ensamblan todas las partes del sistema y se realizan pruebas adicionales para verificar la interoperabilidad y el rendimiento general del software en su conjunto.

6. Entrega (Delivery): Una vez que el software ha superado todas las pruebas y se ha asegurado su calidad, se entrega al cliente o usuario final. En esta fase, se proporciona la documentación necesaria y se realiza la capacitación correspondiente si es necesario.

7. Mantenimiento (Maintenance): La fase de mantenimiento es continua y se extiende a lo largo del ciclo de vida del software. Incluye correcciones de errores, actualizaciones y mejoras según las necesidades cambiantes del cliente o del mercado.

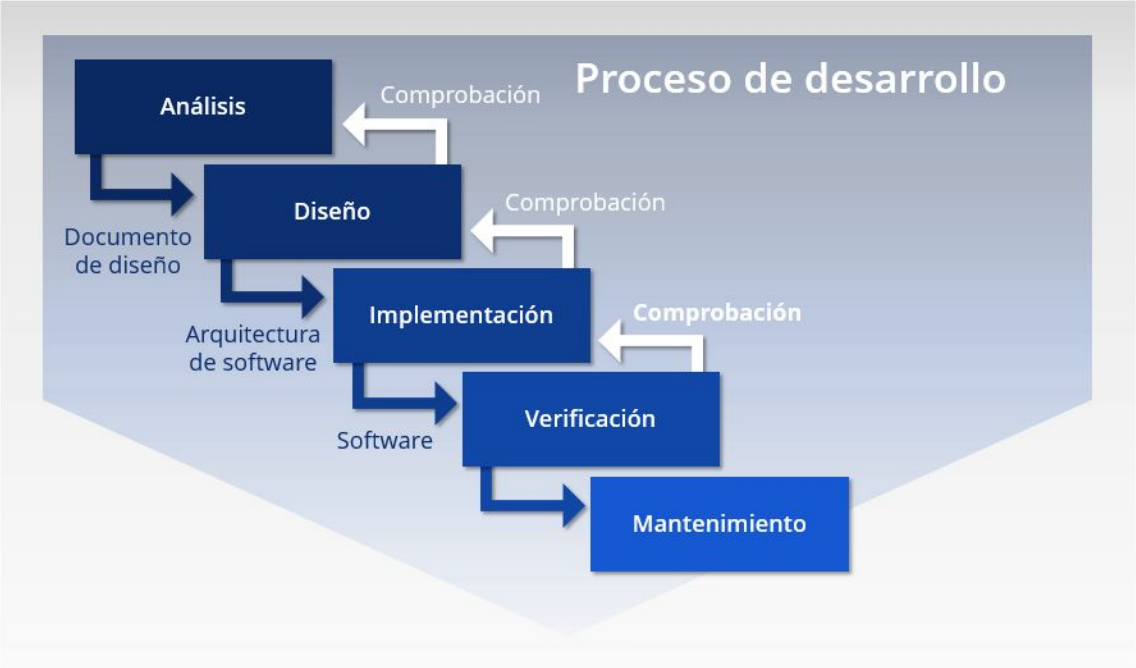

Figura 7 Metodología de cascada [9]

II
ESPIRAL

La metodología en espiral es un enfoque de desarrollo de software que combina elementos de la gestión de proyectos y la ingeniería de software. Fue propuesta por Barry Boehm en la década de 1980 y se basa en la idea de que el desarrollo de software es un proceso iterativo y continuo [16]. A continuación, se detallan las etapas de la metodología en espiral:

1. Determinación de Objetivos (Determining Objectives): En esta etapa, se establecen los objetivos generales del proyecto y se identifican las necesidades iniciales. Se analizan los requisitos del cliente y se definen los objetivos específicos que se deben lograr en esta iteración.

2. Análisis de Riesgos (Risk Analysis): La identificación y evaluación de riesgos es un componente clave de la metodología en espiral. Se realizan evaluaciones de riesgos técnicos, financieros y operativos. Se busca identificar posibles problemas y amenazas que puedan surgir durante el desarrollo del proyecto.

3. Desarrollo y Pruebas (Engineering and Testing): En esta fase, se desarrolla una versión del producto o sistema y se llevan a cabo pruebas rigurosas. Esto incluye la codificación, la implementación y la validación del software. El resultado es un producto funcional que puede ser evaluado por el cliente.

4. Evaluación del Cliente (Customer Evaluation): Una vez que se ha desarrollado una versión del producto, se presenta al cliente para su evaluación. El cliente proporciona retroalimentación valiosa sobre el producto y sus requisitos. Esta retroalimentación se utiliza para refinar y ajustar los objetivos del proyecto.

5. Planificación de la Siguiente Espiral (Planning the Next Spiral): Después de completar una iteración, se evalúan los resultados y se planifica la siguiente etapa del desarrollo. Esto puede implicar ajustar los objetivos, los recursos y los planes en función de lo aprendido en la iteración anterior.

La metodología en espiral se caracteriza por su flexibilidad y capacidad de respuesta a los cambios. Permite a las organizaciones adaptarse a condiciones cambiantes y mejorar la calidad del software a lo largo del tiempo. Cada iteración o "espiral" aborda una porción del proyecto y proporciona la oportunidad de refinar y ajustar el enfoque en función de la retroalimentación del cliente y la evaluación de riesgos como se puede ver en la figura 8.

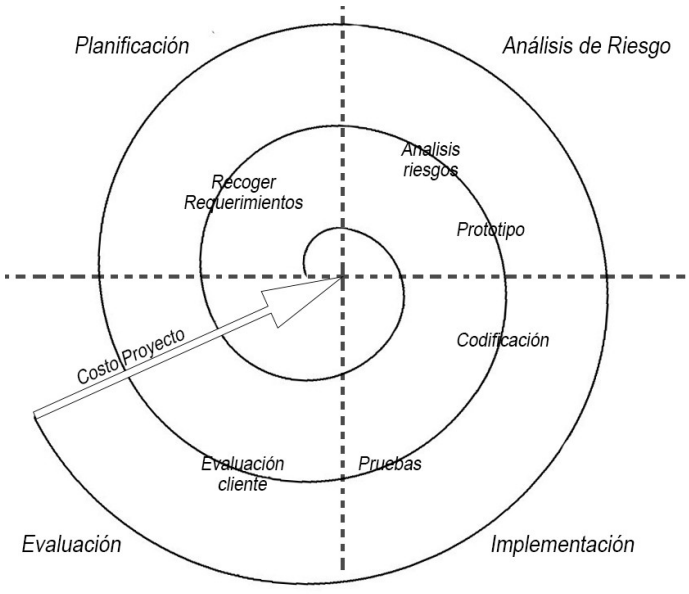

Figura 8 Metodología en espiral [1]

III

SCRUM

Scrum es un marco ágil de gestión de proyectos que se utiliza comúnmente en el desarrollo de software, aunque también puede aplicarse a otros proyectos complejos. Scrum se basa en principios de transparencia, inspección y adaptación, y se enfoca en la colaboración y la entrega de valor de manera iterativa 9. El esquema principal de esta metodología se ilustra en la Figura : [2] y las etapas que lo definen son:

1. Planificación del Producto (Product Backlog): En esta etapa inicial, se crea una lista de elementos de trabajo llamada "Product Backlog". Estos elementos son las características, funciones o tareas que deben realizarse en el proyecto. El Product Owner es responsable de priorizar y mantener esta lista.

2. Sprint Planning (Planificación de Sprint): Se selecciona un conjunto de elementos del Product Backlog para trabajar en un período de tiempo fijo llamado "Sprint". El equipo de desarrollo y el Product Owner colaboran para definir los objetivos del Sprint y planificar cómo se implementarán los elementos seleccionados.

3. Desarrollo del Sprint (Sprint Development): Durante el Sprint, el equipo de desarrollo trabaja en la implementación de los elementos seleccionados del Product Backlog. El trabajo se divide en tareas diarias y se realiza de manera colaborativa. El progreso se registra en el "Sprint Burndown Chart".

4. Reunión Diaria de Scrum (Daily Scrum): El equipo se reúne diariamente durante el Sprint para discutir el progreso, los obstáculos y las próximas tareas. Esta reunión breve y enfocada mejora la comunicación y la colaboración entre los miembros del equipo.

5. Revisión del Sprint (Sprint Review): Al finalizar el Sprint, se lleva a cabo una reunión de revisión en la que el equipo presenta el trabajo completado al Product Owner y a los stakeholders. Se recopila feedback y se ajusta el Product Backlog en consecuencia.

6. Retrospectiva del Sprint (Sprint Retrospective): También al finalizar el Sprint, el equipo realiza una retrospectiva para analizar su desempeño y buscar formas de mejora. Se identifican acciones concretas para optimizar el proceso en el próximo Sprint.

7. Iteración (Siguiente Sprint): El ciclo se repite con un nuevo Sprint. El equipo selecciona nuevos elementos del Product Backlog y trabaja en ellos de manera iterativa. El proceso continúa hasta que se alcanza el objetivo del proyecto.

Scrum fomenta la flexibilidad, la adaptabilidad y la colaboración continua. A medida que se completan los Sprints, el producto evoluciona y se ajusta para satisfacer las necesidades cambiantes del cliente. El marco Scrum es altamente utilizado en la industria para proyectos ágiles y se ha convertido en un estándar en el desarrollo de software.

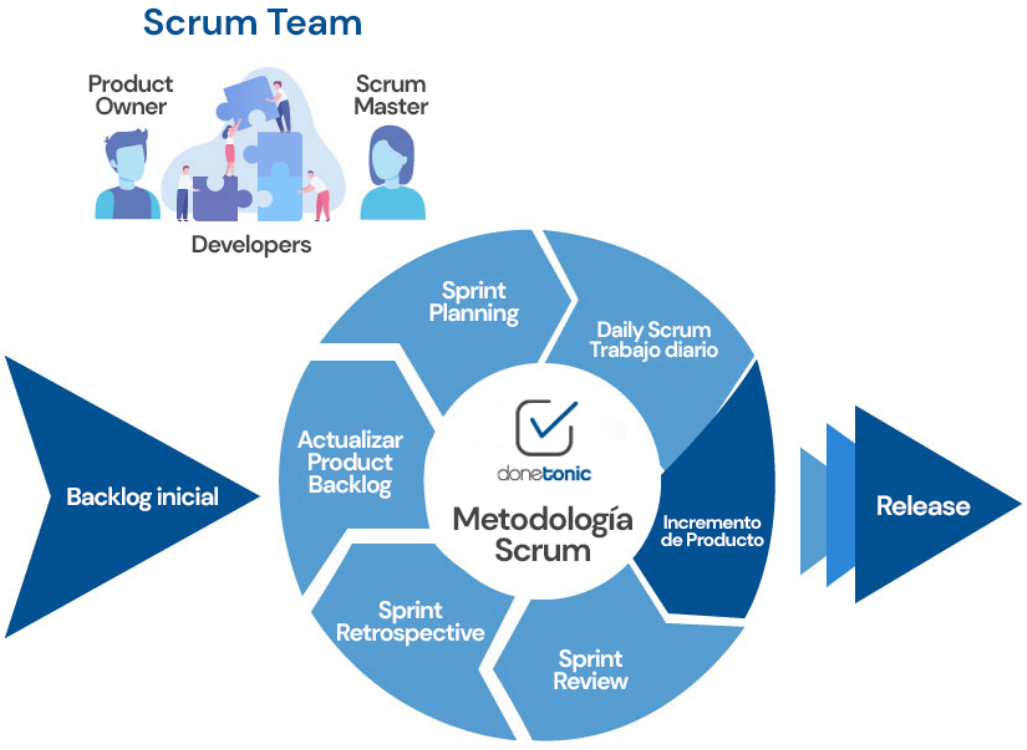

Figura 9 Metodología scrum [2]

IV

XP

La Programación Extrema (XP) es posiblemente el método ágil más conocido y ampliamente utilizado. El nombre fue acuñado por Beck (Beck, 2000) debido a que el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo, y con la participación del cliente en niveles «extremos». [17]

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas.

Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema.

1. Planificación: Se identifican los requisitos iniciales y se establece una visión clara del proyecto. Durante esta fase, el equipo de desarrollo colabora estrechamente con el cliente para comprender sus necesidades y prioridades. Se definen historias de usuario y se establecen los criterios de aceptación para cada una. Además, se identifican posibles riesgos y se elabora un plan para abordarlos. Esta fase también implica la configuración del entorno de desarrollo y la preparación de herramientas y recursos necesarios para comenzar el trabajo.

2. Dieseño: Creación de soluciones efectivas para cumplir con los requisitos establecidos en la fase de exploración. Durante esta etapa, se elabora un diseño detallado que incluye la arquitectura del sistema, la estructura de las clases y componentes, así como los diagramas de flujo y otros artefactos visuales necesarios. Se fomenta la comunicación constante entre los miembros del equipo para garantizar que el diseño sea claro y comprensible para todos.

3. Codificación: Se implementan las soluciones diseñadas durante la fase anterior. Se enfatiza la colaboración y el trabajo en equipo, con frecuentes revisiones de código y pruebas unitarias integradas en el proceso de desarrollo. Se sigue un enfoque de desarrollo impulsado por pruebas (TDD), donde se escriben las pruebas antes de escribir el código de producción. Esto garantiza que el código cumpla con los requisitos y funcione correctamente desde el principio.

4. Pruebas: Se lleva a cabo una serie de pruebas exhaustivas para garantizar la calidad y el correcto funcionamiento del software desarrollado. Esto incluye pruebas unitarias, pruebas de integración y pruebas de aceptación, entre otras. Las pruebas se realizan de manera continua y se automatizan tanto como sea posible para garantizar una retroalimentación rápida sobre el estado del producto.

5. Lanzamiento: Si hemos llegado a este punto, significa que hemos probado todas las historias de usuario o mini-versiones con éxito, ajustándonos a los requerimientos del clientes. Tenemos un software útil y podemos incorporarlo en el producto.

La programación extrema implica varias prácticas que se ajustan a los principios de los métodos ágiles como se observa en la Figura 10.

Figura 10 Metodología XP [17]

V

METODOLOGÍA ELEGIDA PARA EL PROYECTO

La elección de la metodología adecuada es una decisión crucial que puede influir en el éxito de un proyecto. Cuando nos encontramos en la duda de decidir qué enfoque adoptar, es esencial realizar un análisis exhaustivo de las diversas metodologías disponibles. En este contexto, después de una evaluación cuidadosa y una revisión detallada de las opciones disponibles, se ha tomado la decisión de implementar la Metodología de Programación Extrema (XP).

I. ¿Por qué XP?

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas.

Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema.

La programación extrema implica varias prácticas que se ajustan a los principios de los métodos ágiles:

1. El desarrollo incremental se lleva a cabo través de entregas del sistema pequeñas y frecuentes y por medio de un enfoque para la descripción de requerimientos basado en las historias de cliente o escenarios que pueden ser la base para el proceso de planificación.

2. La participación del cliente se lleva a cabo a través del compromiso a tiempo completo del cliente en el equipo de desarrollo. Los representantes de los clientes participan en el desarrollo y son los responsables de definir las pruebas de aceptación del sistema.

3. El interés en las personas, en vez de en los procesos, se lleva a cabo a través de la programación en parejas, la propiedad colectiva del código del sistema y un proceso de desarrollo sostenible que no implique excesivas jornadas de trabajo.

4. El cambio se lleva a cabo a través de las entregas regulares del sistema, un desarrollo previamente probado y la integración continua.

5. El mantenimiento de la simplicidad se lleva a cabo a través de la refactorización constante para mejorar la calidad del código y la utilización de diseños sencillos que no prevén cambios futuros en el sistema.

La Tabla 1.2 presenta un resumen de las características consideradas en la aplicación de la metodología.

Principio o Práctica	**Descripción**
Planificación Incremental	Los requerimientos se registran en tarjetas de historias y las historias a incluir en una entrega se determinan según el tiempo disponible y su prioridad relativa. Los desarrolladores dividen estas historias en tareas de desarrollo.
Entregas Pequeñas	El mínimo conjunto de funcionalidad que proporciona valor de negocio se desarrolla primero. Las entregas del sistema son frecuentes e incrementalmente añaden funcionalidad a la primera entrega.
Diseño Sencillo	Se lleva a cabo el diseño necesario para cumplir los requerimientos actuales.
Desarrollo Previamente Probado	Se utiliza un sistema de pruebas de unidad automatizado para escribir pruebas para nuevas funcionalidades antes de que éstas se implementen.
Refactorización	Se espera que todos los desarrolladores refactoricen el código continuamente tan pronto como encuentren posibles mejoras en el código. Esto conserva el código sencillo y mantenible.
Programación en Parejas	Los desarrolladores trabajan en parejas, verificando cada uno el trabajo del otro y proporcionando la ayuda necesaria para hacer siempre un buen trabajo.

Principio o Práctica	**Descripción**
Propiedad Colectiva	Las parejas de desarrolladores trabajan en todas las áreas del sistema, de modo que no desarrollen islas de conocimientos y todos los desarrolladores posean todo el código.
Integración Continua	En cuanto acaba el trabajo en una tarea, se integra en el sistema entero. Después de la integración, se deben pasar al sistema todas las pruebas de unidad.
Ritmo Sostenible	No se consideran aceptables grandes cantidades de horas extras, ya que a menudo el efecto que tienen es que se reduce la calidad del código y la productividad a medio plazo.
Cliente Presente	Debe estar disponible al equipo de la XP un representante de los usuarios finales del sistema (el cliente) a tiempo completo. En un proceso de la programación extrema, el cliente es miembro del equipo de desarrollo y es responsable de formular al equipo los requerimientos del sistema para su implementación.

Tabla 1.2 Principios y Prácticas de la Programación Extrema (XP)

Capítulo 2

Estado del arte

Existen diversas plataformas y herramientas similares que ofrecen capacidades de Minería de Datos. Además de Python, que es altamente versátil y ampliamente utilizado en este ámbito, existen alternativas como MATLAB, conocido por su potencia en cálculos numéricos y procesamiento de datos. Weka es otra opción popular, una plataforma de código abierto con una interfaz gráfica intuitiva para la Minería de Datos. R, por otro lado, es un lenguaje de programación y entorno estadístico robusto, ampliamente preferido por los estadísticos y analistas de datos. Cada una de estas plataformas tiene sus propias ventajas y características, y la elección dependerá de los requisitos y preferencias del usuario

PARTE I

MATLAB

Es un entorno de programación y desarrollo utilizado en ingeniería, ciencia y matemáticas. Su versatilidad lo hace valioso para una amplia gama de aplicaciones, proporcionando una interfaz interactiva y una gran cantidad de funciones para el análisis de datos, la simulación numérica y la implementación de algoritmos.

Es una plataforma ampliamente reconocida que ofrece una variedad de métodos de Minería de Datos para tareas de clasificación y validación. En cuanto a clasificación, proporciona algoritmos como SVM (Support Vector Machines), árboles de decisión, redes neuronales, entre otros. Para la validación, incluye métodos como el índice Calinski-Harabaz y Silhouette.

Los usuarios pueden aplicar estos algoritmos mediante su lenguaje de programación amigable, construyendo scripts o funciones personalizadas. Con MATLAB, es posible realizar tareas de preprocesamiento de datos, visualización de resultados y evaluación de modelos.

La licencia de MATLAB puede resultar costosa y no es de código abierto, lo que puede limitar su accesibilidad. La evaluación de su rendimiento depende de factores como la disponibilidad de recursos financieros y las necesidades específicas del proyecto.

PARTE II

WEKA

Se trata de una plataforma de Minería de Datos de código abierto utilizado para análisis y aprendizaje automático. Es ampliamente utilizado en la comunidad de investigación, ofreciendo algoritmos para tareas de clasificación, regresión, clustering y asociación.

Incluye algoritmos como Naïve Bayes, Random Forest, Support Vector Machines y Decision Trees, entre otros. Es importante destacar que Weka no proporciona herramientas específicas para la evaluación de particiones en conjuntos de datos. Aunque es capaz de realizar tareas como la clasificación, la regresión y el clustering, la evaluación de particiones, como la validación cruzada o la evaluación de la calidad de las particiones en clustering, requiere del uso de herramientas adicionales o la implementación de procedimientos externos.

Los usuarios pueden aprovechar Weka a través de su interfaz gráfica de usuario intuitiva, lo que facilita la configuración y ejecución de experimentos de Minería de Datos sin necesidad de programación. Permite a los usuarios desarrollar y probar modelos personalizados.

Una ventaja clave es su naturaleza de código abierto, lo que la hace accesible y adaptable a una amplia variedad de aplicaciones. Sin embargo, puede carecer de algunas características avanzadas disponibles en otras herramientas comerciales. La evaluación de Weka se basa en la facilidad de uso, el presupuesto y los requerimientos del usuario.

PARTE III

Python

Es un lenguaje de programación versátil y ampliamente adoptado que ha ganado popularidad en ciencia de datos y aprendizaje automático. Su facilidad de uso y bibliotecas especializadas lo convierten en una opción atractiva para diversas aplicaciones.

Ofrece una amplia gama de bibliotecas y herramientas para tareas de agrupamiento, clasificación y validación. Algunas de las bibliotecas más utilizadas incluyen Scikit-Learn, que proporciona algoritmos de clasificación como SVM, árboles de decisión y Naïve Bayes, así como métricas de validación como Silhouette. Otros paquetes como TensorFlow y Keras se utilizan para redes neuronales.

Los usuarios pueden aprovechar Python escribiendo scripts personalizados o utilizando entornos interactivos como Jupyter Notebook. Python es altamente flexible y extensible, permitiendo la implementación de algoritmos personalizados y la integración con otras bibliotecas de visualización y procesamiento de datos.

Una ventaja clave es que Python es de código abierto, lo que lo hace accesible y económico. Sin embargo, puede requerir un conocimiento de programación más profundo en comparación con herramientas con interfaces gráficas. Python para Minería de Datos depende de la experiencia en programación.

PARTE IV

R

Lenguaje de programación orientado al análisis estadístico y la visualización de datos. Se ha convertido en una opción popular en estadísticas y ciencia de datos debido a su amplia gama de paquetes especializados.

Incluye algoritmos como Random Forest, SVM, Naïve Bayes y Decision Trees, junto con métricas de validación como el índice Calinski-Harabaz y Silhouette.

