



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Herramienta docente para la  
visualización en Web de  
algoritmos de aprendizaje  
Semi-Supervisado**



Presentado por David Martínez Acha  
en Universidad de Burgos — 6 de febrero  
de 2023

Tutor: Álgvar Arnaiz González  
Cotutor: César Ignacio García Osorio







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Álvar Arnaiz González, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas informáticos, junto a D. César Ignacio García Osorio, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas informáticos.

Exponen:

Que el alumno D. David Martínez Acha, con DNI 71310644H, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Herramienta docente para la visualización en Web de algoritmos de aprendizaje Semi-Supervisado.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 6 de febrero de 2023

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Álvar Arnaiz González

César Ignacio García Osorio





## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

aprendizaje automático, aprendizaje semi-supervisado, visualización de algoritmos, web

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

machine learning, semi-supervised learning, algorithm visualization, web



---

# Índice general

---

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Aprendizaje automático . . . . .	5
Técnicas y herramientas	11
4.1. Secciones . . . . .	11
4.2. Referencias . . . . .	12
4.3. Imágenes . . . . .	12
4.4. Listas de items . . . . .	12
4.5. Tablas . . . . .	13
Aspectos relevantes del desarrollo del proyecto	15
Trabajos relacionados	17
6.1. Visualizadores . . . . .	17
6.2. Otras herramientas/librerías . . . . .	20
Conclusiones y Líneas de trabajo futuras	23

**Bibliografía**

**25**

---

## Índice de figuras

---

3.1. Clasificación de aprendizaje automático [11]. . . . .	6
3.2. Funcionamiento general del aprendizaje supervisado [3]. . . . .	7
3.3. Clusters . . . . .	9
4.1. Autómata para una expresión vacía . . . . .	12

---

# Índice de tablas

---

4.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	13
6.1. Características que influyen los objetivos pedagógicos . . . .	20

---

# Introducción

---

Semi-Supervisado



---

## Objetivos del proyecto

---

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.





---

## Conceptos teóricos

---

### 3.1. Aprendizaje automático

Según [5], el aprendizaje automático (*machine learning*) es una rama de la Inteligencia artificial como una técnica de análisis de datos que enseña a las computadoras a aprender de la **experiencia** (es decir, lo que realizan los humanos). Para ello, el aprendizaje automático se nutre de gran cantidad de datos (o los suficientes para el problema concreto) que son procesados por ciertos algoritmos. Estos datos son ejemplos (también llamados instancias o prototipos), [12] mediante los cuales, los algoritmos son capaces de generalizar comportamientos que se encuentran ocultos.

La característica principal de estos algoritmos es que son capaces de mejorar su rendimiento de forma automática basándose en procesos de entrenamiento y también en las fases posteriores de explotación. Debido a sus propiedades, el aprendizaje automático se ha convertido en un campo de alta importancia, aplicándose a multitud de campos como medicina, automoción, visión artificial... Los tipos de aprendizaje automático se suelen clasificar en los siguientes: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Sin embargo, aparece una nueva disciplina que se encuentra a caballo entre el supervisado y no supervisado (utiliza tanto datos etiquetados como no etiquetados para el entrenamiento) [13].

En la figura 3.1 se puede ver una clasificación de aprendizaje automático.



Figura 3.1: Clasificación de aprendizaje automático [11].

## Aprendizaje supervisado

El aprendizaje supervisado es una de las aproximaciones del aprendizaje automático. Los algoritmos de aprendizaje supervisado son entrenados con datos que han sido etiquetados para una salida concreta [9]. Por ejemplo, dadas unas biopsias de pacientes, una posible etiqueta es si padecen de cáncer o no. Estos datos tienen una serie de características (e.g. en el caso de una biopsia se tendría la edad, tamaño tumoral, si ha tenido lugar mitosis o no...) y todas ellas pueden ser binarias, categóricas o continuas [3].

