



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

**IMPLEMENTACIÓN DE UNA APLICACIÓN
WEB PARA EL APRENDIZAJE DE
MACHINE LEARNING**

Oriol Villanova Medina

Director: Eloi Puertas
Realizado en: Departamento de
Matemáticas e
Informática

Barcelona, 17 de enero de 2023

RESUM

En la societat actual, l'aprenentatge automàtic (ML) juga un paper vital, però comprendre-ho pot ser desafiador. AprendreML aborda això amb una plataforma web interactiva que fa que el coneixement de ML sigui accessible per a més persones.

Motivat per desafiaments personals i acadèmics, AprendreML no és només una plataforma d'e-learning, sinó una experiència que simplifica la comprensió d'algorismes de ML. El projecte busca establir una base sòlida per a entendre ML.

En el seu desenvolupament, AprendreML crea amb èxit una aplicació web completa, destacant un desenvolupament escalable i una interfície fàcil d'entendre. Aquest projecte reflecteix el compromís de fer que el coneixement de ML sigui accessible, contribuint al panorama en evolució de l'educació en línia.

La recerca presenta els resultats d'un projecte per a dissenyar i implementar una aplicació web d'educació en ML. Inclou la creació d'algorismes i la gestió de dades. L'objectiu és proporcionar una interfície que faciliti la comprensió de ML, especialment per a estudiants.

La implementació utilitza eines per a millorar la portabilitat i eliminar problemes. La recerca ofereix una visió de l'execució exitosa de AprendreML, contribuint al camp de l'educació en línia i l'accessibilitat a l'aprenentatge automàtic. En treballs futurs es podrien explorar característiques i millores addicionals per a l'experiència en AprendreML.

El projecte ha aconseguit amb èxit la seva meta principal de crear una aplicació web des de zero, abordant tot, des de la planificació fins al llançament final. L'assoliment de pràcticament tots els objectius i requisits inicials destaca l'èxit del projecte.

RESUMEN

En la sociedad actual, el aprendizaje automático (ML) juega un papel vital, pero comprenderlo puede ser desafiante. AprendeML aborda esto con una plataforma web interactiva que hace que el conocimiento de ML sea accesible para más personas.

Motivado por desafíos personales y académicos, AprendeML no es solo una plataforma de e-learning, sino una experiencia que simplifica la comprensión de algoritmos de ML. El proyecto busca establecer una base sólida para entender ML.

En su desarrollo, AprendeML crea con éxito una aplicación web completa, destacando un desarrollo escalable y una interfaz fácil de entender. Este proyecto refleja el compromiso de hacer que el conocimiento de ML sea accesible, contribuyendo al panorama en evolución de la educación en línea.

La investigación presenta los resultados de un proyecto para diseñar e implementar una aplicación web de educación en ML. Incluye la creación de algoritmos y la gestión de datos. El objetivo es proporcionar una interfaz que facilite la comprensión de ML, especialmente para estudiantes.

La implementación utiliza herramientas para mejorar la portabilidad y eliminar problemas. La investigación ofrece una visión de la ejecución exitosa de AprendeML, contribuyendo al campo de la educación en línea y la accesibilidad al aprendizaje automático. En trabajos futuros se podrían explorar características y mejoras adicionales para la experiencia en AprendeML.

El proyecto ha alcanzado con éxito su meta principal de crear una aplicación web desde cero, abordando todo, desde la planificación hasta el lanzamiento final. El logro de prácticamente todos los objetivos y requisitos iniciales destaca el éxito del proyecto.

ABSTRACT

In today's society, machine learning (ML) plays a vital role, but understanding it can be challenging. AprendeML addresses this with an interactive web platform that makes ML knowledge accessible to more people.

Motivated by personal and academic challenges, AprendeML is not just an e-learning platform, but an experience that simplifies the understanding of ML algorithms. The project seeks to establish a solid base to understand ML.

In its development, AprendeML successfully creates a complete web application, highlighting scalable development and an easy-to-understand interface. This project reflects the commitment to make ML knowledge accessible, contributing to the evolving panorama of online education.

The research presents the results of a project to design and implement a web education application in ML. It includes algorithm creation and data management. The goal is to provide an interface that facilitates understanding of ML, especially for students.

Implementation uses tools to improve portability and eliminate problems. The research offers a vision of the successful execution of AprendeML, contributing to the field of online education and accessibility to machine learning. In future work I could explore additional features and improvements for the AprendeML experience.

The project has successfully achieved its main goal of creating a web application from scratch, addressing everything, from planning to final launch. The achievement of almost all of the initial objectives and requirements highlights the success of the project.

ÍNDICE

Resum	2
Resumen	3
Abstract	4
1. Introducción	7
1.1 Motivación	8
1.2 Objetivos	9
1.2.1 Objetivos Generales.....	9
1.2.2 Objetivos Secundarios	9
1.2.3 Objetivos Técnicos.....	9
1.3 Planificación	10
1.3.1 Diagrama de Gantt.....	10
1.3.2 Contenido de la memoria	11
2. Estado del arte	13
3. Análisis	15
3.1 Algoritmos	15
3.1.1 Algoritmos Seleccionados	15
3.2 Análisis del Contexto.....	24
3.2.1 Identificación de Requisitos.....	25
3.2.2 Backlog	28

4. Diseño	32
4.1 Tecnologías.....	32
4.1.1 Servidor.....	32
4.1.2 Interfaz gráfica	34
4.2 Arquitectura	36
4.2.1 Interfaz gráfica	37
4.2.2 Servidor.....	38
4.3 Wireframes	39
5. Implementación	42
5.1 Base de datos	42
5.2 Algoritmos	43
5.3 API.....	44
5.4 Interfaz gráfica.....	44
5.5 Subida a producción.....	45
6. Conclusiones	52
7. Prueba de resultados.....	48
8. Posibles mejoras	¡Error! Marcador no definido.
9. Bibliografía.....	54

1. INTRODUCCIÓN

Actualmente es innegable que el “*machine learning*” o aprendizaje automático se ha convertido en una herramienta fundamental dentro de la estructura de nuestra sociedad. A pesar del interés que despierta este campo, resulta sorprendente cuántas personas, incluso con sólidos conocimientos en informática, desconocen todos sus detalles y potencialidades. Al adentrarnos en la exploración de los diversos tipos de aprendizaje automático y sus algoritmos, nos encontramos con una gran cantidad de información. Aunque esta información es completa y rica en contenido, se presenta un desafío significativo: la dificultad para comprenderla en su totalidad debido a la falta de referencias interactivas.

La comprensión de los algoritmos de aprendizaje automático requiere más que una asimilación de teoría, exige una interacción, la capacidad de visualizar la dinámica entre los algoritmos y las variables de entrada, así como la capacidad de observar cómo los resultados cambian en respuesta a la manipulación de diversos parámetros. Es en este contexto que surge la necesidad de un proyecto que aborde esta brecha cognitiva.

Para esto surge aprendeML, en este trabajo se busca implementar una web interactiva que generalice el conocimiento del aprendizaje automático y lo haga más accesible a cualquier persona interesada en el mundo de los algoritmos. Para ello será necesario implementar una web que tenga varias funcionalidades y varios algoritmos funcionales que simplifiquen este proceso para la enseñanza. En esta búsqueda se definirán algoritmos, el funcionamiento de algoritmos y las tecnologías usadas y necesarias para implementarlos. Se busca generar una plataforma simple, amigable y útil.

1.1 MOTIVACIÓN

Este proyecto de fin de grado surge de una combinación de motivaciones personales y académicas que han guiado mi trayectoria durante la carrera. A lo largo de los años, me he enfrentado a desafíos al estudiar algunos algoritmos concretos en el campo del aprendizaje automático. Hubo momentos en los que la complejidad de algún algoritmo hacía que pasase horas y horas hasta que comprendía su funcionamiento.

Sin embargo, estos momentos de dificultad no solo fueron obstáculos, sino también oportunidades para crecer y superar mis propias limitaciones. En lugar de solo aprender la teoría de aprendizaje automático. Encontrar formas más simples y accesibles de abordar los conceptos de este mismo.

Mi objetivo es utilizar este proyecto de fin de grado como una plataforma para compartir mi experiencia y conocimientos adquiridos a lo largo de la carrera, ofreciendo una perspectiva más simple y accesible sobre el mundo del aprendizaje automático. Quiero apoyar a aquellos que, al igual que yo, han experimentado dificultades al estudiar algoritmos complejos.

Este proyecto no solo se basa en mi pasión por el desarrollo web, especialmente en el *“frontend”*, sino que también representa un desafío personal de contribuir al ámbito de la educación a distancia. En un momento donde la demanda de herramientas y plataformas de e-learning está en aumento, veo esta oportunidad como una forma de mejorar las opciones disponibles actualmente en el mercado.

Este proyecto no solo se limita a ser una plataforma de e-learning más; es un esfuerzo con fundamentos sólidos y el objetivo de simplificar el acceso al conocimiento del aprendizaje automático para aquellos que enfrentan dificultades.

Este proyecto busca facilitar un entendimiento claro sobre cómo operan los algoritmos de aprendizaje automático. Su objetivo principal es proporcionar una experiencia educativa amena y enriquecedora a aquellos que, a pesar de su interés y esfuerzo, enfrentan dificultades para entender los conceptos de cada uno de estos algoritmos de manera más accesible.

1.2 OBJETIVOS

Entendiendo el contexto y la motivación del proyecto, se han definido objetivos que abarcan tanto metas generales como secundarias, incluyendo aspectos técnicos.

Estos objetivos han sido delineados con el fin de alcanzar el propósito global del trabajo de manera integral.

1.2.1 OBJETIVOS GENERALES

El objetivo principal de este proyecto es crear una nueva plataforma para los estudiantes de ingenierías relacionadas con la inteligencia artificial donde se puedan transmitir conocimientos sobre el aprendizaje automático de forma atractiva e interactiva. Incluyendo así mismo herramientas para facilitar el aprendizaje y la exploración de este gran mundo.

1.2.2 OBJETIVOS SECUNDARIOS

- Desplegar el sitio web en el entorno de producción en Azure.
- Integrar redes neuronales en la plataforma.
- Implementar inteligencia artificial en la web utilizando chat GPT para procesar las solicitudes de los usuarios.

1.2.3 OBJETIVOS TÉCNICOS

Procesamiento mediante aprendizaje automático: Diseñar e implementar una REST API que permita gestionar los conjuntos de datos y gestionar la ejecución de algoritmos de clasificación sobre ellos.

Definición de endpoints: Definir los endpoints y las operaciones necesarias para el intercambio de datos relevantes para la correcta comunicación entre la visualización de la aplicación y el procesado de datos.

Aplicación “*frontend*”: Diseño e implementación de un frontal que permita a los usuarios interactuar con las API desarrollados.

1.3 PLANIFICACIÓN

Para poder llevar a cabo el proyecto satisfactoriamente, ha sido necesario organizarse. Esta organización ha sido crucial a la hora de poder llevar a cabo un proyecto de una magnitud como este.

En primer lugar, he intentado analizar qué era lo necesario para llevar a cabo el proyecto. Como cualquier proyecto, antes de empezar a desarrollar es necesario llevar a cabo una búsqueda de información para que el proyecto adquiriera un sentido y una dirección. Una vez se ha adquirido la información necesaria, empieza el periodo de desarrollo que se debe dividir según las necesidades de nuestro proyecto.

Esta búsqueda de información ha tenido como objetivos: adquirir información sobre proyectos previos y definir las tecnologías necesarias. Posteriormente, el desarrollo, al ser un proyecto desarrollado por una persona, se ha tenido que ir implementando de forma simultánea la interfaz gráfica y el servidor donde se llevan a cabo los proyectos. Además, se ha redactado la memoria a medida que se ha desarrollado el proyecto.

1.3.1 DIAGRAMA DE GANTT

En el proyecto, se ha empleado un diagrama de Gantt para estructurar y planificar las diversas etapas y tareas de manera visual y efectiva. Este instrumento gráfico ha sido esencial para representar claramente la secuencia temporal de las actividades, además de distribuir recursos de forma óptima. La visualización detallada de la línea de tiempo y las barras de actividad ha facilitado el seguimiento sistemático del progreso del proyecto, proporcionando una herramienta valiosa para la comunicación interna del equipo y con las partes interesadas.

En resumen, el diagrama de Gantt ha sido una herramienta indispensable para una gestión organizada y transparente de las tareas, asegurando la ejecución eficiente y el cumplimiento de los plazos establecidos.