Los usuarios pueden aplicar estos métodos mediante scripts y funciones personalizadas, aprovechando la flexibilidad y la comunidad activa de R. Esta herramienta también destaca por su capacidad de visualización de datos, lo que facilita la exploración y presentación de resultados.

Puede requerir conocimientos de programación y estadísticas para un uso efectivo. R es de código abierto y gratuito, lo que lo hace económicamente accesible. La elección de R dependerá de la experiencia del usuario.

PARTE V

KNIME

Es una plataforma de análisis de datos visual que permite a los usuarios conectar nodos para procesar datos y ejecutar análisis. Ofrece un enfoque modular para el procesamiento y análisis de datos, facilitando la creación de flujos de trabajo complejos.

Proporciona una amplia gama de algoritmos, incluyendo árboles de decisión, regresión logística, redes neuronales y SVM.Ofrece métricas como Silhouette, índice Calinski-Harabaz y muchas otras.

se destaca por su interfaz gráfica de usuario, que permite a los usuarios crear flujos de trabajo visualmente sin necesidad de escribir código, lo que facilita su uso para aquellos sin experiencia en programación. También es altamente extensible, lo que permite la integración de bibliotecas y herramientas externas.

En comparación con lenguajes de programación como Python o R, KNIME puede ser menos eficiente en términos de velocidad de procesamiento.

PARTE VI

¿Cúal es mejor?

Después de analizar las diferentes herramientas que se ofrecen al público hoy en día se puede observar las diferentes características que contienen dichas herramientas, con lo cual se puede generar una comparativa de estas. La Tabla 2.1 presenta una serie de aspectos que son relevantes considerar cuando se quiere elegir una herramienta en particular, además agrega la comparativa contra la aplicación desarrollada en este trabajo.

Las características que se consideraron son:

- Ampliamente utilizado en análisis numérico:

  Son comúnmente utilizadas en aplicaciones que involucran análisis numérico, que es un campo de las matemáticas y la informática que se enfoca en el procesamiento de datos numéricos y cálculos.

- Enfoque en Minería de Datos:

  Están diseñadas principalmente para tareas de Minería de Datos, que implica descubrir patrones y relaciones en conjuntos de datos grandes y complejos.

- Versatilidad: Se puede utilizar en una variedad de aplicaciones, lo que significa que no está limitada a una sola tarea o dominio de aplicación.

- Enfoque en estadística: Se centran en estadísticas y análisis de datos, lo que implica la recopilación, análisis y presentación de datos para obtener información significativa.

- Interfaz gráfica intuitiva: Ofrecen una interfaz gráfica de usuario que es fácil de entender y usar, lo que facilita su utilización, especialmente para aquellos que no son expertos en programación.

- Funciones avanzadas: Pueden incluir capacidades de procesamiento de datos más sofisticadas o algoritmos más complejos.

- Bibliotecas robustas: Tienen bibliotecas sólidas, lo que significa que proporcionan una amplia gama de funciones y herramientas para el análisis de datos.

- Para usuarios no expertos: Diseñada para ser utilizada por personas que no son expertas en análisis de datos, lo que significa que es más accesible para un público no técnico.
- Algoritmos de agrupamiento y validación juntos: Algoritmos tanto para agrupamiento (como la identificación de patrones en datos no etiquetados) como para validación (algoritmo más viable) en una sola plataforma.

Características	Matlab	Weka	Python	R	KNIME	WebMinerX
Ampliamente utilizado en análisis numérico	✓					✓
Enfoque en Minería de Datos		✓				✓
Versatilidad			✓			✓
Enfoque en estadística				✓		✓
Interfaz gráfica intuitiva		✓			✓	✓
Funciones avanzadas	✓					✓
Bibliotecas robustas		✓	✓			✓
Para usuarios no expertos						✓
Algoritmos de agrupamiento y validación juntos	✓		✓			✓

Tabla 2.1 Comparación de Características

Capítulo 3

Desarrollo de la aplicación propuesta

La aplicación de una metodología en el desarrollo de cualquier tipo de software es crucial. Proporciona estructura, organización y eficiencia al proyecto, estableciendo roles, plazos y recursos. Facilita la reproducción consistente de resultados.

Como se ya se mencionó en el Capítulo 1, la metodología elegida para este proyecto es XP, por lo que a continuación se detalla la ejecución de cada una de sus etapas.

PARTE I

Planificación

La etapa de planificación se divide en tres componentes fundamentales para garantizar el éxito del proyecto. En primer lugar, se centra en la comprensión de los requerimientos del usuario, lo que implica la identificación y análisis de las necesidades y expectativas del usuario. Esto es esencial para asegurarse de que el producto final cumpla con las especificaciones y brinde la funcionalidad requerida.

En segundo lugar, se abordan los requerimientos propios del equipo de desarrollo, donde se definen las restricciones técnicas, la prioridad de las funcionalidades y fechas de entrega. Esto es crucial para establecer un marco de trabajo claro y realista en el que se desarrollará el software.

Finalmente, la etapa también involucra la planificación y ejecución de las iteraciones, siguiendo la metodología XP (Programación Extrema). Durante estas iteraciones, se llevarán a cabo las actividades de diseño, desarrollo y pruebas de manera incremental y colaborativa.

I

IDENTIFICAR LAS FUNCIONES PRINCIPALES DE LA APLICACIÓN

En esta primera tarea se identifican las principales funciones que tendrá la aplicación a desarrollar. Estas funciones se describen a continuación.

I. Carga de datos

Este sistema permite a los usuarios cargar conjuntos de datos desde archivos CSV y archivos ARFF (Attribute-Relation File Format), formatos comunes en la Minería de Datos y el aprendizaje automático. Esto facilita la importación de datos a la plataforma. Además, ofrece capacidades de validación de datos específicas para cada formato de archivo, lo que garantiza que los datos cargados sean integrales y de alta calidad, lo que es esencial para garantizar la confiabilidad de los análisis posteriores. Los usuarios tienen la posibilidad de visualizar una vista previa de los datos antes de confirmar la carga, lo que les permite revisar los datos y asegurarse de que se han importado correctamente.

II. Preprocesamiento de datos

La herramienta permite mapear y transformar datos antes de su procesamiento, lo que incluye opciones como la selección de columnas relevantes o la conversión de tipos de datos. Esto facilita la preparación de los datos para su análisis. Además, proporciona una gestión de errores eficaz y notificaciones claras en caso de problemas durante la carga de datos, lo que ayuda a los usuarios a identificar y abordar rápidamente cualquier problema que pueda surgir durante el proceso de carga.

III. Aplicación de una variedad de algoritmos de Minería de Datos

El sistema permite a los usuarios realizar una serie de acciones, que incluyen la selección de algoritmos desde una lista predefinida. Además, pueden configurar parámetros específicos del algoritmo según sus necesidades. Luego, ejecutan los algoritmos en el conjunto de datos y pueden realizar un seguimiento del progreso de la ejecución. Además, tienen la capacidad de visualizar tanto los resultados intermedios como los resultados finales de los algoritmos aplicados al conjunto de datos.

IV. Visualización de resultados

La plataforma abarca la creación de gráficos y gráficos interactivos que permiten mostrar de manera efectiva los resultados, incluyendo diagramas y representaciones visuales. Los usuarios tienen a su disposición herramientas que les permiten explorar y filtrar los datos visualizados, lo que facilita el análisis en profundidad.

V. Descarga de resultados

la plataforma ofrece la posibilidad de exportar los gráficos y los resultados en una variedad de formatos, como imágenes o informes en formato PDF.

II

REQUERIMIENTOS DEL EQUIPO DE TRABAJO

Esta sección garantiza que el proyecto sea viable y se mantenga dentro de los límites de tiempo y recursos disponibles, evitando expectativas poco realistas. Además, la priorización de funcionalidades permite enfocarse en las más críticas y valiosas, entregando valor de manera eficiente y alineando el proyecto con los objetivos específicos.

I. Identificar las Funcionalidades Clave

Llevar a cabo una revisión detallada de todas las funcionalidades propuestas para la aplicación. El objetivo principal es identificar aquellas características que son esenciales para el funcionamiento básico de WebMinerX y que aportan un valor fundamental al producto.

II. Clasificar las Funcionalidades

Categorizar en función de su importancia y su impacto en los objetivos del proyecto. Esto permite priorizar y asignar recursos de manera eficiente.

III. Definir una Lista Priorizada

Creación de una lista ordenada de funcionalidades, priorizándolas desde las más importantes hasta las menos importante. Esta priorización se basa enla importancia estratégica, la demanda del usuario y la complejidad técnica.

IV. Asignar Fechas de Entrega

Fechas tentativas de entrega a cada funcionalidad, considerando su prioridad y dependencias. Esto facilita la planificación de las iteraciones del proyecto y determina qué características se implementarán en cada ciclo

III

IDENTIFICACIÓN DE ITERACIONES A REALIZAR

Se llevarán a cabo un total de seis iteraciones utilizando la metodología XP (Programación Extrema). Estas iteraciones se planificarán y ejecutarán de manera incremental y colaborativa a lo largo del proyecto.

I. Iteración 1 - Entrega Inicial

- Proporcionar una versión básica funcional de la aplicación que permita a los usuarios cargar datos y aplicar el algoritmo K-Means para la Minería de Datos.
- Carga de datos (CSV y ARFF).
- Aplicación del algoritmo K-Means.
- Visualización básica de los resultados.

II. Iteración 2 - Mejora de la Funcionalidad de K-Means

- Refinar y mejorar la funcionalidad de K-Means y permitir a los usuarios ajustar parámetros y explorar resultados en profundidad.
- Parámetros configurables para el algoritmo K-Means.
- Visualización avanzada de resultados, incluyendo gráficos interactivos.

III. Iteración 3 - Incorporación de DBSCAN

- Agregar la funcionalidad de DBSCAN como otro algoritmo de Minería de Datos disponible para los usuarios.
- Implementación del algoritmo DBSCAN.
- Interfaz para seleccionar y configurar DBSCAN.
- Visualización de resultados DBSCAN.

IV. Iteración 4 - Validación de Clústeres

- Introducir la validación de clústeres como una característica importante.
- Implementación de algoritmos de validación Silhouette, Calinski y Davies-Bouldin.
- Visualización de métricas de validación junto a los resultados.

V. Iteración 5 - Incorporación de Jerárquico

- Agregar el algoritmo de clustering jerárquico para ofrecer a los usuarios más opciones de algoritmos.
- Implementación del algoritmo de clustering jerárquico.
- Interfaz para seleccionar y configurar el clustering jerárquico.
- Visualización de resultados jerárquicos.

VI. Iteración 6 - Mejoras de Usabilidad y Exportación

- Realizar mejoras en la experiencia del usuario y permitir la exportación de resultados.
- Mejoras en la interfaz de usuario para una experiencia más intuitiva.
- Opción para exportar resultados.
- Obtener retroalimentación de los usuarios y realizar ajustes finales antes de la versión completa.
- Habilitar una versión de prueba para usuarios seleccionados.
- Recopilar retroalimentación y realizar ajustes en función de las sugerencias de los usuarios.