Es común que antes del entrenamiento, estos datos son particionados en: conjunto de entrenamiento, conjunto de test o conjunto de validación. De forma resumida, el conjunto de entrenamiento serán los datos que utilice el propio algoritmo para aprender y generalizar los comportamientos ocultos de los mismos. El conjunto de validación se utilizará para tener un control de que el modelo está generalizando y no sobreajustando (memorizando los datos) y por último, el conjunto de test sirve para estimar el rendimiento real que podrá tener el modelo en explotación [14]. En la figura 3.2 puede visualizarse el funcionamiento general.

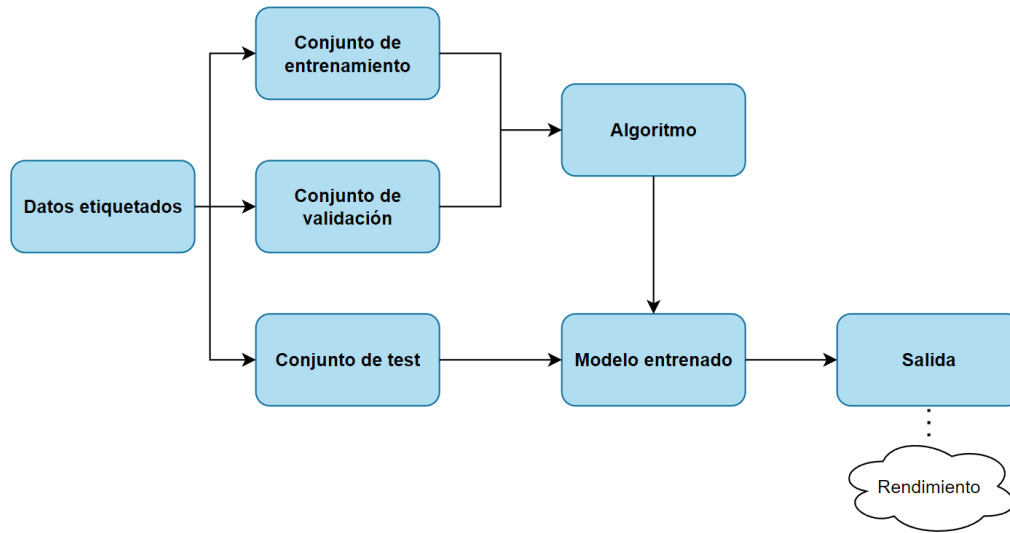


Figura 3.2: Funcionamiento general del aprendizaje supervisado [3].

El aprendizaje supervisado está altamente influenciado por esto. Por un lado, si las etiquetas son categóricas o binarias el modelo será de **clasificación** y por otro, si las etiquetas son continuas el modelo será de **regresión**.

- **Clasificación:** Los algoritmos de clasificación, a veces denominados simplemente como clasificadores, tratan de predecir la clase de una nueva entrada a partir del entrenamiento previo realizado. Estas clases son discretas y en clasificación pueden referirse a clases (o etiquetas) binarias o clases múltiples.
- **Regresión:** En este caso, el algoritmo asigna un valor continuo a una entrada. Es decir, trata de encontrar una función continua basándose en las variables de entrada. Se denomina también ajuste de funciones.

## Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, en el no supervisado, los algoritmos no se nutren de datos etiquetados. En otras palabras, los usuarios no «supervisan» el modelo [4]. Esto quiere decir que no aprenderán de etiquetas, sino de la propia estructura que se encuentre en los datos (patrones). Por ejemplo, dadas unas imágenes de animales, sin especificar cuál es cuál, el aprendizaje no supervisado identificará las similitudes entre imágenes y como resultado podría dar la separación de las especies (o separaciones entre colores, pelaje, raza...).