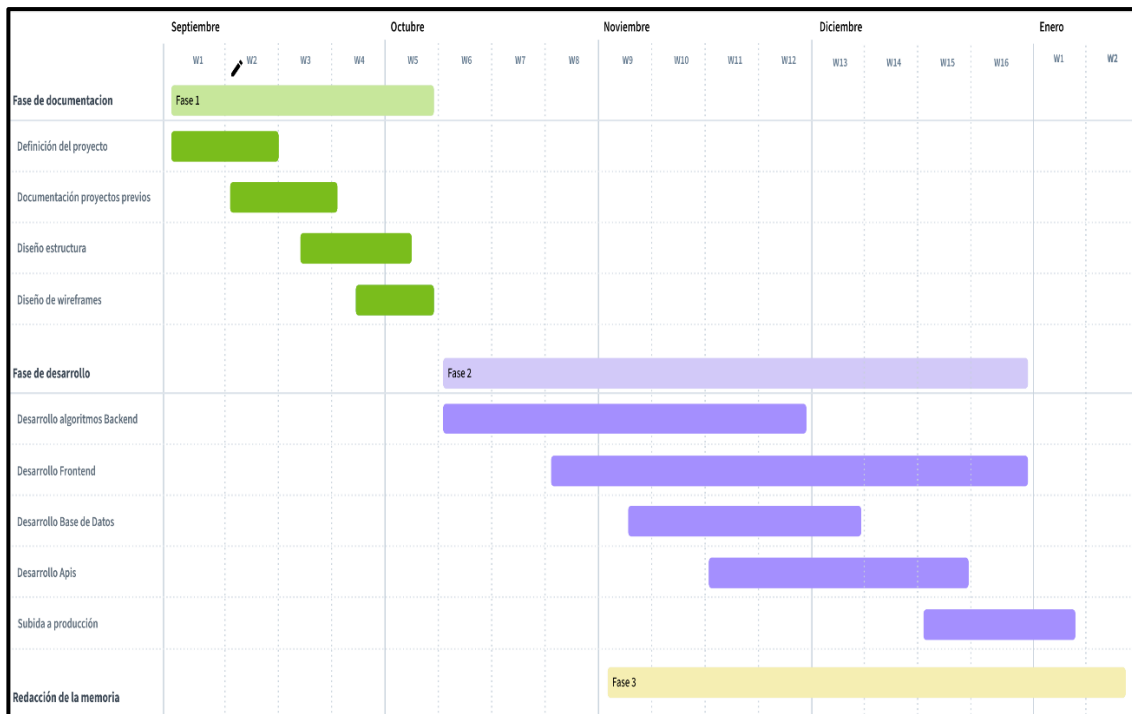


Figura 1: Diagrama de Gantt

Se evidencia que el proyecto se ha estructurado en tres fases principales, distribuidas a lo largo del eje temporal. En la primera fase, se ha llevado a cabo una exhaustiva búsqueda de trabajos previos, se han definido los objetivos específicos del proyecto y se han establecido las bases fundamentales para su desarrollo. La segunda fase se enfocó principalmente en la ejecución y desarrollo práctico del proyecto. Finalmente, en la etapa final del desarrollo, se ha elaborado la memoria correspondiente que recopila y documenta todo el proceso y los resultados obtenidos.

1.3.2 CONTENIDO DE LA MEMORIA

El contenido que encontramos en durante el proceso de elaboración de la memoria está formado por los siguientes ítems:

Estado del arte: Identificación de las plataformas más competentes que encontramos en el ámbito del aprendizaje automático, analizando sus puntos fuertes.

Análisis: Investigación de los requerimientos funcionales y no funcionales que necesita la plataforma y desglose del backlog.

Diseño: Arquitectura de la aplicación y explicación de las tecnologías utilizadas, bases de datos, puesta en producción de la aplicación. Diseño de los “*mock-ups*” de la parte gráfica de la aplicación.

Implementación y resultados: Funcionalidades internas en el ámbito técnico, características y lógica de la aplicación.

Conclusiones: Conclusiones del trabajo con funcionalidades extras que se pueden implementar en un futuro.

Bibliografía: Referencias del origen de la información utilizada para desarrollar el proyecto.

Anexos: Documentación de la aplicación y resultados finales.

2. ESTADO DEL ARTE

La evolución de la tecnología y la creciente relevancia de la inteligencia artificial (IA) han motivado numerosos esfuerzos para hacer accesible el conocimiento sobre algoritmos.

Existen aplicaciones web que han desarrollado funcionalidades sobre esta tecnología. En este caso, nos podemos fijar en diversas implementaciones que pueden servir como antecedentes para nuestro trabajo. Estas tienen dos temáticas principales: facilitar la ejecución de algoritmos en la nube de forma sencilla o intentar aproximar el conocimiento de los algoritmos a la población.

2.1 EJECUCIÓN DE ALGORITMOS

Entre los antecedentes significativos que buscan facilitar la ejecución se destaca la iniciativa de la Universidad de Waikato, que hace casi dos décadas desarrolló el sitio web y la aplicación Weka [2]. Este proyecto pionero tuvo como objetivo facilitar la implementación de algoritmos y su ejecución, marcando el inicio de la accesibilidad a estos conocimientos.

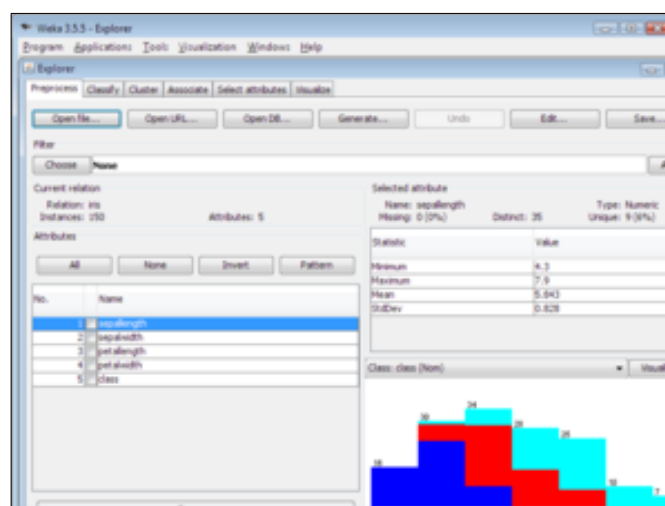


Figura 2: Aplicación Weka

Desde entonces, diversas empresas han emergido con el mismo propósito, destacándose BigML [13] y Orange Data Mining [25]. Estas entidades se especializan en ofrecer productos que simplifican la implementación del aprendizaje automático, dirigidos a un público más amplio. Su contribución ha sido crucial para democratizar

el acceso a herramientas y conocimientos en un campo que, en sus inicios, podría haber parecido exclusivo para expertos.



Figura 3: Aplicación BigML

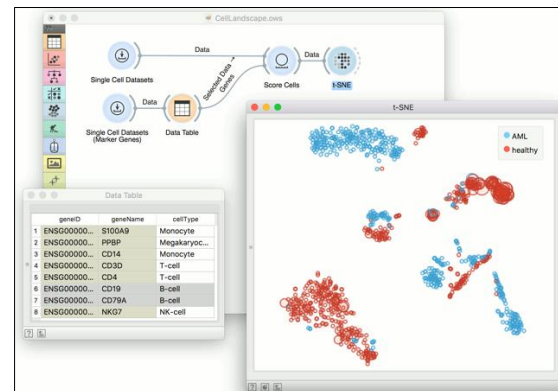


Figura 4: Aplicación Orange Data Mining

2.2 APRENDIZAJE DE ALGORITMOS

Posteriormente, la comunidad de programadores ha desempeñado un papel esencial en el panorama educativo. Plataformas en línea como MLplaygrounds [26] y recursos como Seeing Theory [14] han surgido, proporcionando implementaciones de algoritmos específicos y promoviendo la interacción y comprensión práctica. Estos recursos comunitarios actúan como complemento a las iniciativas académicas e industriales, enriqueciendo el acceso a la información y fomentando la participación de estudiantes y profesionales.

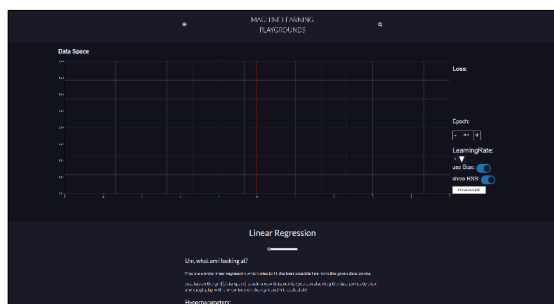


Figura 5: Web MLPlaygrounds

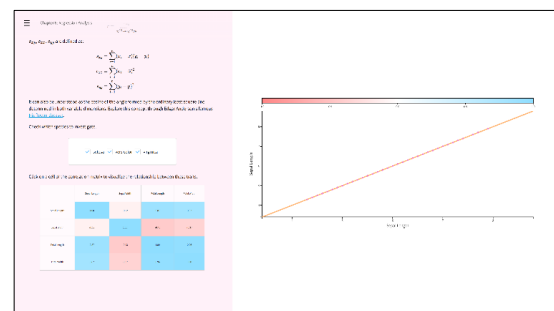


Figura 6 Web Seeing Theory

3. ANÁLISIS

Para llevar a cabo el análisis de necesidades de este proyecto primero se deben establecer los objetivos de los algoritmos que serán necesarios implementar. Asimismo, también se definirán las necesidades funcionales como las no funcionales que debemos suplir para poder organizar este mismo de manera eficiente. Para ello, primero es definido en qué contexto debe estar enfocado el proyecto y a qué tipo de público destinado.

3.1 ALGORITMOS

Existe un gran número de algoritmos con infinitas variaciones de estos. Aunque cada uno destaca en alguna cualidad o característica, hay algunos algoritmos que son más conocidos y usados. Por eso, para introducir a los usuarios y no abrumarlos, se implementará un subgrupo de todos los algoritmos.

3.1.1 ALGORITMOS SELECCIONADOS

Para llevar a cabo este proyecto, es crucial seleccionar cuidadosamente los modelos y el tipo de aprendizaje automático que se utilizará para introducir a los usuarios en este campo. En este sentido, resulta más lógico comenzar desde los cimientos del aprendizaje automático.

En el pasado, el aprendizaje automático no se asemejaba a la forma en que lo conocemos en la actualidad, se refería a lo que se conoce como *“Shallow learning”*, englobando todas las implementaciones anteriores al *“Deep learning”*[3].

Aunque el aprendizaje profundo sea la tecnología que está en la vanguardia en la actualidad, comprenderla plenamente sin conocimientos previos puede resultar desafiante. Por este motivo, el enfoque más apropiado sería implementar algoritmos de *“Shallow learning”* y una red neuronal básica como punto de partida. Este enfoque permitirá a los usuarios familiarizarse gradualmente con los fundamentos antes de adentrarse en el mundo del aprendizaje profundo en el futuro [4].

REGRESIÓN LINEAL

La regresión lineal es un método estadístico que analiza la relación entre dos variables. Busca comprender cómo una variable dependiente cambia en promedio en respuesta a cambios en una variable independiente. La representación gráfica de esta relación se realiza mediante una línea recta llamada línea de regresión, que se ajusta de manera óptima a los datos.

En este contexto, la variable dependiente (y) es la que se estudia y pretende predecir, mientras que la variable independiente (x) se utiliza para predecir la variable dependiente, también conocida como variable predictora. La línea de regresión minimiza la diferencia entre los puntos de datos reales y las predicciones. Los coeficientes de regresión, como la pendiente (m) y la intersección en el eje y (b) en una regresión lineal simple, son parámetros de la ecuación de la línea de regresión. La ecuación, expresada como $y = mx + b$, describe la relación lineal entre las variables. La relación entre estas variables se representa mediante una línea recta, conocida como línea de regresión, que busca minimizar la distancia entre los puntos de datos reales y la predicción de la línea [1]

La regresión lineal se utiliza comúnmente para hacer predicciones y entender la relación entre variables en campos como la economía, la biología, la ingeniería y muchos otros. Es una herramienta versátil y fundamental en el análisis de datos.

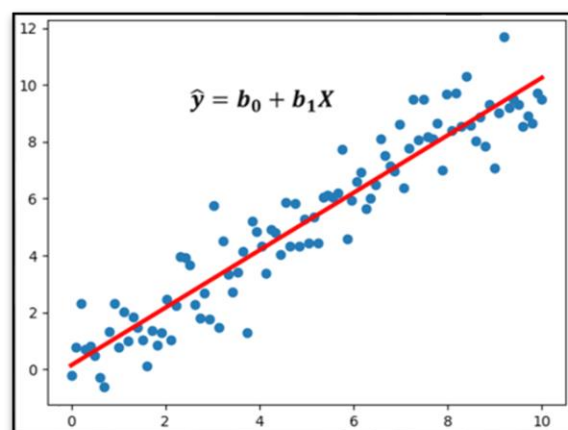


Figura 7: Algoritmo Regresión Lineal

REGRESIÓN LOGARÍTMICA

La regresión logarítmica es un método estadístico que analiza la relación entre dos variables mediante una función logarítmica. Busca comprender cómo una variable dependiente cambia en promedio en respuesta a cambios en una variable independiente, utilizando una representación gráfica mediante una curva logarítmica llamada línea de regresión logarítmica, que se ajusta de manera óptima a los datos.

En este contexto, la variable dependiente (y) es la que se estudia y pretende predecir, mientras que la variable independiente (x) se utiliza para predecir la variable dependiente, también conocida como variable predictora. La línea de regresión logarítmica minimiza la diferencia entre los puntos de datos reales y las predicciones. Los coeficientes de regresión, como la pendiente (m) y la intersección en el eje y (b) en una regresión logarítmica simple, son parámetros de la ecuación de la línea de regresión logarítmica. La ecuación, expresada como $y = m * \ln(x) + b$, describe la relación logarítmica entre las variables. La relación entre estas variables se representa mediante una curva logarítmica, conocida como línea de regresión logarítmica, que busca minimizar la distancia entre los puntos de datos reales y la predicción de la curva.[22]

Es una herramienta versátil y fundamental en el análisis de datos, especialmente cuando la relación entre las variables sigue un patrón logarítmico.