VII. Iteraciones Futuras

- Continuar mejorando la aplicación en función de la retroalimentación de los usuarios y las necesidades emergentes.
- Incorporación de más algoritmos de Minería de Datos.
- Funciones de preprocesamiento de datos.
- Integración con sistemas de almacenamiento en la nube.
- Mejoras en la escalabilidad y rendimiento.

PARTE II

Diseño Inicial

Se realiza un diseño inicial de la arquitectura de la aplicación y se establecen las tecnologías a utilizar. Además, se definen los componentes clave y cómo interactuarán entre sí. Es necesario seleccionar las tecnologías y herramientas necesarias, como frameworks de Python y bibliotecas de visualización, así como la creación de prototipos de las interfaces de usuario para la aplicación WebMinerX. El objetivo principal es obtener retroalimentación temprana de los usuarios y validar la usabilidad y la experiencia del usuario antes de realizar una implementación completa.

INTERFAZ DE USUARIO (FRONTEND)

Para crear una interfaz de usuario amigable y visualmente atractiva, se utilizarán tecnologías como HTML, CSS y JavaScript. Esta interfaz permitirá a los usuarios cargar datos, configurar parámetros de algoritmos y visualizar los resultados de la Minería de Datos.

Se empleará Matplotlib para generar gráficos basados en los resultados de la Minería de Datos, que se mostrarán en el frontend. Además, se diseñarán esquemas básicos que definen la disposición de elementos en la interfaz, incluyendo botones, formularios, gráficos y paneles de control. La disposición de estos elementos se pueden observar en las Figuras 11, 12 y 13.

Para asegurar la usabilidad de la interfaz, se llevarán a cabo pruebas de usabilidad con usuarios representativos. Estos usuarios evaluarán aspectos como la navegación y la facilidad de uso, y proporcionarán comentarios sobre la experiencia general. Todos los comentarios recopilados durante las pruebas de usabilidad, que incluyen observaciones, sugerencias y problemas identificados por los usuarios, serán registrados y considerados en el proceso de diseño y desarrollo.

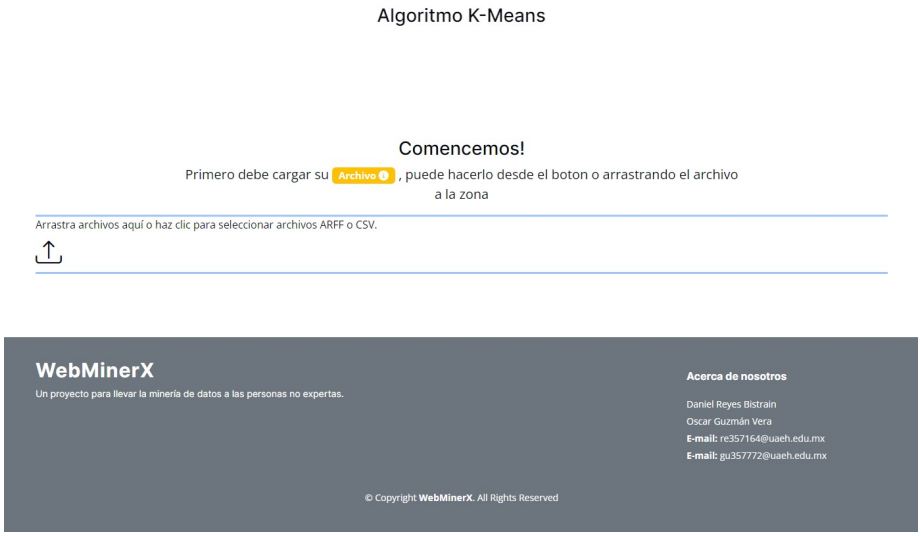

Figura 11 Bosquejo 1

Figura 12 Bosquejo 2

Figura 13 Bosquejo 3

SERVICIOS DE BACKEND

Para el desarrollo del backend de la aplicación, se ha decidido utilizar Python como lenguaje de programación. Este backend será responsable de cargar datos, ejecutar algoritmos de Minería de Datos y enviar los resultados al frontend. Además, se implementarán los algoritmos de validación y agrupamiento en Python, y estarán disponibles como funciones o módulos en el backend, permitiendo su llamada desde el frontend como se ve en la Figura 14.

En cuanto a la generación de visualización de resultados, se utilizará Matplotlib. Es importante destacar que no se empleará una base de datos en este escenario. Los datos cargados por los usuarios se mantendrán en la memoria durante la sesión de la aplicación. Por lo tanto, se gestionará la carga, el almacenamiento temporal y la eliminación segura de datos cargados para evitar problemas de rendimiento y uso excesivo de memoria. El diagrama de caso de uso presentado en la Figura 15 muestra este proceso.

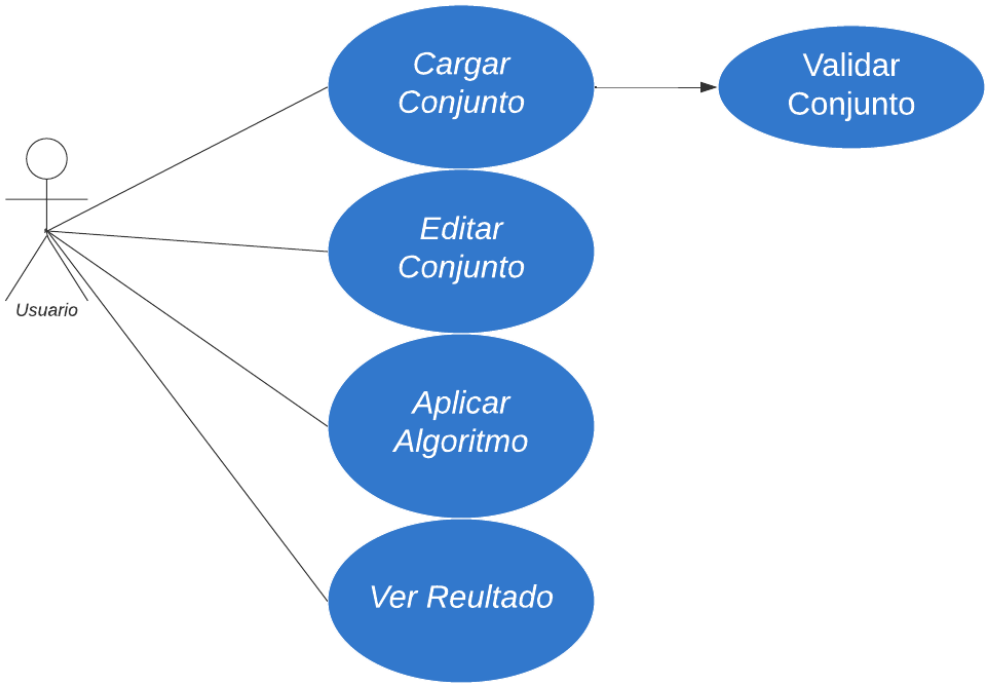

Figura 14 Utilización del archivo para aplicación de algoritmos

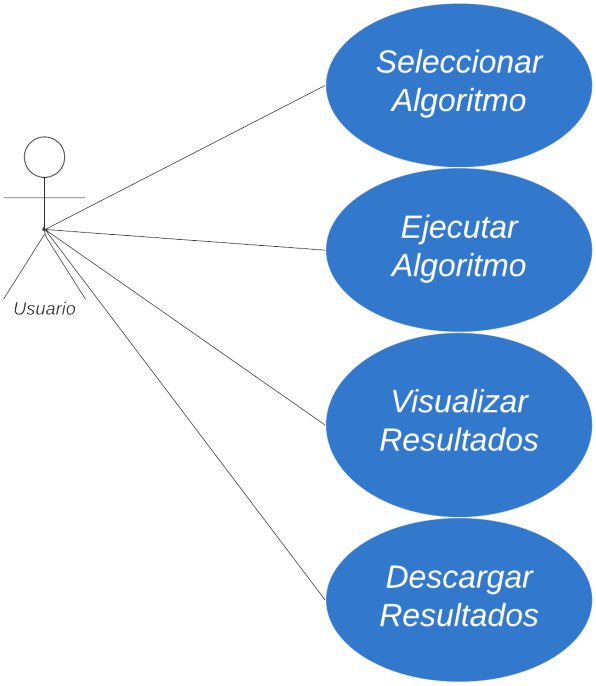

Figura 15 Visualización de resultados

El proyecto WebMinerX se desarrollará utilizando FastAPI como el framework principal para la aplicación web debido a su rendimiento y facilidad de uso. Además, Matplotlib se empleará para crear gráficos estáticos y visualizaciones detalladas cuando sea necesario, ya que proporciona un control completo sobre la apariencia de los gráficos

COMUNICACIÓN FRONTEND-BACKEND

En el proceso de comunicación entre el frontend y el backend, el frontend enviará solicitudes al backend para cargar datos, ejecutar algoritmos y obtener resultados. El backend, a su vez, se encargará de procesar estas solicitudes, realizará el procesamiento necesario y devolverá los resultados correspondientes al frontend. Esta interacción entre el frontend y el backend asegura el flujo de información y la ejecución de las tareas necesarias para el funcionamiento de la aplicación.

PARTE III

Codificación

La fase de codificación, que constituye una parte fundamental del desarrollo del proyecto, se llevará a cabo siguiendo las iteraciones previamente establecidas en la etapa de planificación.

Se utilizaron varias bibliotecas para agilizar y mejorar la eficiencia en las tareas realizadas. Entre estas herramientas se incluyen scikit-learn (sklearn), NumPy, Matplotlib y Dropzone.

I. Iteración 1 - Entrega Inicial

La versión inicial funcional de la aplicación ha sido codificada con éxito, permitiendo a los usuarios cargar datos y aplicar el algoritmo K-Means para llevar a cabo tareas de Minería de Datos. En este desarrollo, se ha incorporado la capacidad de cargar datos en formatos diversos, como CSV y ARFF, brindando flexibilidad en la entrada de información. Para cargar los datos se usaron las siguientes funciones

- cargar_tabla(): Esta función se encarga de cargar los datos del archivo en una tabla HTML. Analiza el contenido del archivo, detecta si es un archivo ARFF o CSV y luego crea una tabla con los datos. También gestiona la eliminación de columnas si no todas las filas tienen el mismo número de columnas.

```
archivo[1] = archivo[1].replaceAll(/\r/g, "");

% Verificar el tipo de botón y cargar contenido correspondiente
si $("#btn_centroides").data("tipo") es igual a 1 y archivo_nuevo
tiene longitud diferente de 0:
    contents = contenido de archivo_nuevo[1]
    $("#btn_centroides").data("tipo", 2)
sino:
    contents = contenido de archivo[1]
    $("#btn_centroides").data("tipo", 1)

% Dividir contenido en líneas
lines = contents.split("\n")

% Inicializar variables
colums_removed = falso
columnas = arreglo
options_columnas = '<option value="">Seleccione...</option>'
filas = arreglo
tabla = '<table class="table" id="file_table">'

% Procesar archivo ARFF
si la extensión de archivo[0] es igual a "arff":
    data = falso
    para cada línea en lines:
        valores = línea.split(" ")
        si valores[0] es igual a "@attribute":
            agregar valores[1] a columnas
        agregar '<option value="' + valores[1] + '">' + valores[1] + '</option>' a opt
        sino si valores[0] es igual a "@data":
            data = verdadero
        sino si data es verdadero:
            valores = línea.split(",")
            si longitud de valores es diferente de longitud de columnas:
```

```
            columns_removed = verdadero
        sino:
            agregar valores a filas

% Procesar archivo CSV
sino:
    columnas = longitud de lines[0] dividido por la coma
    para cada línea en lines:
        valores = línea.split(",")
        si longitud de valores es diferente de longitud de columnas:
            columns_removed = verdadero
        sino:
            agregar valores a filas
```