Como principales usos del aprendizaje no supervisado, suele aplicarse a:

1. **Agrupamiento (Clustering):** Este modelo de aprendizaje no supervisado trata de dividir los datos en grupos. Para ello, estudia las similitudes entre ellos y también en las disimilitudes con otros. Estos modelos pueden tanto descubrir por ellos mismos los «clústeres» o grupos que se encuentran o indicarle cuántos debe identificar [4].
2. **Reducción de la dimensionalidad:** Para empezar, el término «dimensionalidad» hace referencia al número de variables de entrada que tienen los datos. En la realidad, los conjuntos de datos sobre los que se trabaja suelen tener una dimensionalidad grande. Según [6] la reducción de dimensionalidad se denomina como «Una forma de convertir conjuntos de datos de alta dimensionalidad en conjunto de datos de menor dimensionalidad, pero garantizando que proporciona información similar». Es decir, simplificar el problema pero sin perder toda esa estructura interesante de los datos. Algunos ejemplos pueden ser:
  - Análisis de Componentes Principales (PCA)
  - Cuantificación vectorial
  - Autoencoders

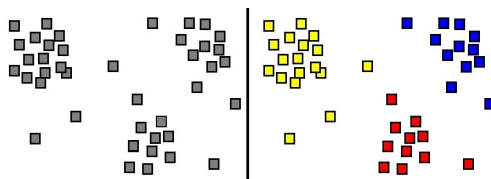


Figura 3.3: Clusters - By hellisp - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=36929773>. Ejemplo de clustering, a la izquierda los datos no etiquetados y a la derecha los datos coloreados según las clases identificadas por el algoritmo de clustering.

## Aprendizaje semi-supervisado

Según [13] el aprendizaje semi-supervisado es la rama del aprendizaje automático referido al uso de datos tanto etiquetados como no etiquetados simultáneamente para realizar tareas de aprendizaje. Se encuentra a caballo entre el aprendizaje supervisado y no supervisado. Concretamente, los problemas donde más se aplica, y donde más investigación se realiza es en clasificación. Los métodos semi-supervisados resultan especialmente útil cuando se tienen escasos datos etiquetados, que, aparte de ser una situación común en problemas reales, el proceso de etiquetado es una labor compleja, que consume tiempo y es costosa.

### Suposiciones

El objetivo de usar datos no etiquetados es construir un clasificador que sea mejor que el aprendizaje supervisado, en el que solo se tienen datos etiquetados. Pero para que el aprendizaje semi-supervisado mejore a lo ya existente, tiene una serie de suposiciones que han de cumplirse.

En primera instancia se dice que la condición necesaria es que la distribución  $p(x)$  del espacio de entrada contiene información sobre la distribución posterior  $p(y/x)$  [13].

Pero la forma en el que interactúan los datos de una distribución y la posterior, no siempre es la misma:

#### *Smoothness assumption*

Esta suposición indica que si dos ejemplos (o instancias) de la entrada están cerca en ese espacio de entrada, entonces, probablemente, sus etiquetas sean las mismas.

*Low-density assumption*

Esta suposición indica que en clasificación, los límites de decisión deben encontrarse en zonas en las que haya pocos de estos ejemplos (o instancias).

*Manifold assumption*

Los datos pueden tener una dimensionalidad alta (muchas características) pero generalmente no todas las características son completamente útiles. Los datos a menudo se encuentran en unas estructuras de más baja dimensionalidad. Estas estructuras se conocen como «manifolds». Esta suposición indica que si los datos del espacio de entrada se encuentran en estas «manifolds» entonces aquellos puntos que se encuentren en el mismo «manifolds» tendrán la misma etiqueta. [7, 13]

*Cluster assumption*

Como generalización de las anteriores, aquellos datos que se encuentren en un mismo clúster tendrán la misma etiqueta.

De estas suposiciones se extrae el concepto de «similitud» en el que en todas ellas se encuentra presente. Y en realidad, todas son versiones de *Cluster assumption* en la que los puntos similares tienden a pertenecer al mismo grupo. Además, la suposición de clúster resulta necesaria para que el aprendizaje semi-supervisado mejore al supervisado. Si los datos no pueden ser agrupados, entonces no mejorará ningún método supervisado [13].

---

# Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

Las secciones se incluyen con el comando `section`.

## 4.1. Secciones

### Subsecciones

Además de secciones tenemos subsecciones.

### Subsubsecciones

Y subsecciones.