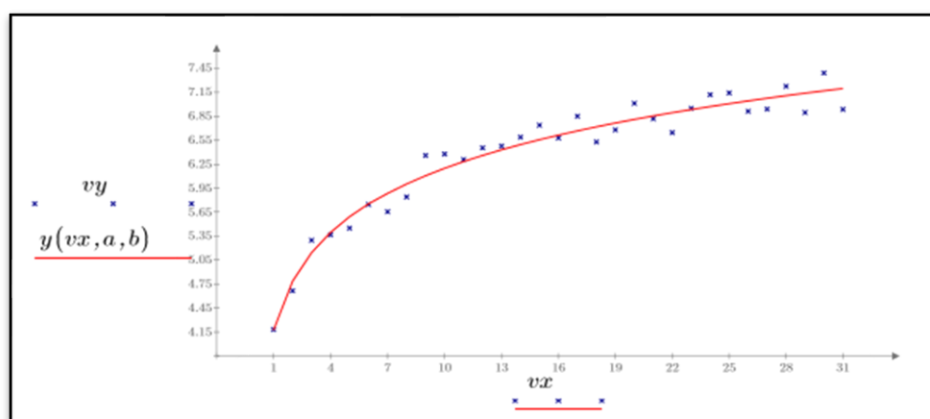


Figura 8: Algoritmo Regresión Logarítmica

K-NEAREST NEIGHBORS

El método K-Nearest Neighbors (K-NN) es un algoritmo de clasificación no paramétrico que estima la función de densidad de las predictoras por cada clase. Utiliza la cercanía entre puntos en el espacio de características para asignar una etiqueta a un punto de datos. Es eficaz para estimar la probabilidad de que un elemento pertenezca a una clase específica.

El algoritmo de vecinos más cercanos es un método de aprendizaje automático que se utiliza para clasificación y regresión. Se fundamenta en la idea de que las instancias de datos similares tienden a agruparse en el espacio de características. K-NN busca determinar la clasificación o el valor de una instancia desconocida basándose en las k instancias más cercanas en el conjunto de entrenamiento. En el contexto de la clasificación, K-NN asigna una etiqueta a una instancia desconocida según la mayoría de las etiquetas de sus k vecinos más cercanos.[20]

La esencia de K-NN radica en la métrica de distancia, que determina la similitud entre instancias. Comúnmente, se utiliza la distancia euclidiana, pero también se pueden emplear otras métricas según la naturaleza de los datos. La elección del valor de k es crucial, ya que afecta la suavidad de las predicciones y la sensibilidad a los detalles locales del conjunto de datos.

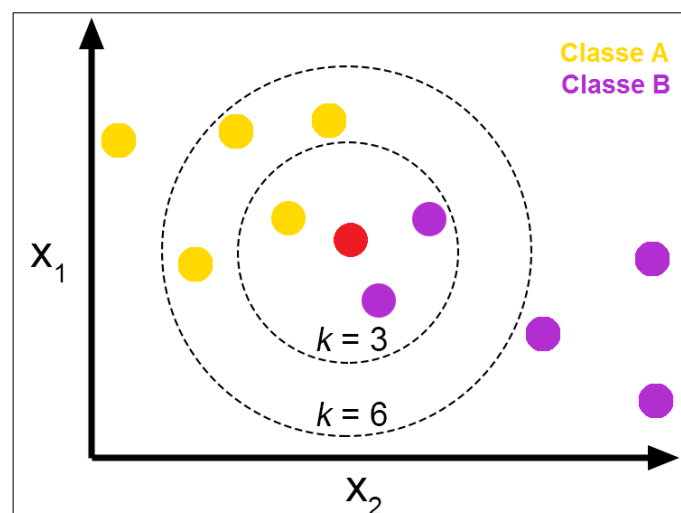


Figura 9: Algoritmo K-nn

K-MEANS

K-Means, en primer lugar, se basa en la noción de centroide, representando el centro geométrico de un grupo de puntos en un espacio multidimensional. Este algoritmo busca agrupar datos de manera que la distancia entre cada punto y el centroide del grupo al que pertenece sea mínima.

Iniciando con un número predeterminado de centroides, el algoritmo itera de manera iterativa para asignar puntos a los grupos y ajustar la posición de los centroides, buscando así minimizar la distancia total dentro de cada grupo. Este proceso se repite hasta que los centroides convergen y la asignación de puntos a grupos se estabiliza.[5]

Es crucial destacar que "K" representa el número de grupos predefinidos por el usuario. Este valor determina la cantidad de centroides iniciales y, por ende, la cantidad final de grupos en los que se clasificarán los datos. Surge como una alternativa al SDG, K-Means busca superar limitaciones computacionales. En lugar de operar sobre todo el conjunto de datos en cada iteración, el algoritmo selecciona aleatoriamente un conjunto reducido de puntos en cada iteración, agilizando el proceso de agrupamiento.

Aunque utilizar el conjunto completo de datos puede proporcionar resultados más precisos, en casos de conjuntos de datos extensos, esta solución resulta inviable en términos de eficiencia computacional. El algoritmo K-Means busca minimizar la varianza dentro del grupo, es decir, la distancia entre los puntos de datos dentro de un clúster, y maximizar la varianza entre clústeres. Es eficiente y efectivo para identificar patrones de similitud en conjuntos de datos, siendo particularmente útil en problemas de segmentación y agrupamiento.

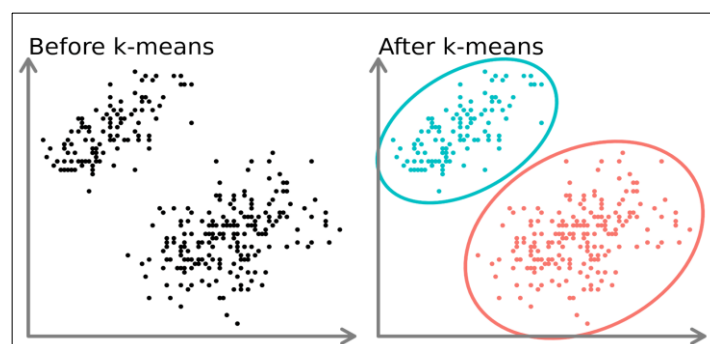


Figura 10 : Algoritmo k-means

NAÏVE BAYES

Un clasificador Naïve Bayes es un clasificador probabilístico fundamentado en el teorema de Bayes junto con algunas hipótesis simplificadoras adicionales. Son estas hipótesis las que determinan la independencia de las variables predictoras, las que le atribuyen el apelativo de naïve o ingenuo.

Este clasificador asume que la ausencia o presencia de una característica no está relacionada con la ausencia o presencia de cualquier otra característica. Por lo tanto, un clasificador ingenuo de Bayes considera que cada una de las características de un elemento contribuye de manera independiente a la probabilidad de que ese elemento forme parte de un conjunto o no, independientemente del resto de características[7].

Una gran ventaja de este método es que requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios para la clasificación.

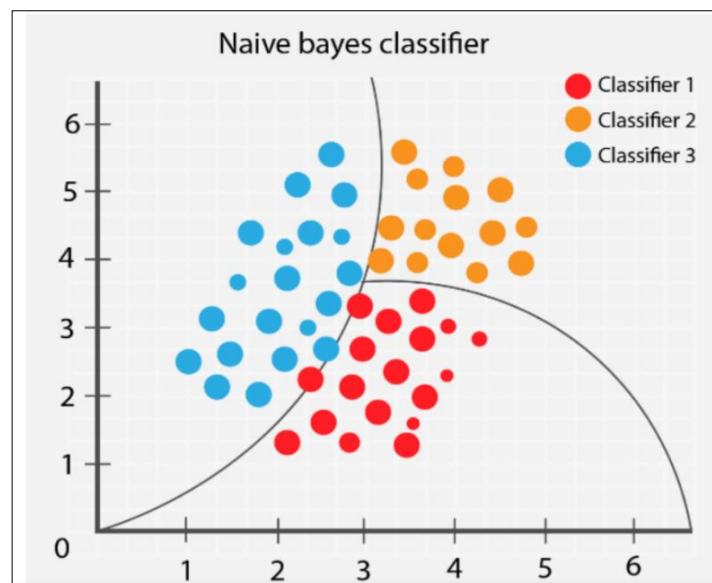


Figura 11: Algoritmo Naïve Bayes

SUPPORT VECTOR MACHINE (SVM)

La “*Support Vector Machine*” (SVM) asigna etiquetas a los datos de entrenamiento en diversas clases, representándolos como puntos en el espacio con el objetivo de lograr una separación entre las clases, maximizando el espacio entre ellas. Durante la fase de entrenamiento, se busca encontrar un hiperplano que maximice la distancia con los puntos más cercanos a él. Luego, en la fase de prueba, los datos son clasificados según su proximidad a cada clase[9].

La característica distintiva de las SVM radica en la búsqueda de una separación óptima. Estos algoritmos buscan identificar el hiperplano que tenga la máxima distancia con respecto a los puntos más cercanos a él. En consecuencia, los puntos a un lado de este hiperplano se etiquetan con una categoría específica, mientras que aquellos al otro lado se etiquetan con otra.

Aunque la separación más simple se logra con líneas o planos rectos, como se ilustra en la Figura 2.6 de ejemplo, en la práctica, los conjuntos de datos para clasificar rara vez presentan la situación ideal de solo dos dimensiones. En lugar de eso, suelen involucrar múltiples variables predictoras, curvas no lineales de separación, clasificaciones en más de dos categorías, entre otras complejidades.

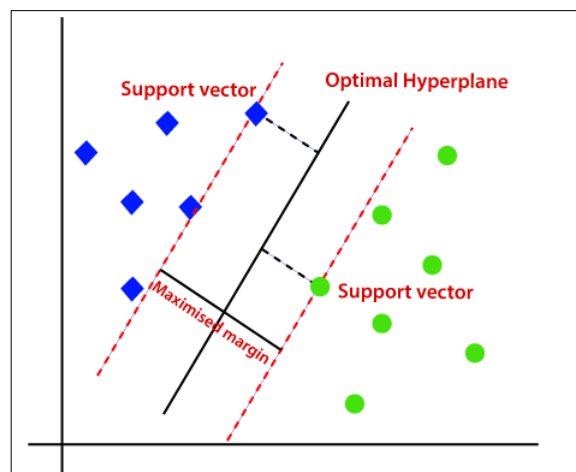


Figura 12: Algoritmo SVM

PCA

El Análisis de Componentes Principales (PCA) es una técnica en el campo de la reducción de dimensionalidad que tiene como objetivo transformar un conjunto de datos original en un nuevo sistema de coordenadas, destacando las direcciones con mayor varianza. Esta transformación permite representar los datos de manera más eficiente y conservando la información esencial.

Durante la fase de entrenamiento, PCA busca identificar los componentes principales, que son las direcciones en las cuales los datos presentan la mayor variabilidad. Estos componentes principales se utilizan para construir un nuevo espacio de características, donde los datos proyectados conservan la mayor parte de su información original.

La característica clave de PCA radica en la búsqueda de una representación óptima que capture la mayor variabilidad en los datos. A través de la identificación de los componentes principales, PCA busca maximizar la varianza a lo largo de estas direcciones en el nuevo espacio, permitiendo así una representación más compacta y significativa de los datos.

En la práctica, PCA se aplica a conjuntos de datos que pueden involucrar múltiples variables originales, superando las limitaciones de representación en dimensiones más altas. Aunque la ilustración más sencilla de PCA puede presentar un cambio a lo largo de líneas o planos rectos, la técnica es particularmente valiosa cuando los datos exhiben complejidades, como relaciones no lineales o la presencia de múltiples características interrelacionadas.[23]

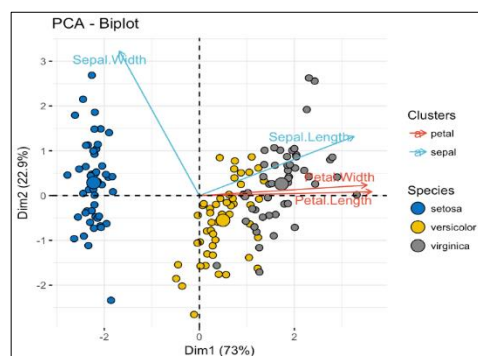


Figura 13: Algoritmo PCA

REDES NEURONALES

Las redes neuronales asignan valores o etiquetas a los datos de entrenamiento a través de un proceso inspirado en el funcionamiento del cerebro humano, utilizando nodos interconectados llamados neuronas artificiales. Estas redes procesan la información a través de múltiples capas, cada una compuesta por neuronas que transforman y pasan la información a la siguiente capa. Durante el entrenamiento, se ajustan los pesos de las conexiones entre las neuronas para minimizar el error en las predicciones.

La característica distintiva de las Redes Neuronales radica en su capacidad para aprender patrones complejos y no lineales en los datos. Estos modelos buscan optimizar una función de pérdida, ajustando los parámetros internos (como los pesos y sesgos) para mejorar la precisión de las predicciones. Como resultado, las redes neuronales pueden capturar relaciones y estructuras subyacentes en conjuntos de datos complejos que no son fácilmente separables por métodos más simples[17].