- format_data(): Esta función formatea los datos del archivo para su posterior procesamiento. Identifica si es un archivo ARFF o CSV y realiza las adecuaciones necesarias.

```
función format_data():
    // Inicializar variables
    contents = contenido de archivo[1]
    lines = dividir contents por saltos de línea
    columnas = arreglo vacío
    filas = arreglo vacío

    // Verificar el tipo de archivo
    si la extensión de archivo[0] es igual a "arff":
        data = falso
        // Procesar cada línea
        para cada línea en lines:
            valores = dividir línea por espacios
            // Verificar si es atributo, datos o fila de datos
            si valores[0] es igual a "@attribute":
                agregar valores[1] a columnas
            sino si valores[0] es igual a "@data":
                data = verdadero
            sino si data es verdadero:
                // Procesar fila de datos
                valores = dividir línea por comas
          // Verificar si el número de valores coincide con el número de columnas
```

```
                si longitud de valores es igual a longitud de columnas:
                    // Eliminar retornos de carro de cada valor
                    para cada valor en valores:
                reemplazar en valor los retornos de carro por cadena vacía
                    agregar valores a filas
        sino:
         columnas_num = longitud de la primera línea dividida por comas
            // Crear nombres de columnas por defecto
            para cada índice de 1 hasta columnas_num:
                agregar "Columna " + (índice + 1) a columnas
            // Procesar cada línea de datos
            para cada línea en lines:
                valores = dividir línea por comas
            // Verificar si el número de valores coincide con el número de columnas
                si longitud de valores es igual a columnas_num:
                    // Eliminar retornos de carro de cada valor
                    para cada valor en valores:
                   reemplazar en valor los retornos de carro por cadena vacía
                    agregar valores a filas

        // Devolver filas y columnas como resultado
        devolver [filas, columnas]
```

Para el algoritmo kmeans se realizo lo siguiente

- Validación de Formulario: Comprueba si el formulario es válido antes de continuar. Esto asegura que los parámetros necesarios estén configurados correctamente antes de enviar la petición al servidor.

- Obtención de Datos Formateados: Llama a la función `format_data()` para obtener los datos del formulario en un formato adecuado para enviar al servidor.

- Construcción de la Petición: Crea un objeto que contiene los datos basicos para la petición al servidor, incluyendo el número de clusters, el número máximo de iteraciones y si se debe usar un estado aleatorio.

- Envío de la Petición al Servidor: Utiliza `fetch()` para enviar una petición POST al servidor. El cuerpo de la petición contiene los datos formateados en formato JSON.

- Obtención de la Respuesta del Servidor: Espera la respuesta del servidor utilizando `await`. Cuando se recibe la respuesta, se convierte a formato JSON utilizando `response.json()`.

- Procesamiento de la Respuesta: La respuesta del servidor se guarda en la variable `data`. Se extraen los datos necesarios de esta respuesta, como los gráficos y los centros de los clusters. Luego se actualizan los elementos HTML en la página para mostrar estos datos.

- Modificación de Archivos: Si el archivo cargado es de tipo ARFF, se modifica para incluir una nueva línea que identifique el cluster al que pertenece cada instancia.

- Actualización de la Interfaz de Usuario: Se actualiza la tabla de datos en la interfaz de usuario para reflejar los cambios realizados en los archivos y se muestran los nuevos botones para descargar los centroides de los clusters.

```
si $("#parametros_form")[0].checkValidity():
    datos = llamar a format_data()
    peticion = objeto vacío
    peticion.clusters = valor de $("#num_cluster")
    peticion.iteraciones = valor de $("#max_iter")
  peticion.random_state = valor de $("#random_state") está marcado
    peticion.datos = datos[0]
  respuesta = esperar respuesta de fetch(document.location.origin + ":8000/kmeans", {
        método: "POST",
        cabeceras: {
           "Content-Type": "application/json",
        },
        cuerpo: convertir peticion a JSON
    })
    data = esperar respuesta.json()
    respuesta = JSON.parse(data)
```

II. Iteración 2 - Mejora de la Funcionalidad de K-Means

Programación de la aplicación con el objetivo específico de refinar y mejorar la funcionalidad del algoritmo K-Means. En esta fase de desarrollo, se ha otorgado a los usuarios la capacidad de ajustar parámetros, permitiéndoles personalizar la ejecución del algoritmo de acuerdo con sus necesidades específicas.

- Mejora en la construcción de la Petición: Crea un objeto `peticion` que contiene los datos necesarios para la petición al servidor, incluyendo el número de clusters, el número máximo de iteraciones y si se debe usar un estado aleatorio.

III. Iteración 3 - Incorporación de DBSCAN

Programar la funcionalidad de DBSCAN como otro algoritmo de Minería de Datos disponible para los usuarios dentro de la aplicación. Esto implica la completa implementación del algoritmo DBSCAN, proporcionando a los usuarios una interfaz intuitiva para seleccionar y configurar los parámetros específicos de DBSCAN según sus necesidades.

- El algoritmo DBSCAN sigue una serie de pasos similares al algoritmo K-Means en términos de procesamiento y visualización de datos. Sin embargo, la distinción clave radica en cómo se construye la petición para enviar al servidor, ya que son algoritmos diferentes con requisitos distintos.

  Similar a K-Means, el proceso de DBSCAN comienza con la validación del formulario y la obtención de datos formateados, seguido por la construcción de la petición para enviar al servidor. A diferencia de K-Means, DBSCAN necesita parámetros específicos, como el número mínimo de clusters y el valor de epsilon, que determinan la densidad y la distancia mínima entre puntos para considerarlos vecinos.

  ```
  si $("#parametros_form")[0].checkValidity():
      datos = llamar a format_data()
      peticion = objeto vacío
      peticion.min_cluster = valor de $("#min_cluster")
      peticion.eps = valor de $("#eps")
      peticion.datos = datos[0]
  ```

```
respuesta = esperar respuesta de fetch(document.location.origin + ":8000/dbscan", {
    método: "POST",
    cabeceras: {
       "Content-Type": "application/json",
    },
    cuerpo: convertir peticion a JSON
 })
 data = esperar respuesta.json()
 respuesta = JSON.parse(data)
```

IV. Iteración 4 - Validación de Clústeres

Codificar los algoritmos de validación, incluyendo Silhouette, Calinski y Davies-Bouldin, que permitirán evaluar la calidad de los clústeres generados por los algoritmos de Minería de Datos.

- Validación de Formulario: Verifica si el formulario es válido antes de continuar. Esto asegura que los parámetros necesarios estén configurados correctamente antes de enviar la petición al servidor.
- Obtención de Datos Formateados: Llama a la función `format_data()` para obtener los datos del formulario en un formato adecuado para enviar al servidor.
- Construcción de la Petición: Crea un objeto `peticion` que contiene los datos necesarios para la petición al servidor. En este caso, solo se incluyen los datos formateados obtenidos del dataset.
- Envío de la Petición al Servidor: Utiliza `fetch()` para enviar una petición POST al servidor.
- Obtención de la Respuesta del Servidor: Espera la respuesta del servidor utilizando `await`. Cuando se recibe la respuesta, se extrae el valor de la métrica Silhouette, Calinski-Harabasz o Davies-Bouldin de la respuesta y se actualiza el elemento HTML con el id `#metric` con este valor.
- Actualización de la Interfaz de Usuario: Se muestra el elemento HTML.

```
si $("#parametros_form")[o].checkValidity():
   datos = llamar a format_data()
   peticion = objeto vacío
   peticion.datos = datos[o]
```

```
respuesta = esperar respuesta de fetch(document.location.origin + ":8000/calinski,s
    método: "POST",
    cabeceras: {
       "Content-Type": "application/json",
    },
    cuerpo: convertir peticion a JSON
 })
 $("#metric").html(JSON.parse(esperar respuesta.json()).score)
 $("#centros_card").attr("hidden", false)
```

V. Iteración 5 - Incorporación de Jerárquico

Finalmente integrar el algoritmo jerárquico. Completa implementación del algoritmo de clustering jerárquico.

- Esta función realiza tareas similares a las funciones anteriores, pero en este caso, realiza un algoritmo de clustering jerárquico en lugar de KMeans o DBSCAN, y muestra los resultados correspondientes en la interfaz de usuario.

```
Si $("#parametros_form")[0].checkValidity() es verdadero:
datos = format_data()
peticion = {}
peticion.clusters = $("#num_cluster").val()
peticion.datos = datos[0]
response = await fetch(document.location.origin + ":8000/jerarquico", {
method: "POST",
headers: {
"Content-Type": "application/json",
},
body: JSON.stringify(peticion),
})
data = await response.json()
respuesta = JSON.parse(data)
```

VI. Iteración 6 - Mejoras de Usabilidad y Exportación

Realizar mejoras significativas en la experiencia del usuario, enfocándose en una interfaz más intuitiva y funcional. Estas mejoras incluirán ajustes en la interfaz de usuario para garantizar una experiencia más intuitiva y amigable. Además, se implementará una función que permita a los usuarios exportar resultados.

- botonera(): Esta función genera la interfaz de usuario para las opciones de conjunto, como eliminar columnas y caracteres especiales.
- comillas() y caracter(): Estas funciones se utilizan para eliminar comillas y caracteres especiales de las líneas del archivo.
- borrar_columna(): Elimina una columna seleccionada por el usuario de la tabla de datos.
- descargar_kmeans(), descargar_jerarquico(), descargar_dbscan(): Estas funciones permiten la descarga de datos procesados en diferentes algoritmos de clustering en formato de archivo especificado.

Para mayor información sobre cómo funciona la aplicación web, consultar este tutorial que explica su funcionamiento de manera detallada: `https://youtu.be/CJABCtJcgQw` o buscarlo en la plataforma de youtube como **WebMinerX-Licenciatura en Ciencias Computaciones-Universidad Autónoma del Estado de Hidalgo**. Este vídeo tiene la intención de guiar a través de los distintos aspectos y funcionalidades de la aplicación, desde la carga de archivos hasta la interpretación de resultados.

PARTE IV

Pruebas

Se llevaron a cabo pruebas integrales de usabilidad en la aplicación WebMinerX, involucrando a participantes que no poseían experiencia previa en Minería de Datos. Estas pruebas se realizaron con el objetivo de evaluar la accesibilidad y facilidad de uso de la aplicación por parte de un público diverso.

Capítulo 4

Resultados

A continuacion se presentan los resultados alcanzados tras el desarrollo del proyecto. Se describen las iteraciones antes planteadas en la etapa de planificación y realizadas en la codificación.