## 4.2. Referencias

## 4.3. Imágenes

Se pueden incluir imágenes con los comandos standard de  $\text{\LaTeX}$ , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 4.1: Autómata para una expresión vacía

## 4.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

**Primer item** más información sobre el primer item.



Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 4.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

**Segundo item** más información sobre el segundo item.

■

## 4.5. Tablas

Igualmente se pueden usar los comandos específicos de  $\text{\LaTeX}$  o bien usar alguno de los comandos de la plantilla.



---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.



---

# Trabajos relacionados

---

Con el aprendizaje automático y el uso de gran cantidad de datos, es completamente necesaria la visualización de los datos, procesos de entrenamiento y ciertas estadísticas. Al fin y al cabo, cuando se construyen modelos, se debe tener una realimentación de cómo de bien está funcionando para poder extraer conclusiones sobre el mismo.

De forma general, sin centrarse directamente en el aprendizaje automático, resulta interesante y conveniente que los visualizadores de algoritmos sean accesibles y que, como están apareciendo, se creen aplicaciones Web que resultan mucho más directas. Desde el punto de vista de la docencia y aprendizaje los visualizadores permiten culminar la comprensión los aspectos teóricos subyacentes.

## 6.1. Visualizadores

### Seshat

Es una herramienta web que trata de facilitar el aprendizaje de la teoría de lenguajes y autómatas (análisis léxico) que utilizan los compiladores [2], puede accederse desde <http://cgosorio.es/Seshat>. La herramienta propone en primera instancia unas explicaciones teóricas de los algoritmos con sus conceptos generales, el concepto concreto de las expresiones regulares y qué es un autómata finito. A partir de la teoría, se encuentran implementados varios algoritmos que se visualizan paso a paso. En el momento de la ejecución también se tienen elementos de interés como la propia descripción o explicación del algoritmo. Los algoritmos implementados por la herramienta son:

1. Construcción de un autómata finito no determinista (AFND) a partir de una expresión regular.
2. Conversión de un autómata finito no determinista (AFND) a un autómata finito determinista (AFD).
3. Construcción de un autómata finito determinista (AFD) a partir de una expresión regular.
4. Minimización de un autómata finito.

Para su construcción se ha usado el framework Flask en Python que actúa como servidor. La interfaz de usuario está construida con HTML, SVGs y Javascript para proporcionar el contenido dinámico.

En este proyecto se realizó un estudio de la opinión de 42 estudiantes después de su uso. El estudio consistía en unas afirmaciones (5) respecto a la buena utilidad, buen diseño o el interés que puede producir al visualizar los algoritmos de esa forma. De forma general, en una escala del 1 al 5 (donde el 5 representa que están profundamente de acuerdo con la afirmación) el 95 % de los resultados se concentran entre el 4 y el 5 en la valoración. Esto indicó que la herramienta sí que resultó útil e incluso ellos mismos solicitaban este recurso para facilitar su aprendizaje.

### **Herramienta de apoyo a la docencia de algoritmos de selección de instancias**

[1]. Es una herramienta de escritorio para la visualización de la ejecución y resultados de los algoritmos de selección de instancias debido a la carencia de este tipo de aplicaciones para dichos algoritmos. La herramienta es altamente personalizable pudiendo subir el conjunto de datos o seleccionar qué característica visualizar en los ejes. En la visualización se pueden ver todos los pasos de los algoritmos junto, por ejemplo, a las regiones de Boronoi o el pseudocódigo. Esto hace que el alumno pueda conocer el progreso y los conceptos particulares (influencia, vecindad...). Los algoritmos implementados por la herramienta son:

1. Algoritmo Condensado de Hart (CNN)
2. Algoritmo Condensado Reducido (RNN).
3. Algoritmo Subconjunto Selectivo Modificado (MSS)
4. Algoritmos Incremental Reduction Optimization Procedure (DROP)

5. Algoritmo Iterative Case Filtering (ICF)
6. Algoritmo Democratic Instance Selection (DIS)

Está desarrollado completamente en Java lo que lo hace portable a cualquier plataforma y sin instalación.