Aunque las redes neuronales pueden comenzar con estructuras y conexiones simples, su capacidad de adaptación les permite modelar relaciones más complicadas. En la práctica, los conjuntos de datos que se abordan raramente se limitan a un número reducido de dimensiones o presentan relaciones lineales claras. En cambio, los desafíos suelen incluir múltiples características, patrones no lineales, y la necesidad de clasificar en múltiples categorías, lo que requiere redes neuronales con múltiples capas y neuronas interconectadas para abordar estas complejidades.

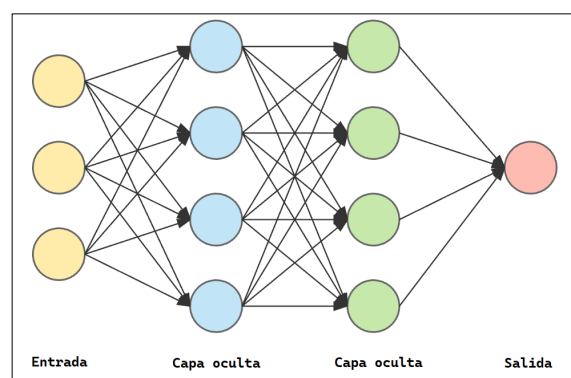


Figura 14: Diagrama Red Neuronal

3.1.2 VISUALIZACIÓN DE ALGORITMOS

Los gráficos interactivos desempeñan un papel crucial en este trabajo, especialmente en el ámbito del aprendizaje, donde la visualización de datos y resultados es fundamental [2]. La capacidad de interactuar con gráficos proporciona una dimensión adicional al objetivo de este proyecto, en la que se ejecutan modelos y se exploran patrones complejos. Estos ofrecen una amplia variedad de ventajas:

Exploración de Datos Dinámica: Los gráficos interactivos permiten una exploración de datos más dinámica y detallada. Puedo ajustar parámetros, filtrar información y examinar visualmente diferentes aspectos de los conjuntos de datos. Esto facilita la identificación de patrones sutiles y la comprensión profunda de las características de los datos, lo cual es esencial en la fase de preprocesamiento en *“machine learning”*.

Evaluación de Modelos en Tiempo Real: Durante el desarrollo de modelos de *“machine learning”*, es crucial evaluar su rendimiento. Los gráficos interactivos permiten la visualización en tiempo real de las métricas de evaluación, facilitando la comparación entre modelos y la toma de decisiones informadas sobre ajustes y mejoras.

Interactividad en la Enseñanza: Al utilizar gráficos interactivos, se logra un mayor compromiso por parte de los estudiantes. La capacidad de manipular gráficos en tiempo real proporciona una experiencia más participativa y estimulante, permitiendo a los estudiantes, explorar conceptos y experimentar con modelos de manera práctica.

Comunicación Efectiva de Resultados: En la presentación de resultados, la visualización interactiva es una herramienta poderosa. Puedo destacar áreas específicas de interés y permitir que otros interactúen directamente con los resultados. Esto no solo facilita la comprensión, sino que también fomenta la colaboración y el intercambio de ideas.

Retroalimentación Instantánea: La interactividad de los gráficos proporciona una *“feedback”* instantáneo, lo que es invaluable en el proceso de ajuste y mejora de modelos. Puedo realizar ajustes sobre la marcha, observar cómo afectan los cambios y tomar decisiones informadas basadas en resultados visuales inmediatos.

3.2 ANÁLISIS DEL CONTEXTO

Para poder entender el aprendizaje automático, se debe tener un mínimo de conocimientos sobre algorítmica y de programación. Por lo tanto, este es un proyecto destinado a todos los públicos que no tienen grandes conocimientos sobre aprendizaje automático, en concreto el target principal se definiría como los estudiantes de alguna ingeniería o grado técnico que requiera de entender sobre estos algoritmos y aprender como implementarlos.

3.2.1 IDENTIFICACIÓN DE REQUISITOS

REQUISITOS FUNCIONALES

Los requisitos funcionales especifican las funciones y características que el sistema debe cumplir. En este proyecto, se han identificado los siguientes requisitos funcionales:

R1 Gestión de Usuarios:

El sistema debe permitir la creación y modificación de perfiles de usuarios.

R2 Gestión de conjuntos de datos:

Los usuarios deben poder cargar y gestionar conjuntos de datos de manera intuitiva.

R4 Edición de conjuntos de datos:

Debe existir la posibilidad de editar conjuntos de datos previamente cargados.

R5 Implementación de ejemplos:

Los usuarios no registrados deben poder ejecutar una demostración del funcionamiento de cada uno de los algoritmos implementados.

R6 Análisis de conjuntos de datos:

Se requiere la implementación de herramientas de análisis de conjuntos de datos para poder mostrar su estructura y contenido.

R7 Ejecución de modelos:

El sistema debe permitir la ejecución de algoritmos de aprendizaje automático sobre los conjuntos de datos cargados.

R8 Descarga de conjuntos de datos calculados:

Los usuarios deben poder descargar los conjuntos de datos subidos para su uso posterior.

R9 Mostrar resultados interactivos:

La aplicación debe mostrar de manera clara y comprensible los resultados de las ejecuciones realizadas, con un formato interactivo que permita la visualización del resultado calculado.

R10 Mostrar implementación de algoritmos:

La aplicación web debe mostrar una implementación en código Python sobre todos los algoritmos para facilitar su comprensión.

REQUISITOS NO FUNCIONALES

RNF1 Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de usar, priorizando la accesibilidad para perfiles no técnicos.

RNF2 Seguridad:

La aplicación debe implementar medidas de seguridad robustas, incluyendo la encriptación de datos y la autenticación segura de usuarios.

RNF3 Rendimiento:

El sistema debe ser capaz de manejar las implementaciones de los distintos algoritmos de datos de manera eficiente, garantizando tiempos de respuesta aceptables.

RNF4 Escalabilidad:

La arquitectura del sistema debe ser escalable para adaptarse a un crecimiento futuro de usuarios y datos.

RNF5 Compatibilidad:

La aplicación debe ser compatible con una variedad de navegadores web para garantizar una experiencia consistente.

RNF6 Fiabilidad:

El sistema debe ser confiable, minimizando el riesgo de fallos y asegurando la integridad de los datos funcionales.

3.2.2 BACKLOG

En este apartado se presenta el concepto de “*backlog*” personalizado para el proyecto, subrayando su importancia crucial en la planificación y desarrollo de la plataforma. El “*backlog*” se estructura como un listado de funcionalidades priorizadas que incluye tanto los requisitos funcionales como los no funcionales esenciales para lograr los objetivos predefinidos.

A través del “*backlog*”, se establece un enfoque claro, se dan prioridades a las tareas y se fomenta la colaboración entre los miembros del equipo de desarrollo. Esta sección explora cómo la utilización del “*backlog*” en el proyecto permite una gestión eficiente del proyecto, asegurando que las necesidades de los usuarios y los objetivos del trabajo se aborden de manera efectiva. Con estas historias de usuario, podemos visualizar el “*backlog*” completo de todas las funcionalidades de este proyecto.

UNR1

Como usuario no registrado, quiero ejecutar demostraciones de los algoritmos implementados para comprender su funcionamiento antes de registrarme.

El usuario puede acceder a la página principal sin un registro previo.

El usuario puede acceder a todas las explicaciones y “*demos*” de forma gratuita y sin ningún impedimento.

El usuario tiene bloqueado cualquier acceso a páginas que requieran inicio de sesión.

U1

Como nuevo usuario, quiero poder crear una cuenta para acceder a las funciones completas de la plataforma.

El usuario debe poder acceder al formulario de registro desde cualquier página de la web.

El usuario, al registrarse, debe ser redirigido a la página de inicio de sesión.

UR1

Como usuario registrado, quiero poder iniciar sesión con mi cuenta existente para acceder a todas las funcionalidades de la plataforma.

El usuario debe poder acceder al formulario de inicio de sesión desde cualquier página de la web.

El usuario en completar el inicio de sesión satisfactorio debe ser redirigido a la página principal.

UR2

Como usuario registrado, quiero poder crear y modificar mi perfil para personalizar mi experiencia en la plataforma.

El usuario debe poder modificar su perfil en todo momento.

Al modificar los datos, se debe abrir la pestaña de inicio de sesión con los datos actualizados.

UR3

Como usuario registrado, quiero poder seguir a otros usuarios para recibir actualizaciones sobre sus actividades y descubrir nuevos contenidos relevantes.

El usuario debe tener la opción de seguir a cualquier usuario para llevar un seguimiento de su actividad.

El usuario debe tener una opción de ver la actividad reciente de sus seguidos.

UR4

Como usuario registrado, quiero poder cargar y gestionar conjuntos de datos de manera sencilla para facilitar el proceso de análisis.

El usuario debe poder cargar un “*dataset*” propio.

El usuario debe poder añadir título, descripción y configurar el dataset como quiera.

El usuario debe poder eliminar el “*dataset*”.

UR5

Como usuario registrado, necesito la capacidad de editar conjuntos de datos previamente cargados para corregir o actualizar la información según sea necesario.

El usuario debe poder cambiar el nombre del “*dataset*”.

El usuario debe poder cambiar la descripción del “*dataset*”.

El usuario debe poder cambiar el tipo del “*dataset*”.

El usuario debe poder cambiar el “*dataset*”.

UR6

Como usuario, quiero ejecutar algoritmos de aprendizaje automático sobre los conjuntos de datos cargados para obtener resultados específicos.

El usuario debe poder escoger los parámetros del algoritmo a calcular.

El usuario debe poder visualizar la gráfica generada a partir del cálculo del “*dataset*”.

UR7

Como usuario, quiero utilizar herramientas de análisis de conjuntos de datos para visualizar la estructura y el contenido de manera clara.

El usuario debe poder visualizar todos los datos del “*dataset*” para analizarlos.

UR8

Como usuario, quiero que la aplicación presente los resultados de las ejecuciones de manera clara e interactiva para comprender y explorar los resultados fácilmente.

El usuario debe poder interactuar con la ejecución.

La visualización debe ser fidedigna con el modelo.

UR9

Como usuario, quiero poder buscar conjuntos de datos subidos por otros usuarios para encontrar conjuntos de datos relevantes para sus análisis.

El usuario debe poder tener la opción de buscar conjuntos de datos por el título en un buscador.

Los conjuntos de datos deben mostrarse por parecido a la búsqueda en título.

UR10

Como usuario, necesito poder descargar los modelos de clasificación generados para utilizarlos posteriormente en mis proyectos.

Debe existir un botón para descargar el “*dataset*”.

Al pulsar, el usuario debe iniciar una descarga del “*dataset*”.

UR11

Como usuario, quiero poder cerrar sesión para salir de mi cuenta.

El usuario debe poder tener la opción de cerrar sesión.

Al cerrar sesión, debe ser redirigido a la página principal si se encuentra en una página solo para usuarios registrados.

4. DISEÑO

Para poder llevar a cabo el proyecto debemos establecer cómo será la estructura para poder cumplir los requisitos establecidos anteriormente. Llevar un diseño eficiente y claro permite que, en el momento de empezar la implementación del código, se tenga claro cómo proceder en todo momento.

4.1 TECNOLOGÍAS

Se necesitarán usar diversas tecnologías para desarrollar este proyecto. Se dividirá en dos grupos, las tecnologías usadas en la gestión de datos y las tecnologías usadas en la interfaz gráfica.

4.1.1 SERVIDOR

El denominado servidor se describe como el área de la programación que se especializa en la gestión de todos los procesos que no sean visuales en la aplicación.

PYTHON

El desarrollo del servidor del proyecto se llevará a cabo mediante el uso de Python, un lenguaje de programación extremadamente versátil y de amplio reconocimiento. La elección de Python se fundamenta en varios aspectos clave, entre ellos su sintaxis clara, su extenso ecosistema de bibliotecas y su destacada eficiencia para gestionar las operaciones esenciales requeridas tanto para el procesamiento de datos como para la ejecución de modelos de aprendizaje automático. La versatilidad y robustez de Python no solo facilitan el desarrollo del servidor, sino que también contribuyen significativamente a la escalabilidad y mantenimiento del sistema, asegurando un rendimiento óptimo en todas las facetas del proyecto.

PLOTLY

En el contexto del servidor, se emplea Plotly Express, una biblioteca de Python, para simplificar y potenciar la creación de gráficos interactivos. Esta herramienta ofrece una amplia variedad de opciones de visualización. El punto fundamental radica en su compatibilidad con la interfaz gráfica, ya que existe la misma librería para el framework que usaremos.

SCIKIT-LEARN

También conocida simplemente como scikit, es una biblioteca de aprendizaje automático muy influyente en el entorno Python. Esta herramienta, ampliamente utilizada en nuestro proyecto, ofrece un conjunto robusto de herramientas eficientes diseñadas para abordar diversas tareas en el ámbito del aprendizaje automático, tales como clasificación, regresión y “*clustering*”[18].

La presencia de scikit-learn en nuestro proyecto desempeña un papel fundamental al facilitar el desarrollo de modelos predictivos. Esta capacidad nos permite incorporar análisis predictivos y de clasificación de manera precisa y eficaz en nuestro sistema. La versatilidad de scikit-learn, proporciona una base sólida para la implementación, contribuyendo así a la eficacia y precisión de nuestra web.