I. Iteración 1 - Entrega Inicial

Versión inicial funcional de la aplicación con la capacidad de permitir a los usuarios cargar datos y aplicar el algoritmo K-Means para llevar a cabo tareas de Minería de Datos. Esto incluye la capacidad de cargar datos en formatos como CSV y ARFF. Además, la aplicación debe es capaz de ejecutar el algoritmo K-Means sobre los datos proporcionados. Ilustrado en la Figura 16

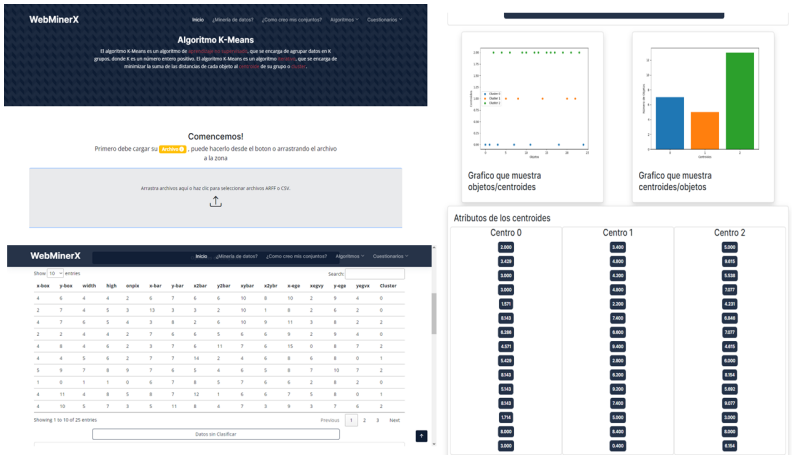

Figura 16 Actividades realizadas en la iteración 1

II. Iteración 2 - Mejora de la Funcionalidad de K-Means

Refinar y mejorar la funcionalidad del algoritmo K-Means, capacidad de ajustar parámetros y explorar los resultados de manera más detallada. Respresentado en la figura 17

Figura 17 Actividades realizadas en la iteración 2

III. Iteración 3 - Incorporación de DBSCAN

Incorporar la funcionalidad de DBSCAN. Proporcionando a los usuarios una interfaz intuitiva para seleccionar y configurar los parámetros específicos de DBSCAN según sus necesidades como en la figura 18.

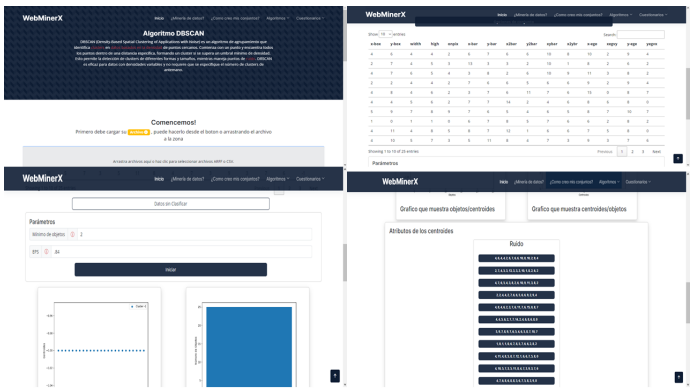

Figura 18 Actividades realizadas en la iteración 3

IV. Iteración 4 - Validación de Clústeres

Implementación de algoritmos de validación, incluyendo Silhouette, Calinski y Davies-Bouldin, que permitirán evaluar la calidad de los clústeres generados por los algoritmos de Minería de Datos. La aplicación también proporcionará una visualización clara y concisa de estas métricas de validación, presentándolas junto a los resultados ilustrado en la figura 19.

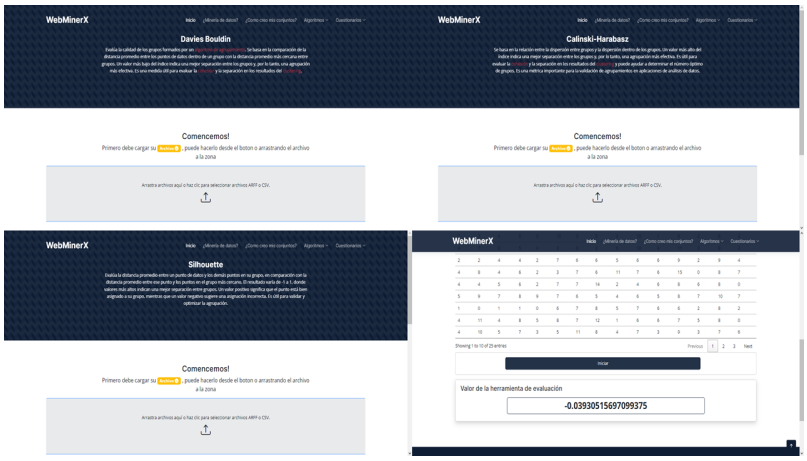

Figura 19 Actividades realizadas en la iteración 4

V. Iteración 5 - Incorporación de Jerárquico

Integrar el algoritmo de clustering jerárquico para ampliar las opciones de algoritmos disponibles para los usuarios dentro de la aplicación. Completa implementación del algoritmo de clustering jerárquico, proporcionando a los usuarios una interfaz intuitiva para seleccionar y configurar los parámetros específicos de este algoritmo como en la figura 24.

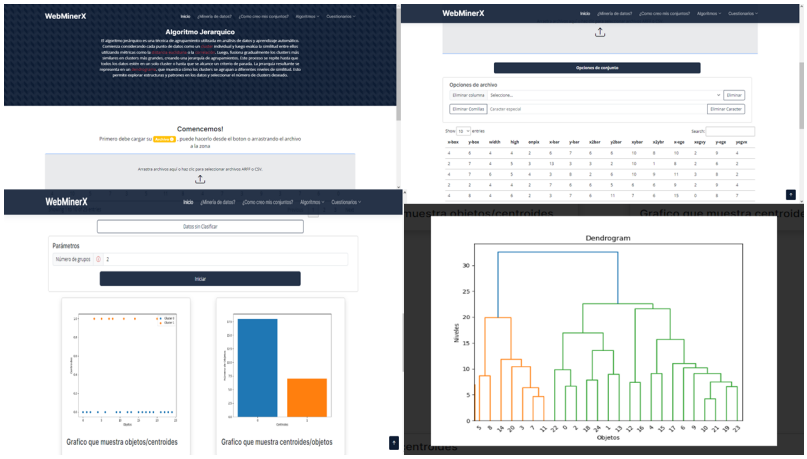

Figura 20 Actividades realizadas en la iteración 5

VI. Iteración 6 - Mejoras de Usabilidad y Exportación

Realizar mejoras significativas en la experiencia del usuario, enfocándose en una interfaz más intuitiva y funcional. Estas mejoras incluirán ajustes en la interfaz de usuario para garantizar una experiencia más intuitiva y amigable. Además, se implementará una función que permita a los usuarios exportar resultados como se ve en la figura 21.

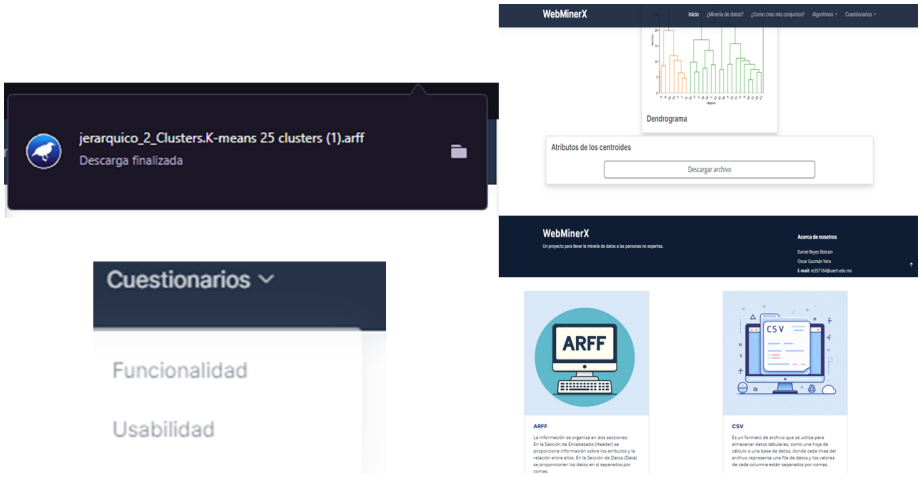

Figura 21 Actividades realizadas en la iteración 6

VII. Pruebas del sistema

Se llevaron a cabo diversas pruebas de los algoritmos con el objetivo de revisar su funcionamiento y garantizar su eficacia en la realización de tareas. Estas pruebas fueron escenarios simulados, proporcionando así un para evaluar el rendimiento de cada algoritmo. Los resultados obtenidos fueron analizados para comprobar si los algoritmos ejecutaban las tareas de manera precisa y coherente, asegurando así su correcto desempeño en diversas situaciones y entornos. En las siguientes Figuras se muestran algunos de nuestros resultados