PANDAS

Pandas es una librería especializada en Python diseñada para la manipulación y análisis eficiente de datos mediante estructuras flexibles. Esta herramienta desempeña un papel crucial en diversas etapas del proceso, destacando especialmente en la limpieza y transformación de datos. Pandas simplifica la preparación de la información, permitiendo que los datos estén listos para su procesamiento y visualización. Su capacidad para trabajar con conjuntos de datos complejos y realizar operaciones sofisticadas facilita la obtención de información significativa y prepara los datos de manera óptima para su posterior análisis y presentación.

FLASK

Para asegurar una comunicación eficiente en mi proyecto basado en Flask, he optado por utilizar Flask, una herramienta reconocida por su capacidad para simplificar el desarrollo de APIs sólidas y de alto rendimiento. La elección de Flask se debe a su habilidad para agilizar la configuración y gestión de endpoints de manera rápida y estructurada[19].

La sintaxis declarativa de Flask contribuye significativamente a un desarrollo eficiente de APIs. Este enfoque no solo facilita la construcción de interfaces robustas,

sino que también garantiza una comunicación fluida entre la interfaz gráfica y el servidor del sistema. La elección de Flask como framework se basa en sus capacidades para establecer una API sólida y confiable, respaldando la interacción efectiva entre las distintas partes del sistema.

SQLITE

SQLite es un sistema de gestión de bases de datos relacional que se caracteriza por ser ligero, sin necesidad de un servidor independiente y basado en archivos. Fue diseñado para ser eficiente, fácil de usar y autónomo.

4.1.2 INTERFAZ GRÁFICA

El denominado “*frontend*” se describe como el área de la programación donde que se especializa en la interfaz gráfica donde el usuario interactúa con todos los procesos y recursos que ofrece la aplicación.

TYPESCRIPT

En el desarrollo de la interfaz gráfica, se ha optado por incorporar “*TypeScript*”, una versión heredada de JavaScript. Esta elección estratégica no solo amplía las capacidades del lenguaje base, sino que también mejora la calidad del código de manera significativa. Al provenir de JavaScript, “*TypeScript*” introduce características adicionales, como el uso de tipos estáticos, lo que contribuye a reducir errores comunes durante el desarrollo. Esta mayor capacidad para detectar y prevenir posibles problemas en tiempo de compilación resulta fundamental para garantizar la robustez y estabilidad del código.

Además, la adopción de “*TypeScript*” simplifica el mantenimiento del proyecto a medida que evoluciona. La introducción de tipos proporciona una documentación implícita del código, lo que facilita la comprensión y modificación por parte de los desarrolladores a lo largo del tiempo. Esta mejora en la legibilidad y estructura del código se traduce en una mayor eficiencia y agilidad en el desarrollo continuo de la interfaz gráfica, asegurando un proceso más fluido y una base sólida para futuras actualizaciones.

ANGULAR

La interfaz gráfica de la aplicación se construye utilizando Angular, un framework robusto y modular diseñado para el desarrollo de aplicaciones web. Angular ofrece una estructura organizada y modular que simplifica significativamente la creación de interfaces de usuario interactivas y escalables. [24] Este framework se destaca por sus componentes reutilizables y su arquitectura basada en módulos, lo que facilita la administración de la complejidad inherente a la interfaz de usuario. Gracias a estas características, Angular proporciona un entorno de desarrollo eficiente y estructurado para crear interfaces de usuario que no solo son visualmente atractivas, sino también fáciles de mantener y escalar[15][13].

PLOTLY

Su uso en la interfaz gráfica se alinea perfectamente con el framework que estamos empleando, ya que nos crea una sinergia que evita posibles errores o incompatibilidades. Esta sinergia permite una transición suave en la implementación de gráficos interactivos entre el servidor y la interfaz gráfica, puesto que ambas instancias comparten la misma librería. La versatilidad de Plotly en la interfaz gráfica facilita la generación de visualizaciones dinámicas y atractivas, brindando a los usuarios una experiencia de usuario enriquecida e interactiva.

4.2 ARQUITECTURA

La arquitectura fundamental de este sistema se basará en un modelo cliente-servidor, donde el cliente, también conocido como “*frontend*” o interfaz gráfica, realizará solicitudes y enviará peticiones al servidor, denominado “*backend*” o servidor. Ambas entidades poseerán estructuras arquitectónicas distintas, diseñadas para cumplir funciones específicas y complementarse mutuamente a través de llamadas a una interfaz de programación de aplicaciones (API).

En este enfoque, el cliente y el servidor interactuarán de manera colaborativa, desempeñando roles diferenciados pero complementarios en el proceso general del sistema. La interfaz gráfica, o cliente, se centrará en la presentación y la interacción directa con el usuario, proporcionando una interfaz intuitiva y amigable. Por otro lado, el servidor, se encargará de gestionar la lógica, procesar las solicitudes del cliente y manejar la persistencia de datos.

Ambas arquitecturas, aunque distintas en sus funciones y responsabilidades, se integrarán de manera efectiva a través de interfaces de programación de aplicaciones (API) bien definidas. Estas API actuarán como puntos de conexión que permitirán la comunicación y transferencia de datos entre la interfaz gráfica y el servidor. La implementación de llamadas a la API facilitará la coordinación de los procesos, asegurando una colaboración entre las dos capas de la arquitectura.

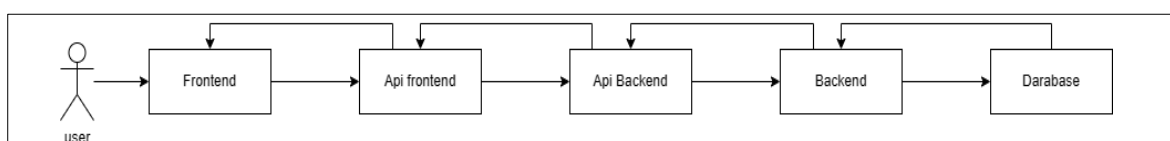


Figura 15: Diagrama

El usuario se comunica con la web a través del navegador web, este interactuará principalmente con la interfaz gráfica de la web, como lo haría en una web convencional no relacionada con el mundo del aprendizaje automático. Sin embargo, la API actúa como intermediario entre la interfaz gráfica y el usuario cuando hay una solicitud de cálculo sobre un “*dataset*”. Este envía una solicitud al servidor y, dependiendo de la solicitud, hace cálculos sobre el “*dataset*” o interactúa con los datos de usuario.

4.2.1 INTERFAZ GRÁFICA

La interfaz gráfica se encarga de mostrar todas las opciones y posibilidades que ofrece la aplicación web; por lo tanto, debe tener un diseño con una estructura intuitiva y que facilite al usuario navegar sin encontrarse obstáculos.

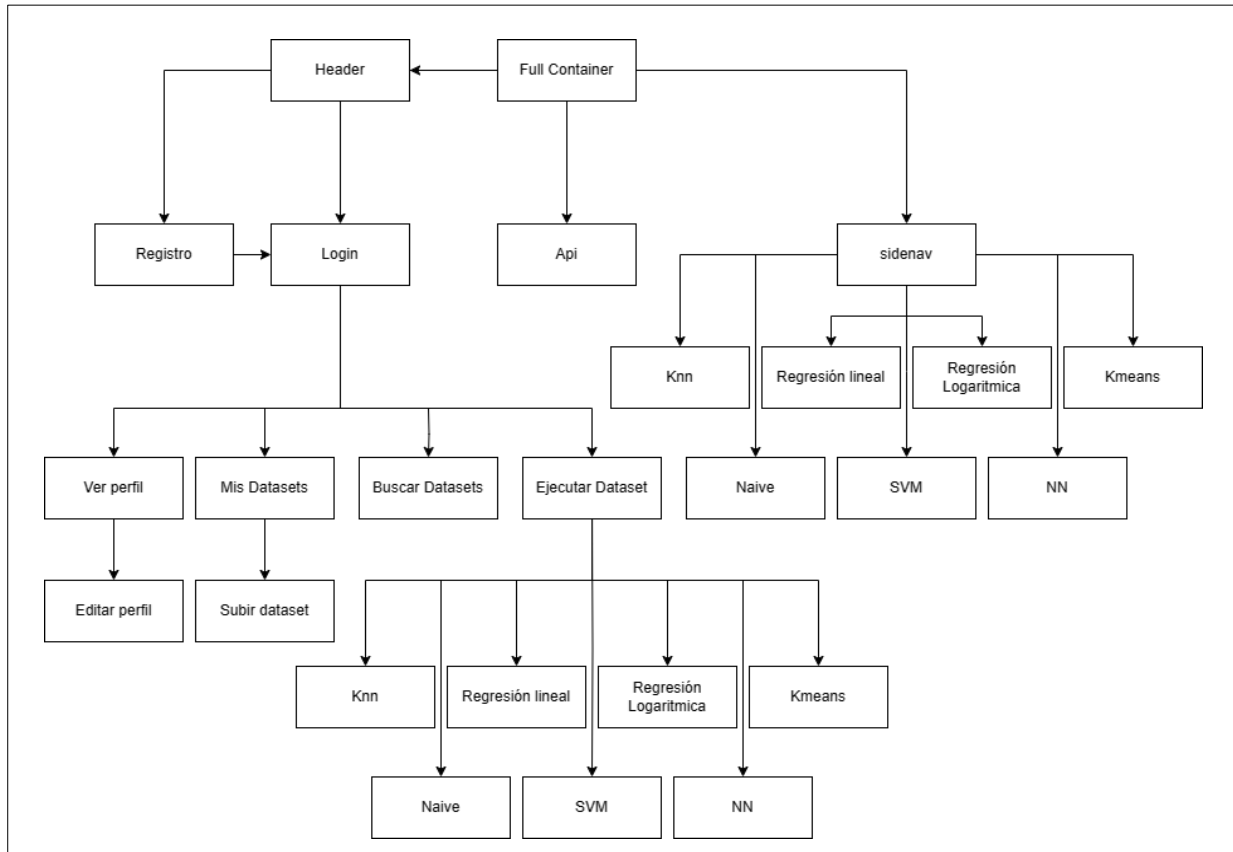


Figura 16

Como se evidencia en la imagen, la interfaz gráfica comienza con un módulo principal que contiene únicamente el “Full Container”, encargado de cargar el módulo completo. Dentro de este módulo principal se encuentran tres componentes fundamentales. En primer lugar, el encabezado que alberga los botones de inicio de sesión, registro y la navegación a la página principal. En segundo lugar, el componente de la barra de navegación lateral, que recopila los subdirectorios del módulo router, donde residen todas las cargas de los módulos de la aplicación. Al intentar acceder a estos subdirectorios, redirige la página al módulo correspondiente. Por último, se encuentra el componente que sirve como enlace con el servidor para el cual se envían las solicitudes de cada pestaña de la web, donde se mostrarán los contenidos.

Dentro de esta arquitectura, es relevante destacar el contenido de un tipo específico de módulo: aquellos dedicados a la implementación y desarrollo de un modelo. Estos módulos hacen uso de un componente denominado “app-plot”, encargado de renderizar en el cliente el contenido de un gráfico y hacer que este sea interactivo.

4.2.2 SERVIDOR

En el diagrama del servidor, se establece una conexión crucial entre la interfaz gráfica y el servidor a través de una API basada en Flask. Esta API actúa como un intermediario eficiente que gestiona las interacciones entre el usuario en la interfaz gráfica y el servidor, facilitando la comunicación y el intercambio de datos.

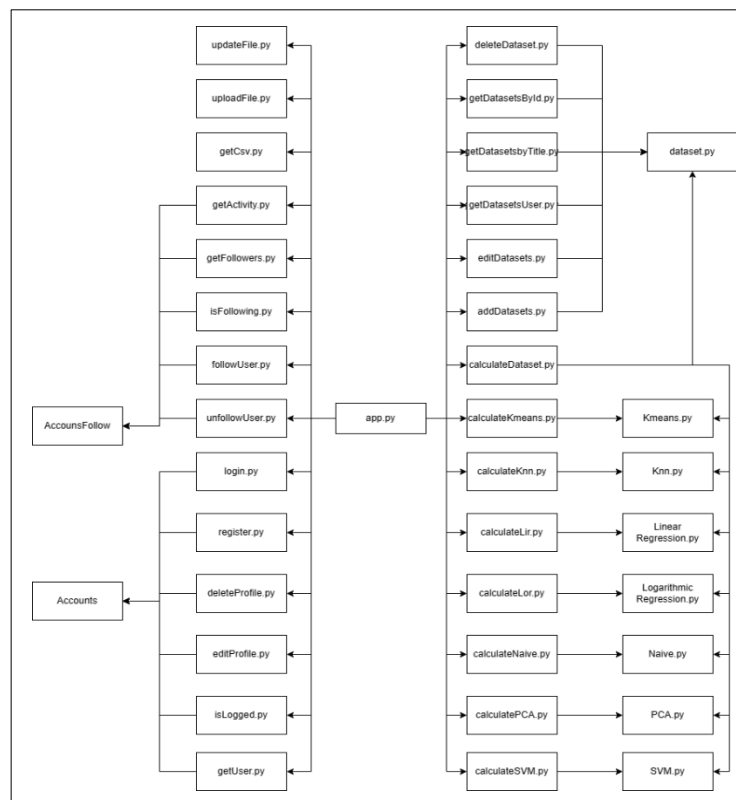


Figura 17

Cuando la interfaz gráfica realiza una llamada a la API, se desencadenan tres tipos distintos de peticiones, cada una abordando una funcionalidad específica. La primera petición implica la ejecución de cálculos sobre datos utilizando un algoritmo de aprendizaje automático. Este proceso implica la manipulación de datos a través del algoritmo, lo que permite al usuario obtener resultados predictivos o analíticos.

La segunda petición se centra en la gestión de usuarios. Esta funcionalidad abarca aspectos como el inicio de sesión, el registro de nuevos usuarios y la administración de perfiles. La API, se encarga de gestionar estas operaciones de manera segura y eficiente, garantizando la integridad y la seguridad de la información del usuario.

Por último, la tercera categoría de peticiones está relacionada con la manipulación de archivos. Los usuarios pueden realizar solicitudes para cargar archivos al servidor o descargar archivos específicos desde el servidor. Esto puede ser útil para intercambiar datos, documentos o cualquier otro tipo de información entre la interfaz gráfica y el servidor.

4.3 WIREFRAMES

Los “*wireframes*” o bocetos de diseño, son representaciones esquemáticas y estructuradas de la interfaz de usuario, actuando como mapas visuales que delimitan la disposición y funcionalidad de los elementos clave. En este segmento, se explorará la importancia de los bocetos de diseño en la creación de experiencias de usuario coherentes y eficientes. A través de estos bocetos iniciales, se da vida a la arquitectura del proyecto, proporcionando una visión clara de la disposición de los elementos y la interacción entre ellos.

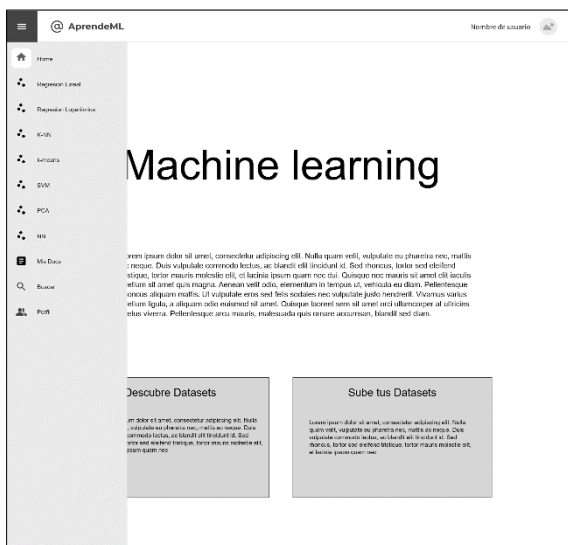


Figura 18: Wireframe pagina principal

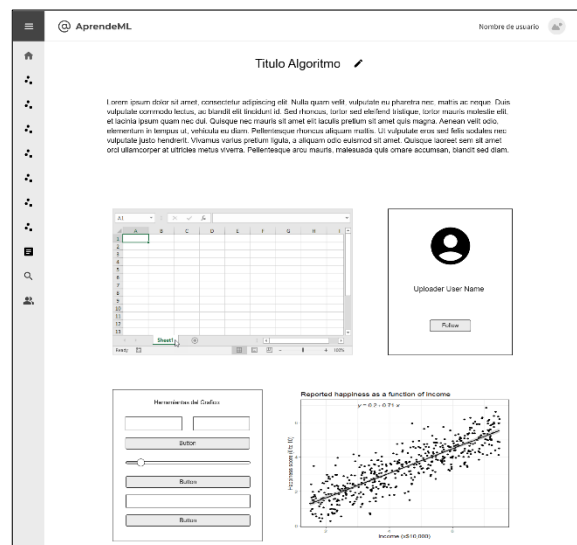


Figura 19 : Wireframe Dataset Usuario

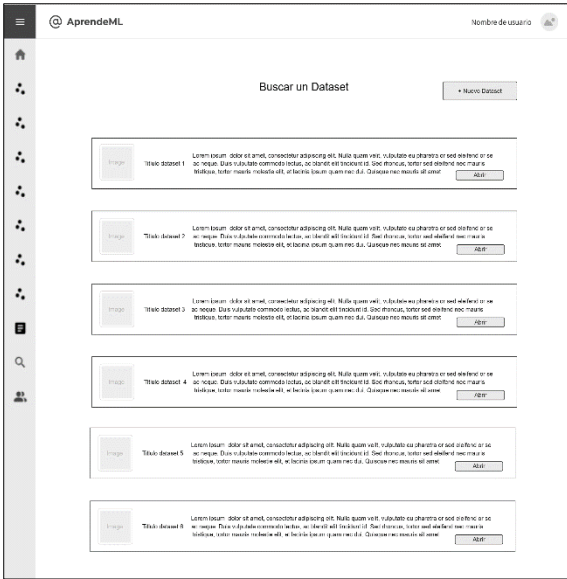


Figura 20: Wireframe Mis datasets

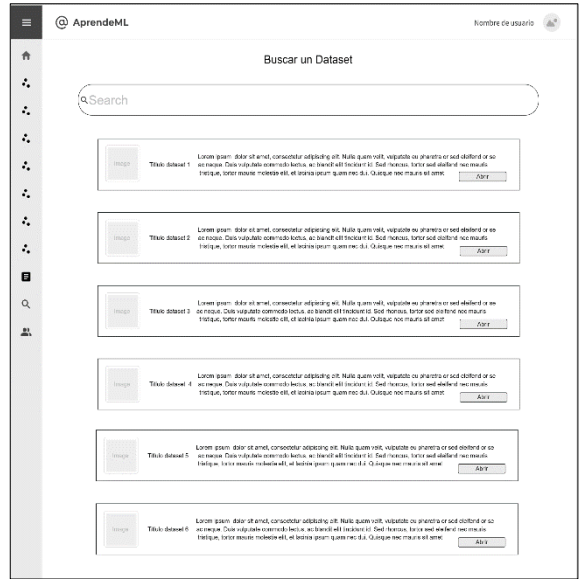


Figura 21 Wireframe buscar Dataset

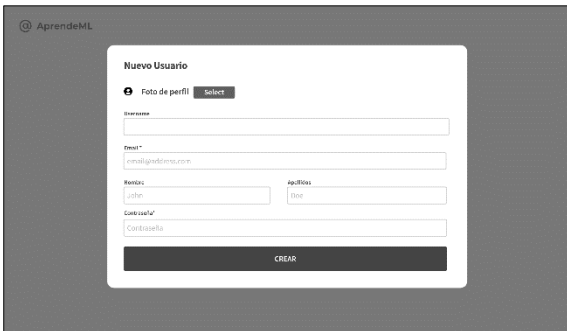


Figura 22: Wireframe Registro

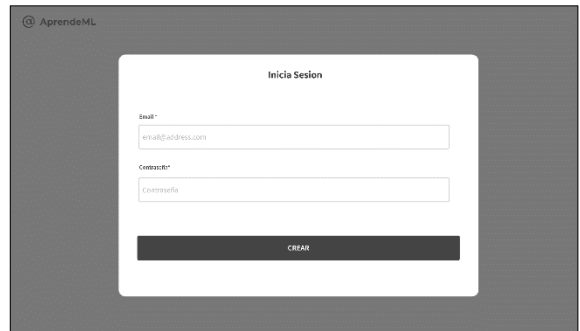


Figura 23: Wireframe inicio de sesión

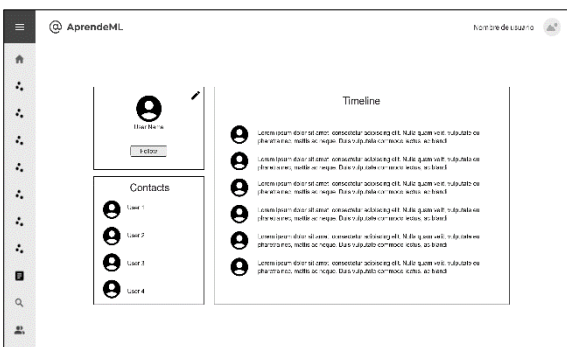


Figura 24: Wireframe mi perfil

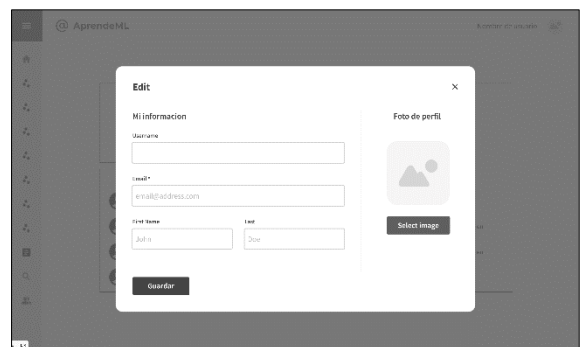


Figura 25: Wireframe editar perfil

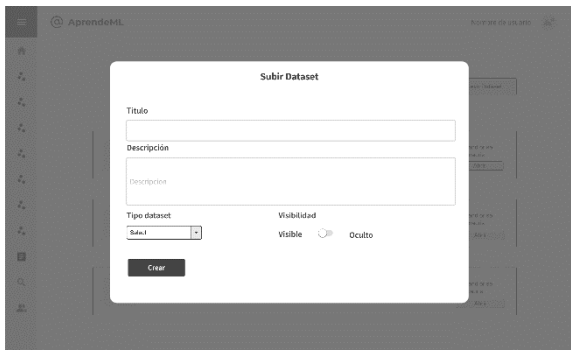


Figura 26: Wireframe subir Dataset

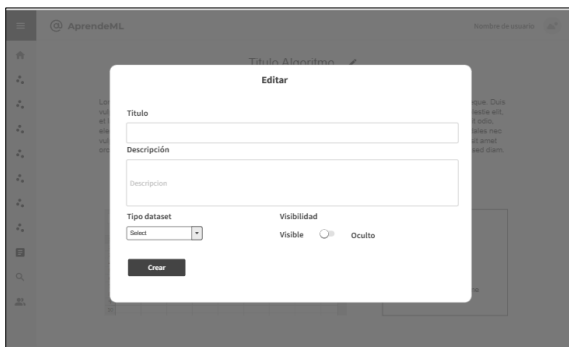


Figura 27: Wireframe editar Dataset



Figura 28: Wireframe vista Algoritmo

5. IMPLEMENTACIÓN

En este apartado se describen los puntos sobre los cuales se han implementado los requisitos y diseños anteriormente descritos.

5.1 BASE DE DATOS

Para iniciar el proyecto, se necesitó establecer una base de datos para almacenar información esencial. Para desplegar esta base de datos, se optó por utilizar “*Flask*” y “*Flask-SQLAlchemy*”, dos herramientas fundamentales en el ámbito del desarrollo web con Python.

En el archivo principal de la aplicación, denominado `app.py`, se configuró la aplicación “*Flask*” y se conectó “*SQLAlchemy*” para manejar las interacciones con la base de datos. Estas configuraciones se reflejan en las siguientes líneas de código:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///data.db'
db = SQLAlchemy(app)
```

Esta configuración especifica que se está utilizando una base de datos SQLite para simplificar el desarrollo, aunque se podría cambiar fácilmente a una base de datos más robusta, como MySQL, simplemente ajustando la URL de conexión.

A continuación, se definieron los modelos de datos, que representan esencialmente las tablas de la base de datos. Se crearon tres modelos: `Usuario`, `Dataset` y `UserFollower`, cada uno representando entidades específicas del sistema.