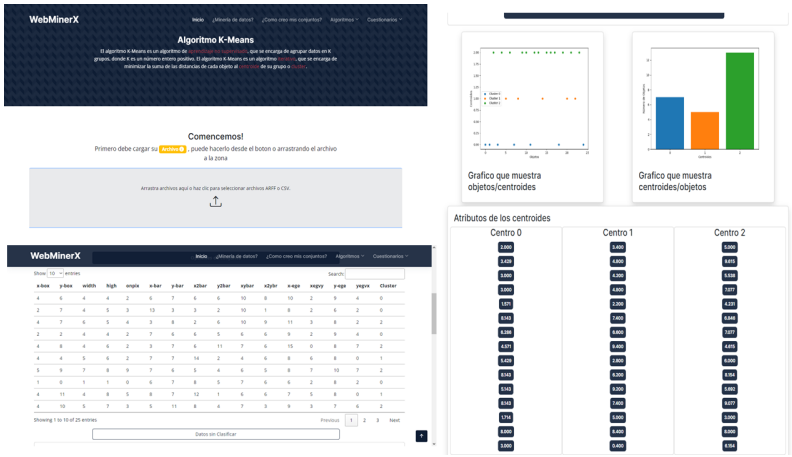

Figura 22 Algoritmo K-means

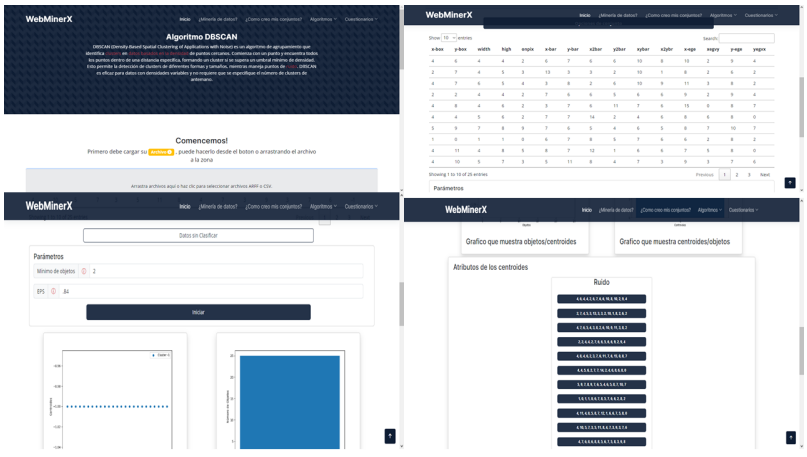

Figura 23 Algoritmo DBScan

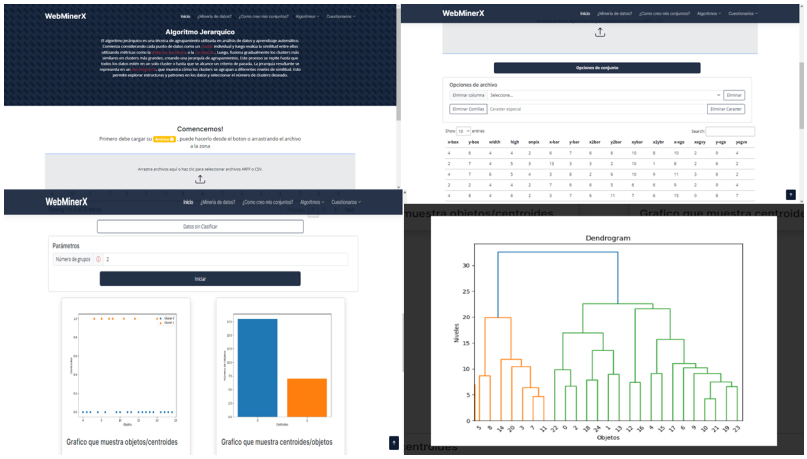

Figura 24 Algoritmo Jerárquico

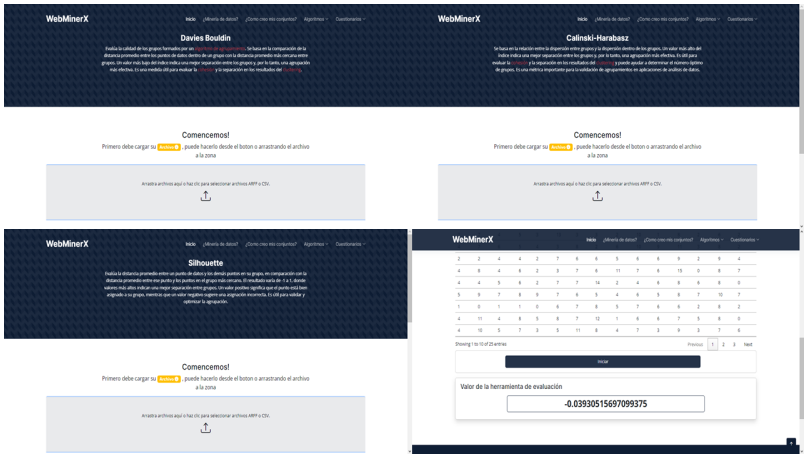

Figura 25 Algoritmos de Validación

VIII. Pruebas de Usabilidad

Se llevaron a cabo una serie de pruebas que permiten identificar problemas de diseño de la interfaz de usuario y a la vez, ayudan a asegurar que la aplicación sea fácil de usar. Se proporcionó la herramienta a un grupo de 20 usuarios que tuvieron la oportunidad de interactuar con la aplicación. Una vez que terminaron de usar la herramienta, se les dió a responder un cuestionario de 10 preguntas con las cuales se evaluará el aspecto de usabilidad.

Para evaluar la usabilidad, se empleó la conocida escala de Likert durante las pruebas del sistema. Los usuarios proporcionaron su retroalimentación sobre diversos aspectos de la interfaz, desde la facilidad de uso hasta la claridad de las instrucciones. Estas respuestas fueron fundamentales para comprender la experiencia del usuario y guiar los ajustes necesarios para mejorar la accesibilidad y la eficacia del sistema para una amplia gama de usuarios.

Para evaluar la respuesta de cada pregunta se ocupó la siguiente escala:

1. Totalmente en desacuerdo.
2. En desacuerdo.
3. Neutro.
4. De acuerdo.
5. Totalmente de acuerdo.

Los resultados obtenidos en cada pregunta reflejan los siguientes resultados:

1. Creo que me gustaría utilizar este sistema con frecuencia.

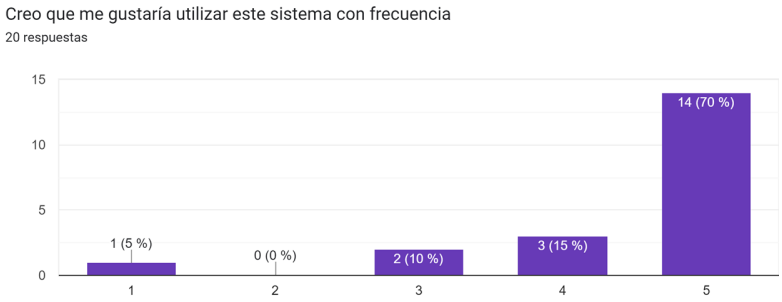

Figura 26 Resultados Pregunta 1 Usabilidad

2. Encontré el sistema innecesariamente complejo.

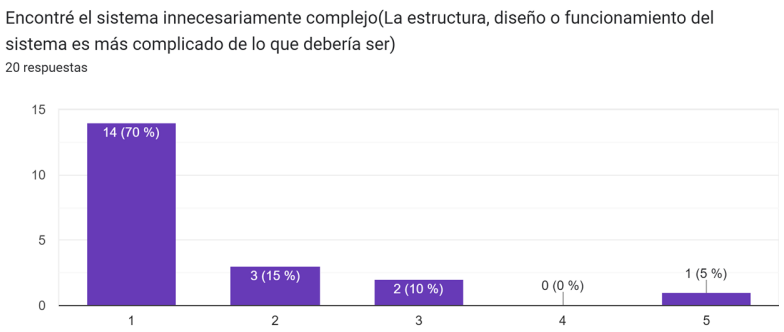

Figura 27 Resultados Pregunta 2 Usabilidad

3. Pensé que el sistema era fácil de usar.

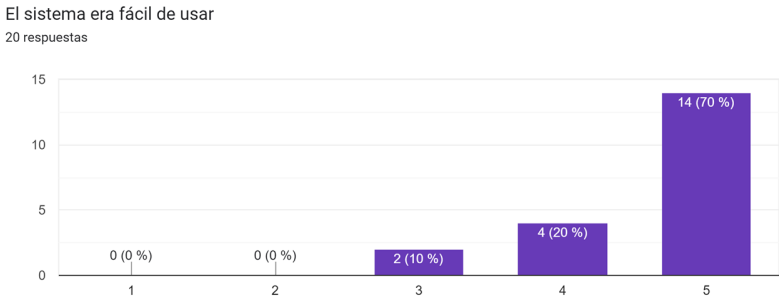

Figura 28 Resultados Pregunta 3 Usabilidad

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.

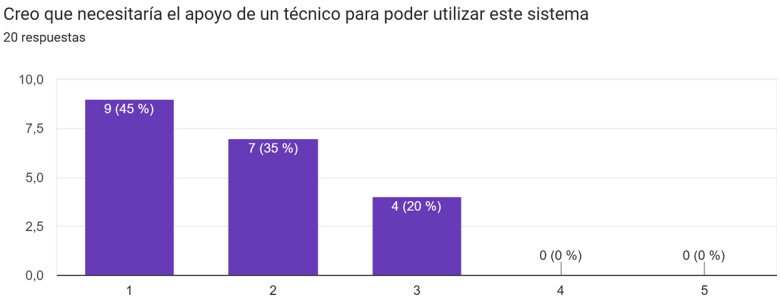

Figura 29 Resultados Pregunta 4 Usabilidad

5. Encontré que las diversas funciones de este sistema estaban bien integradas.

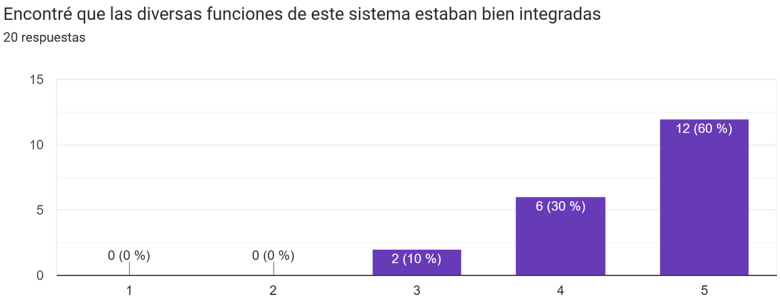

Figura 30 Resultados Pregunta 5 Usabilidad

6. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.

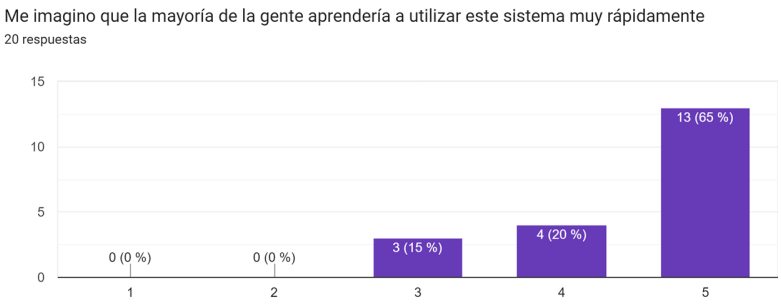

Figura 31 Resultados Pregunta 6 Usabilidad

7. Encontré el sistema muy complicado de usar.

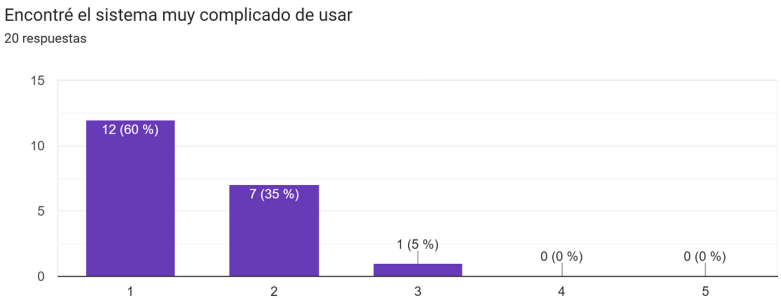

Figura 32 Resultados Pregunta 7 Usabilidad

8. Me sentí muy seguro usando el sistema.

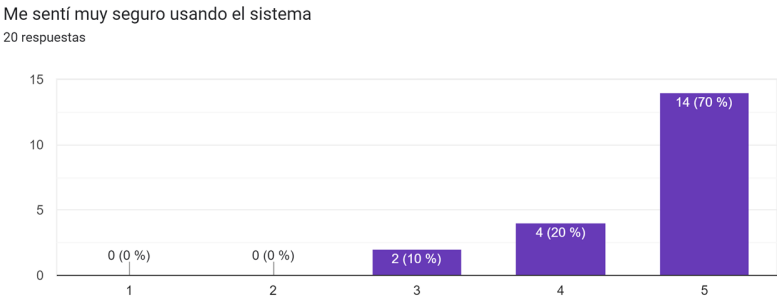

Figura 33 Resultados Pregunta 8 Usabilidad

9. Necesitaba aprender muchas cosas antes de empezar con este sistema.

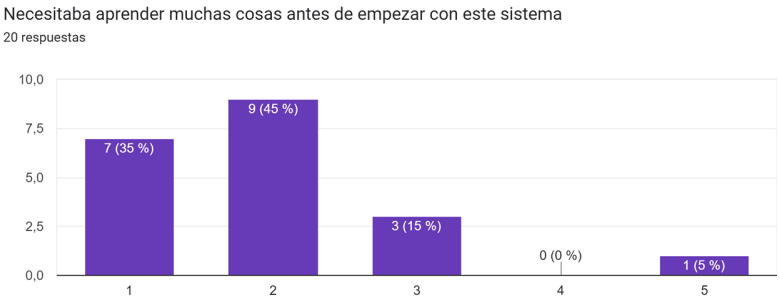

Figura 34 Resultados Pregunta 9 Usabilidad

10. ¿El programa presenta una interfaz intuitiva y fácil de usar que facilita la exploración y comprensión de los datos?