```
class DatasetModel(db.Model):
    __tablename__ = 'Dataset'
    dataset_id = db.Column(db.Integer, primary_key=True, unique=True)
    title = db.Column(db.String(30), nullable=False)
    textContent = db.Column(db.String(500), nullable=False)
    typeDat = db.Column(db.String(30), nullable=False)
    visibility = db.Column(db.Boolean, nullable=False)
    src = db.Column(db.String(255), nullable=False)
    # Correct the table name in the ForeignKey
    user_id = Column(db.Integer, ForeignKey('accounts.user_id'))
    user = relationship('AccountsModel', foreign_keys=[user_id])
    date = db.Column(db.DateTime, nullable=False)
```

```

class AccountsModel(db.Model):
    __tablename__ = 'accounts'
    user_id = Column(Integer, primary_key=True, unique=True,
nullable=False)
    username = Column(String(30), unique=True, nullable=False)
    name = Column(String(30), nullable=False)
    surnames = Column(String(50), nullable=False)
    email = Column(String(50), nullable=False)
    birthdate = Column(DateTime, nullable=False)
    password = Column(String(50), nullable=False)
    account_image = Column(String(), nullable=False)
    description = Column(String(), nullable=False)
    datasets = relationship('DatasetModel', back_populates='user')

class AccountsFollowModel(db.Model):
    __tablename__ = 'followUsers'
    id = Column(Integer, primary_key=True, unique=True)
    follower_id = Column(Integer, ForeignKey('accounts.user_id'))
    user_id = Column(Integer, ForeignKey('accounts.user_id'))
    following = relationship("AccountsModel", foreign_keys=[follower_id])

```

Posteriormente, se ejecutó un script para crear la base de datos y las tablas correspondientes. Esto se logró mediante el siguiente código:

```

from app import db
db.create_all()

```

Este proceso generó un archivo llamado “*data.db*” en el directorio de la aplicación. Esta base de datos es ahora la estructura que respalda el sistema.

A partir de este punto, se puede interactuar con la base de datos en el código Python, por ejemplo, para agregar un nuevo usuario o recuperar información específica. Con la base de datos configurada y lista para su uso, el proyecto está preparado para avanzar en su desarrollo web con Flask y SQLAlchemy.

5.2 ALGORITMOS

En el proceso de creación de los algoritmos, me he basado en los modelos de “*scikit-learn*”. Las llamadas al entrenamiento de modelos constan de las listas pertinentes de datos de entradas y los posibles parámetros para cada uno de ellos. Una vez creados los modelos, se generan la visualización de métricas adicionales necesarias para cada una de ellas y por última instancia se generan unos gráficos

mediante plotly con el objetivo de conseguirlos ya renderizados y adecuados a cada algoritmo para posteriormente transformarlos a JSON y enviar a la interfaz el gráfico.

5.3 API

Para desarrollar este proyecto ha sido necesario desarrollar una buena comunicación entre el servidor y la interfaz gráfica. Esta comunicación se ha producido mediante una API servidor por parte del servidor y una API cliente por parte de la interfaz gráfica. Para poder llevar a cabo la funcionalidad completa de la página web, se han usado tres tipos de llamadas:

En primer lugar, las llamadas “*calculate*”. Estas llamadas se usan con el fin de visualizar y trabajar con los datos de cada uno de los modelos desarrollados, en estas llamadas se tiene en cuenta el tipo de parámetros de entrada que se necesitan en concreto para cada necesidad,

En segundo lugar, se han desarrollado las llamadas de gestión de perfiles. Estas llamadas contienen toda la lógica detrás de la administración de perfiles, creación y autenticación, para mayor seguridad y estabilidad de la página, se ha establecido un token que se genera de forma automática al iniciar sesión. Este token tiene como objetivo asegurar constantemente la autenticidad del usuario y evitar así posibles accesos o modificaciones de un usuario no con intenciones maliciosas.

Por último, se han desarrollado las llamadas de gestión de los conjuntos de datos. Estas tienen una estructura similar a la estructura de las llamadas de usuario, ya que accede a la base de datos. En concreto, esta se diferencia al trabajar con los archivos subidos por los usuarios y tratar con los datos.

5.4 INTERFAZ GRÁFICA

Para la interfaz gráfica se ha desarrollado en primera instancia un proyecto vacío mediante:

```
ng new aprendeML
```

Con el “*template*” vacío de la interfaz gráfica se ha implementado un router que facilita la navegación entre las diferentes vistas de la aplicación. En el archivo `app.routing.ts`,

se configuran las rutas de la aplicación, en cada una de las rutas se carga el módulo correspondiente a cada dirección. Este módulo se carga dentro de un componente con nombre FullComponent. Este componente es el contenedor principal de la aplicación. En él tenemos el “header” donde se puede ver el logo de la aplicación, tu usuario y permite el acceso a otro componente con nombre app-sidebar. Este componente se trata de una barra lateral que incluye todas las rutas y permite navegar a través de la página.

Con el componente principal implementado, el siguiente paso a seguir fue implementar la API cliente que hacía las llamadas al servidor Flask del servidor. Con esta API implementada, el siguiente paso a seguir es crear la autenticación de usuarios. Para esto se ha usado un servicio de auth, que

mediante una clase de usuario, nos permite alojar temporalmente el usuario autenticado.

Una vez el usuario ya existe en nuestra instancia, se puede crear el UserGuard que su objetivo principal es prevenir que usuarios sin registrar accedan a rutas exclusivas de usuarios registrados.

La implementación, al llegar a este punto, ha restado de implementar cada uno de los módulos accesibles desde las rutas. En especial, cada módulo hace llamadas a la API para gestionar los datos relacionados con su funcionalidad. Para los módulos donde ha sido necesario visualizar los gráficos interactivos, se ha necesitado crear un componente app-plot que, dependiendo de cada implementación, se iba adaptando para ajustarse a ese algoritmo. También cabe destacar la implementación de ventanas emergentes mediante diálogos que han tenido la función de subir y editar conjuntos de datos, y de editar el perfil.

5.5 SUBIDA A PRODUCCIÓN

Docker es una plataforma de virtualización a nivel de aplicación que permite empaquetar y distribuir aplicaciones junto con sus dependencias en forma de contenedores. Un contenedor Docker es una unidad ligera e independiente que contiene todo lo necesario para ejecutar una aplicación, incluido el código, las

bibliotecas y las configuraciones. Esta tecnología ofrece una manera eficiente y reproducible de distribuir y ejecutar aplicaciones, eliminando problemas de incompatibilidad y asegurando una alta portabilidad entre diferentes entornos.

Para generar estos contenedores, primero definimos un Dockerfile, que es simplemente un archivo de texto simple con un conjunto de comandos o instrucciones. Estas órdenes/instrucciones se ejecutan sucesivamente para realizar acciones sobre la imagen base y crear una nueva imagen de Docker. Haciendo un *“build”* de este *“Dockerfile”*, aún no hemos generado nuestro contenedor, sino que tenemos una imagen de este; finalmente, para crear el contenedor, deberemos utilizar el comando *“run”* en la imagen. Para esta implementación se han generado dos *“Dockerfiles”*, que representan las dos partes de nuestro proyecto: una imagen para la interfaz gráfica y una imagen para el servidor.

```
LABEL authors="Oriol"
FROM python:3.9
RUN mkdir app
WORKDIR /app
RUN pip install --upgrade pip
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt
COPY . /app
EXPOSE 3001
CMD python runserver 0.0.0.0:3001
```

El *“Dockerfile”* del servidor comienza estableciendo la imagen base como Python 3.9. A continuación, se crea un directorio llamado *“app”* dentro del contenedor y se establece como el directorio de trabajo actual. Para asegurarse de tener la versión más reciente de pip, se ejecuta el comando `pip install --upgrade pip`. Luego, se copia el archivo `requirements.txt` al directorio de trabajo del contenedor y se instalan las dependencias necesarias para el proyecto con el comando `pip install --no-cache-dir -r requirements.txt`.

Posteriormente, se copia todo el contenido del servidor al directorio de trabajo del contenedor; este paso es importante para que el servidor pueda funcionar correctamente dentro del contenedor. Se abre el puerto 3001 para permitir conexiones al servidor de Flask, lo que permite que el servidor sea accesible desde el exterior. Finalmente, se ejecutan los siguientes comandos para realizar las migraciones,

establecer la base de datos y poner en marcha el servidor: `python runserver 0.0.0.0:3001`.

Este *"Dockerfile"* configurado para la parte de la interfaz gráfica comienza estableciendo una imagen que proporciona un entorno de trabajo Node.js preconfigurado. Esta imagen ya incluye todo lo necesario para ejecutar aplicaciones Node.js. Como hicimos en el *"Dockerfile"* anterior, creamos un directorio *"app"* dentro de nuestro contenedor; este será el directorio de trabajo para todos los pasos siguientes.

Los archivos *"package.json"* y *"package-lock.json"* del proyecto se copian al directorio de trabajo del contenedor. Estos archivos contienen información sobre las dependencias y los scripts necesarios para que la aplicación funcione correctamente.

Una vez copiados los archivos, se ejecuta el comando `"npm install"` para instalar las dependencias del proyecto. Esto asegura que todas las bibliotecas requeridas estén disponibles para la aplicación. A continuación, se copia todo el contenido de la aplicación, incluidos los archivos de código fuente, al directorio de trabajo del contenedor. Luego, se ejecuta el comando `"npm run build"` para construir la aplicación y generar los archivos de compilación. Finalmente, se establece el comando `"npm start"` como comando de inicio para el contenedor, indicando al contenedor que inicie la aplicación utilizando el servidor web proporcionado por la aplicación.

```
FROM node:16
RUN mkdir app
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . /app
RUN npm run build
CMD ["npm", "start"]
```

6. PRUEBA DE RESULTADOS

Con la intención de analizar el rendimiento y la utilidad de la página web, se ha procedido a hacer una prueba de usuario con otros alumnos y exalumnos del grado de ingeniería informática. Esta prueba ha consistido en proporcionar acceso a la página web a estos usuarios durante una media de 30 minutos aproximadamente cada uno. Posteriormente a esta prueba se ha hecho un cuestionario sobre el rendimiento, utilidad, experiencia de usuario y con preguntas sobre las posibles mejoras.

Este formulario, que ha sido rellenado por 6 alumnos, ha arrojado distintos resultados a los esperados, pero bastante satisfactorios.

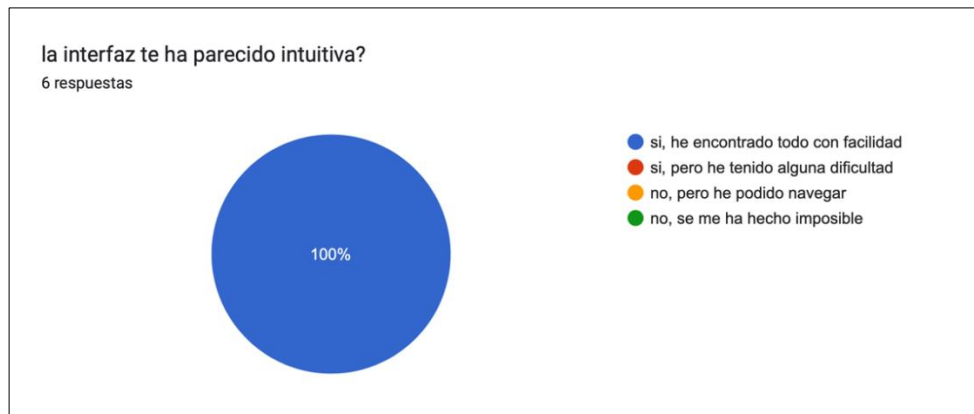


Figura 29: Pregunta 1 cuestionario

En primer lugar, se puede apreciar que la interfaz ha sido intuitiva y ha cumplido bien su función de ser llamativa.

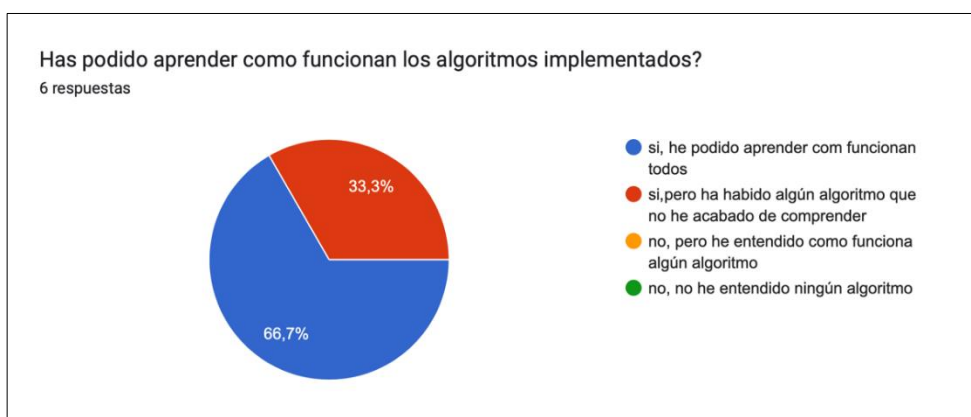


Figura 30 Pregunta 2 cuestionario

En segundo lugar, tenemos que todos los usuarios han podido entender en su mayoría o por completo el funcionamiento de los algoritmos implementados.

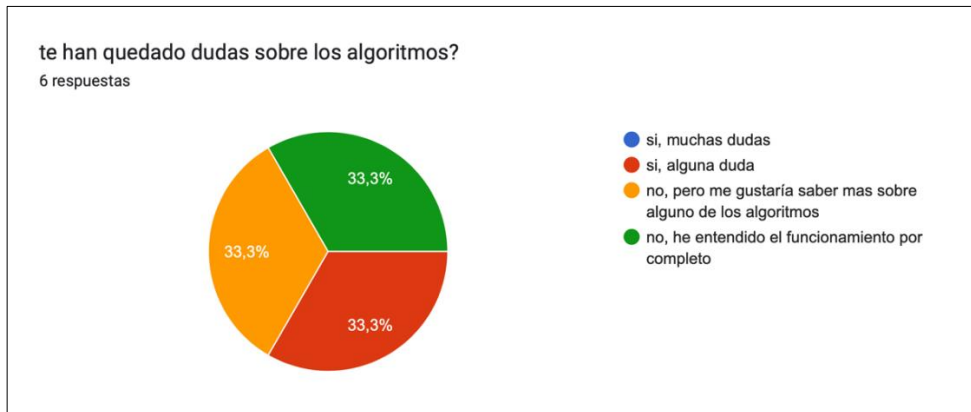


Figura 31 Pregunta 3 cuestionario

También podemos apreciar que la mayoría no tienen dudas, aunque se puede apreciar que ha habido dos usuarios que han experimentado alguna dificultad con alguno de los algoritmos.

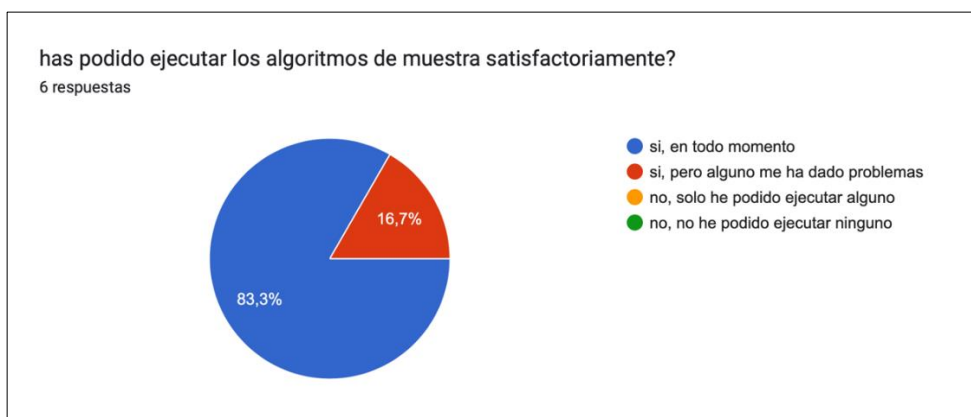


Figura 32 Pregunta 4 cuestionario

Este gráfico nos indica que, en su totalidad, los usuarios han podido ejecutar todos los algoritmos donde se usaban datos generados, a excepción de un usuario que ha experimentado dificultades con uno de ellos.

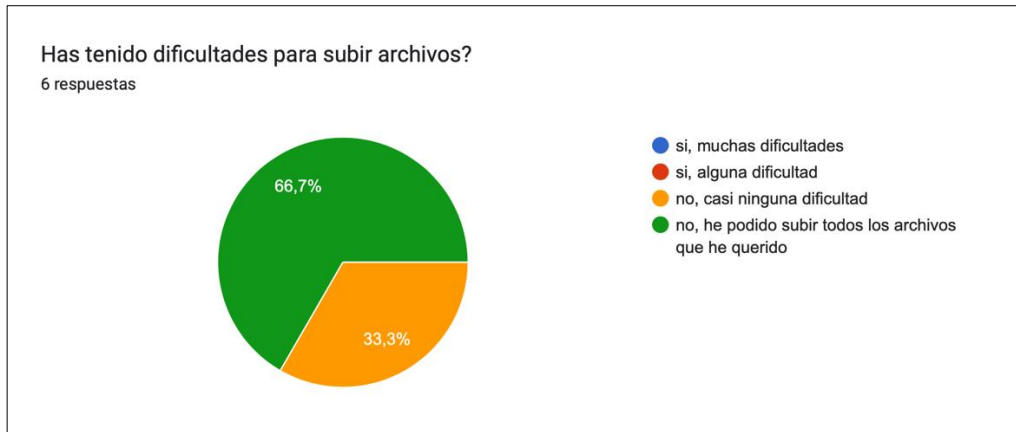


Figura 33 Pregunta 5 cuestionario

Las siguientes respuestas nos indican que el sistema de subida de conjuntos de datos funciona correctamente, aunque se podría mejorar algún aspecto para que fuese más estable.

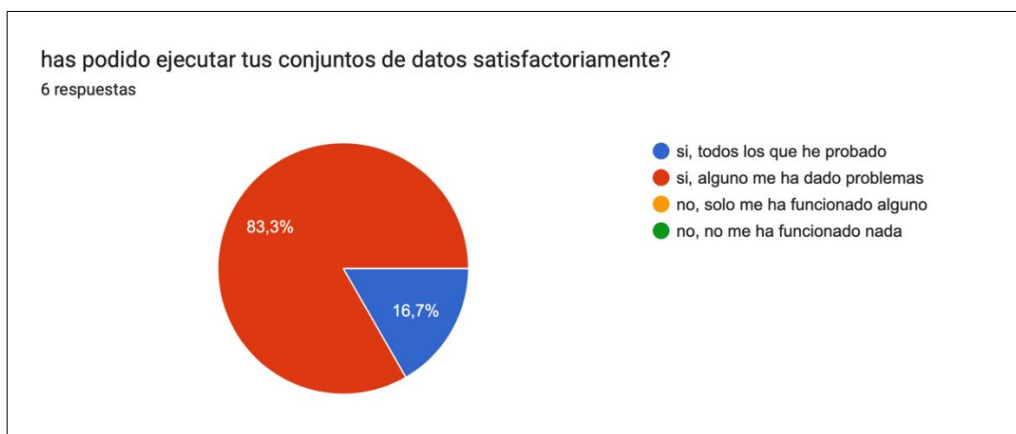


Figura 34 Pregunta 6 cuestionario

En el ámbito de la ejecución de conjuntos de datos subidos por el usuario podemos apreciar que la mayoría ha experimentado problemas al intentar ejecutar alguno de sus conjuntos, por lo tanto, es un punto que se debería mejorar de cara a una segunda iteración a futuro, darles más estabilidad y robustez a los conjuntos de datos subidos por el usuario.

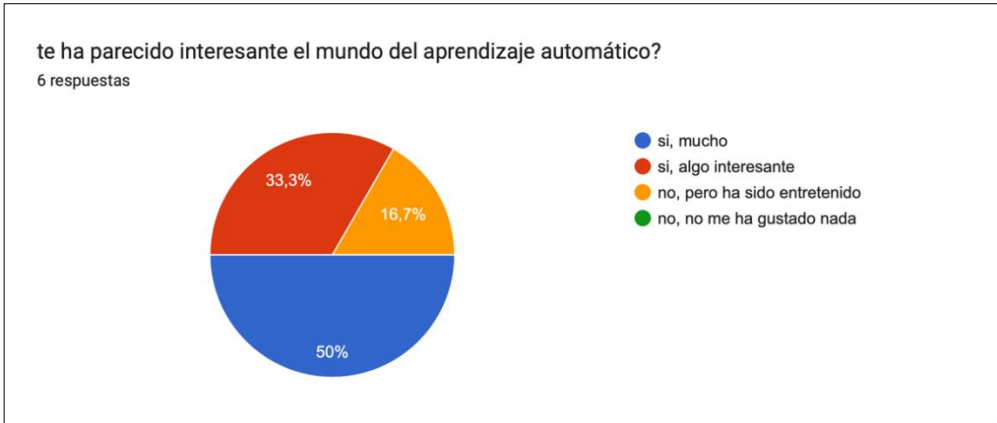


Figura 35 Pregunta 7 cuestionario

En último lugar, tenemos que la mayoría de los usuarios ha despertado interés en el mundo del aprendizaje automático, lo cual indica que la aplicación web cumple en gran medida con el objetivo inicial del proyecto: dar a conocer e incitar a los usuarios a adentrarse en este maravilloso mundo.

7. CONCLUSIONES Y POSIBLES MEJORAS

El propósito central de este proyecto consistió en concebir y desarrollar una aplicación web de manera integral, abarcando desde la definición de la arquitectura hasta la implementación final y el despliegue de la aplicación. En este sentido, me enorgullece haber alcanzado exitosamente todas las metas que me propuse al inicio del proyecto.

En lo que respecta al desarrollo del servidor, se estableció una REST API que destaca por su escalabilidad y un diseño inicial orientado a la coherencia y seguridad. La aplicación de técnicas adquiridas en otras asignaturas garantizó la solidez y eficiencia del sistema. La arquitectura se diseñó teniendo en cuenta la capacidad de escalar fácilmente y la simplicidad en el mantenimiento para futuros desarrollos.

En el ámbito de la interfaz gráfica, se ideó una interfaz moderna e intuitiva que refleja fielmente el diseño inicial, incluso mejorándolo mediante la adición de parámetros adicionales y una definición más precisa, en comparación con los wireframes iniciales.

Se lograron satisfactoriamente prácticamente todos los objetivos y requisitos establecidos en las etapas iniciales del proyecto, indicando que la planificación inicial fue acertada.

En conclusión, este proyecto ha sido una experiencia sumamente interesante y enriquecedora, especialmente en el campo del desarrollo web. Resulta gratificante aplicar con éxito la teoría y los conocimientos adquiridos previamente en otras asignaturas, consolidando así una experiencia educativa integral.

En un futuro, me gustaría mejorar el proyecto en diversas áreas para ofrecer una experiencia aún más satisfactoria a los usuarios. En primer lugar, se consideraría la posibilidad de ampliar la variedad de algoritmos disponibles. Esto permitiría brindar a los usuarios un abanico más extenso de conocimientos en el ámbito de la ingeniería informática.

Otra área de mejora potencial se centra en optimizar el flujo de ejecución de los algoritmos, especialmente al trabajar con datos impredecibles subidos por los usuarios. La intención sería proporcionar una experiencia más fluida y consistente durante este proceso.

Siendo consciente del interés de los usuarios y las tendencias tecnológicas emergentes, se contemplaría explorar el "*Deep Learning*" como una adición estratégica al proyecto. Esto aseguraría que la plataforma se mantenga actualizada y alineada con las últimas innovaciones en el campo.

Una atención especial se dirigiría a mejorar la estabilidad al ejecutar conjuntos de datos subidos por los usuarios. Mediante una revisión detallada, se buscaría fortalecer la robustez del sistema y garantizar una ejecución más eficiente para evitar posibles inconvenientes.

En cuanto al sistema de subida de conjuntos de datos, aunque se ha indicado que funciona correctamente, se podría considerar la optimización de la interfaz de usuario y la implementación de medidas preventivas para hacerlo más intuitivo y estable.

9. BIBLIOGRAFÍA

- [1] Linear Regression. (n.d.). Retrieved October 27, 2023, from <https://mlu-explain.github.io/linear-regression/> .
- [2] Pro, P. (22 de 07 de 2023). Why data preparation is an important part of datascience? Obtenido de <https://www.projectpro.io/article/why-data-preparation-is-an-important-part-of-data-science/242>
- [3] Rojo, O. (21 de 07 de 2023). Machine Learning Algorithms With Scikit-Learn. Obtenido de <https://medium.com/swlh/machine-learning-algorithms-with-scikit-learn-ec6181365ba>
- [4] Tavasoli, S. (11 de 08 de 2023). Top 10 Machine Learning Algorithms. Obtenido de <https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article>
- [5] S. L. David W. Hosmer, Applied Logistic Regression. Wiley-Interscience Publication,2000.
- [6] O. Kramer, K-Nearest Neighbors. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013,pp. 13–23. [En línea]. Disponible en: https://doi.org/10.1007/978-3-642-38652-7_2
- [7] L. M. Surhone, Naive Bayes Classifier. Betascript Publishers, 2010.
- [8] Orza, P. (26 de 07 de 2022). MLaaS Platforms: The Comparative Guide. Obtenidode <https://levity.ai/blog/mlaas-platforms-comparative-guide>
- [9] IBM Topics. IBM. Obtenido de <https://www.ibm.com/topics>
- [10] UI-Grid. UI-Grid. Obtenido de <http://ui-grid.info/> (Consultado en octubre de 2023)
- [11] Linear Regression Explained. (n.d.). MLU-Explain. Obtenido de <https://mlu-explain.github.io/linear-regression/> (Visitado en septiembre de 2023)

[12] BigML. BigML. Obtenido de <https://bigml.com/> (Accedido en diciembre de 2023)

[13] NG Bootstrap. NG Bootstrap. Obtenido de <https://ng-bootstrap.github.io/#/home> (Revisado en noviembre de 2023)

[14] Seeing Theory. Seeing Theory. Obtenido de <https://seeing-theory.brown.edu/> (Explorado en septiembre de 2023)

[15] Angular Material Components. Angular. Obtenido de <https://material.angular.io/components> (Verificado en octubre de 2023)

[16] Aprende Machine Learning. Aprende Machine Learning. Obtenido de <https://www.aprendemachinelearning.com/> (Consultado en noviembre de 2023)

[17] Hack, S. Machine Learning – 4 Books in 1. (Revisado en diciembre de 2023)

[18] Scikit-Learn User Guide. Scikit-Learn. Obtenido de https://scikit-learn.org/stable/user_guide.html (Explorado en septiembre de 2023)

[19] Flask Advanced Documentation. FastAPI. Obtenido de <https://flask.palletsprojects.com/en/3.0.x/> (Accedido en noviembre de 2023)

[20] K Nearest Neighbors Explained. Towards Data Science. Obtenido de <https://towardsdatascience.com/k-nearest-neighbors-k-nn-explained-8959f97a8632> (Visitado en diciembre de 2023)

[21] Introduction to Linear Regression. Towards Data Science. Obtenido de <https://towardsdatascience.com/linear-regression-5100fe32993a> (Consultado en octubre de 2023)

[22] Introduction to Logistic Regression. Towards Data Science. Obtenido de <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148> (Revisado en septiembre de 2023)

[23] A Step-by-Step Introduction to PCA. Towards Data Science. Obtenido de <https://towardsdatascience.com/a-step-by-step-introduction-to-pca-c0d78e26a0dd>
(Explorado en noviembre de 2023)

[24] Angular Documentation. Angular. Obtenido de <https://angular.io/docs>
(Verificado en diciembre de 2023)

[25] Orange data mining. . Obtenido de <https://orangedatamining.com/>
(Verificado en diciembre de 2023)

[26] MIPlaygrounds. . Obtenido de <https://mlplaygrounds.com/> (Verificado en diciembre de 2023)