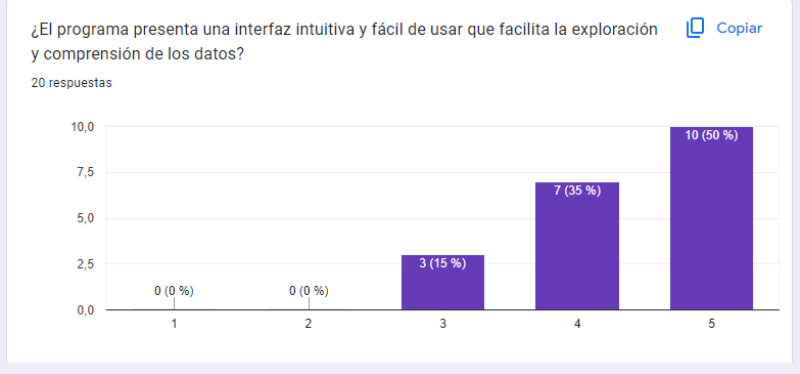

Figura 35 Resultados Pregunta 10 Usabilidad

Después de recopilar las respuestas de los participantes de la serie de 10 preguntas utilizando la escala de Likert, se calculó el puntaje de cada pregunta sumando los valores asignados a las respuestas de cada participante. Una vez obtenidos los puntajes individuales para cada pregunta, se obtuvo el promedio de estos puntajes para cada ítem. Luego de realizar este análisis, se descubrió que el promedio de los puntajes de las preguntas fue de 4.43 sobre un total posible de 5. Este resultado sugiere una evaluación generalmente positiva por parte de los participantes en relación con los aspectos medidos por las preguntas de la encuesta.

Conclusiones

La Minería de Datos es una herramienta poderosa para el reconocimiento de patrones en grandes conjuntos de datos. Esta disciplina permite extraer información útil y relevante de forma automática, utilizando técnicas como el aprendizaje automático, la estadística, la lógica y la optimización. Además, se puede aplicar a diversos dominios y problemas, como la detección de fraudes, la segmentación de clientes, el diagnóstico médico, la clasificación de imágenes, el análisis de redes sociales y muchos más. Las ventajas que ofrece son competitivas y estratégicas a las organizaciones que la utilizan, así como a las nuevas investigaciones, ya que les permite mejorar la toma de decisiones, la innovación y la eficiencia.

El uso de diversos algoritmos para el procesamiento de grandes conjuntos de información, hace que sea necesario contar con herramientas que permitan su aplicación. Sin embargo, a pesar de que existen diversas plataformas, usuarios no expertos en la materia, pudieran no entender la forma en como se usan, ya que no son del todo descriptivas. Es por esta razón, que se propone el desarrollo de una herramienta que guie al usuario en la aplicación de algunos de los algoritmos más utilizados de Minería de Datos.

La creación de una aplicación web impulsada por Python, diseñada para facilitar la Minería de Datos a usuarios no expertos, surge como una respuesta necesaria y valiosa en el actual panorama tecnológico. La existencia de barreras de entrada, como la necesidad de conocimientos avanzados en programación o estadísticas, ha limitado el acceso de muchos a las técnicas de análisis de datos más avanzadas. La ausencia de una herramienta que aborde específicamente esta problemática crea la oportunidad para una solución innovadora.

La nueva herramienta propuesta no solo simplificará el proceso de Minería de Datos, sino que también guiará a los usuarios en la selección de las técnicas más adecuadas para sus conjuntos de datos y objetivos específicos. La interfaz intuitiva y amigable permitirá la carga de datos, la aplicación de algoritmos de agrupamiento y validación, y la visualización clara de resultados. La presentación de índices de validación de manera comprensible contextualizará la información, empoderando a los usuarios no expertos para tomar decisiones informadas sobre la técnica más efectiva para sus necesidades.

Esta herramienta no solo es una solución técnica, sino también una puerta de acceso para una amplia variedad de sectores que pueden beneficiarse de la toma de decisiones basada en datos. Al eliminar las barreras técnicas y de conocimiento, se anticipa que esta aplicación desempeñará un papel fundamental al proporcionar a las personas y organizaciones las herramientas necesarias para aprovechar al máximo el valor oculto en sus datos. Su potencial innovador radica en su capacidad para democratizar la Minería de Datos y transformarla en una herramienta accesible para todos.

Logros alcanzados

La aplicación desarrollada tiene como objetivo principal la creación de una plataforma web intuitiva y atractiva que facilite la Minería de Datos para usuarios no expertos.

La base de esta aplicación es una interfaz de usuario cuidadosamente diseñada, que permite a los usuarios acceder de manera sencilla. Una vez dentro, el usuario tiene la capacidad de descargar sus conjuntos de datos en diversos formatos, como CSV y Excel, lo que simplificará la preparación de los datos para su análisis.

Una característica central es la implementación de algoritmos de Minería de Datos, en particular de las tareas de agrupamiento y validación interna de los modelos generados por estos algoritmos. Se integraron las técnicas K-Means, Agrupamiento Jerárquico y DBScan por parte de los algoritmos de agrupamiento, los cuales permiten segmentar los datos en grupos significativos. Además, se integraron algoritmos de validación, Davies Bouldin, Calinski Harabaz y Silhoutte, para evaluar el rendimiento de los modelos resultantes y tomar decisiones informadas.

La visualización interactiva de los resultados juega un papel fundamental, ya que los usuarios pueden explorar y comprender fácilmente los grupos y patrones identificados por los algoritmos. Se desarrollaron gráficos y representaciones visuales para presentar resultados de validación y comparación de algoritmos, simplificando la interpretación de los análisis.

La exploración de datos es otra característica clave de la aplicación, brindando a los usuarios herramientas para comprender sus conjuntos de datos antes de aplicar los algoritmos. Análisis descriptivos básicos y estadísticas clave se presentan de manera visual y accesible.

Esta plataforma esta diseñada para su acceso desde cualquier lugar, asegurando una experiencia fluida y responsiva tanto en dispositivos móviles como de escritorio. Los usuarios podrán aprovechar la funcionalidad completa sin importar dónde se encuentren.

El desarrollo de la aplicación incluye pruebas en una variedad de conjuntos de datos para garantizar su funcionalidad y precisión. Una fase crítica de evaluación implicó la obtención de comentarios y retroalimentación de usuarios no expertos. Lo que permitió ajustar la aplicación según sus necesidades y mejorar su usabilidad.

Trabajos futuros

La tecnologia siempre esta en constante cambio y los datos están volviéndose más complicados. Por eso, necesitamos seguir mejorando las herramientas que se utilizan para analizar estos datos. Los trabajos futuros que se proponen son muy importantes porque ayudarán a mantener este proyecto actualizado y útil. Esto asegurará que la herramienta siga siendo relevante y en algún momento pueda competir con otras herramientas similares que están disponibles.

1. **Integración de una Base de Datos:** Agregar soporte para bases de datos, como SQLite o MySQL, para permitir a los usuarios almacenar conjuntos de datos cargados previamente, resultados de análisis y configuraciones personalizadas.

2. **Mejora de la Interfaz de Usuario (UI):** Realizar ajustes en la interfaz de usuario para hacerla más intuitiva y fácil de usar. Explorar diseños de interfaz más modernos y atractivos utilizando técnicas avanzadas de CSS y bibliotecas de componentes de interfaz de usuario.

3. **Optimización del Rendimiento:** Optimizar el rendimiento de la aplicación para manejar grandes conjuntos de datos de manera eficiente y rápida. Implementar técnicas de carga asíncrona para mejorar los tiempos de carga y la capacidad de respuesta de la página.

4. **Añadir Más Algoritmos y Métricas:** Investigar e incorporar otros algoritmos de agrupamiento y validación de datos disponibles en bibliotecas como scikit-learn y otras. Ampliar la gama de métricas de evaluación disponibles para proporcionar a los usuarios una evaluación más completa de los resultados de sus análisis.

5. **Implementar Funcionalidades Colaborativas:** Agregar funciones de colaboración en tiempo real que permitan a múltiples usuarios trabajar juntos en un mismo análisis. Incorporar herramientas de comentarios y anotaciones para facilitar la comunicación y la colaboración entre usuarios.

6. **Añadir Capacidades de Exportación:** Permitir a los usuarios exportar los resultados de sus análisis en diferentes formatos, como CSV, Excel o PDF, para facilitar su uso posterior. Implementar opciones de personalización para que los usuarios puedan seleccionar qué resultados desean exportar y en qué formato.

Bibliografía

[1] Modelo de espiral. desarrollo de sistemas. ciclo de vida. desarrollo de software. modelo conceptual., ángulo, texto, monocromo png | pngwing, 2023. Pngwing.com.

[2] Admin_Donetonic. Los 5 eventos en scrum claves para el desarrollo de producto, May 26 2021. DoneTonic.

[3] Estrategias de Trading https://estrategiastrading.com/clustering jerarquico/. Clustering jerárquico - agrupar elementos con minería de datos, July 12 2019.

[4] Universidad Politécnica de Madrid Departamento de Matemática Aplicada. Introducción al aprendizaje automático. `https://dcain.etsin.upm.es/~carlos/bookAA/introAA.html`, 2021.

[5] Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley, 5th edition, 2011.

[6] C. García. Implementación del algoritmo de los kvecinos más cercanos (k-nn) y estimación del mejor valor local de kpara su cálculo, 2012.

[7] Cristina García Cambronero and Irene Gómez Moreno. Algoritmos de aprendizaje: KNN & KMEANS. *Barranco blog*, 2012.

[8] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 3rd edition, 2012.

[9] IONOS. El modelo en cascada: desarrollo secuencial de software, March 21 2019. IONOS Digital Guide; IONOS.

[10] Elizabeth León Guzmán. Métricas para la validación de clustering. 2016.

[11] J. Montero. Algoritmos de agrupamiento basados en densidad y validación de clusters], 2016.

[12] J. Montero. Algoritmos de agrupamiento basados en densidad y validación de clusters, 2016.

[13] José Ignacio Montero Núñez. Algoritmos de agrupamiento basados en densidad y validación de clusters, 2016.

[14] Pastrán. Indice ch algoritmo de selecciÓn y validaciÓn del mÉtodo de clusterizaciÓn Óptimo para datos no supervisados, 2021.

[15] Luisa Fernanda Pastrán Ramírez and Santiago Gongora Aya. Algoritmo de selecciÓn y validaciÓn del mÉtodo de clusterizaciÓn Óptimo para datos no supervisados, 2021.

[16] Julio Roche. ¿qué es el desarrollo en espiral?, April 8 2020. Deloitte Spain.

[17] Ian Sommerville. *Ingeniería de Software*. Addison-Wesley, Reading, MA, USA, 9th edition, 2011.