Como punto característico a tener en cuenta, la herramienta estaba muy orientada a ofrecer bastante granularidad en cada paso, de tal forma que mediante, por ejemplo, la visualización del pseudocódigo (al mismo tiempo que la visualización gráfica), permite para analizar el comportamiento subyacente.

Los estudiantes, de forma general, valoraron muy positivamente la herramienta pues les ayudó a comprender los algoritmos.

### **Towards Developing an Effective Algorithm Visualization Tool for Online Learning**

Este artículo resulta especialmente interesante de cara a las consideraciones a la hora del desarrollo de una buena visualización.

La realidad actual es que resulta muy difícil hacer que los alumnos entiendan todos los conceptos que involucran los algoritmos y más aún en la cada vez más común modalidad «a distancia». Para solventar este problema se haya desarrollado múltiples herramientas de visualización de algoritmos (AV) que pretenden facilitar la comprensión de los mismos, pero la realidad de estas herramientas es que se desarrollan con una fuerte componente «bonita» y no enriquecedora o pedagógica. En este artículo se trata de abordar este último problema, el cómo construir herramientas de visualización de algoritmos que sean divertidas y atractivas, pero siendo fieles a los conceptos y cómo estas pueden ayudar a los docentes a asegurar el aprendizaje de sus alumnos [10].

Para lograr el objetivo de desarrollar un visualizador efectivo, se analiza desde el punto de vista de la pedagogía, usabilidad y accesibilidad. En la Tabla 6.1 se pueden algunos de las características que se deben tener en cuenta a la hora de desarrollar una herramienta de este estilo.

### **Interactive System for Algorithm and Data Structure Visualization**

En la línea del anterior artículo en [8] se presenta un proyecto en desarrollo de una página abierta para la visualización de algoritmos interactivos.

<b>Incrementar comprensión</b>	Explicaciones textuales Realimentación con las acciones del usuario Ayuda integrada Comodidad en la entrada de datos
<b>Promocionar interés</b>	Permitir introducir datos al algoritmo Manipular la visualización Capacidad de volver "<rebobinar"> Variación de la velocidad Avanzar en la ejecución
<b>Aprendibilidad</b>	Controles familiares Generalidad con otros sistemas Predictibilidad de las acciones Interfaz simple
<b>Memorabilidad</b>	Interfaz común que soporte múltiples animaciones
<b>Facilidad de uso</b>	Interfaz común que soporte múltiples animaciones Claridad de los modelos y conceptos
<b>Robustez</b>	Explicaciones de la visualización Integración de ajustes predefinidos Recuperación de errores
<b>Eficacia</b>	Visibilidad del estado del sistema Control y libertad del usuario Prevención de errores Flexibilidad de uso Ayuda al usuario para reconocer, diagnosticar y recuperarse de errores
<b>Accesibilidad</b>	Globalmente disponible Plataforma y dispositivo independiente

Tabla 6.1: Características que influyen los objetivos pedagógicos

En el que precisamente la «interactividad» busca conseguir esos objetivos anteriores (pedagogía, usabilidad y accesibilidad).

## 6.2. Otras herramientas/librerías

Algorithm-Visualizer, <https://algorithm-visualizer.org/>. Se trata de una Web que incluye una cantidad enorme de algoritmos, todos están programados directamente en Javascript que genera una visualización de los mismos y también incluye una explicación corta de cada uno. Algunos de los algoritmos son: Backtracking, Divide and Conquer, Greedy o más concretamente K-Means Clustering, entre muchos otros.

VISUALGO, <https://visualgo.net/en>. Es una página web creada por la Universidad Nacional de Singapur. Esta Web permite visualizar algoritmos básico (como de ordenación o camino más corto) pero también estructuras de datos (como «hash tables» o conjuntos disjuntos).



Visualizer, por Zhenbang Feng, <https://jasonfenggit.github.io/Visualizer/> (Github: <https://github.com/JasonFengGit/Visualizer>). Su autor lo define como una página web para proporcionar visualizaciones intuitivas e innovadoras de algoritmos generales y de inteligencia artificial. Las visualizaciones son muy visuales con las que se capta muy bien el funcionamiento de los algoritmos. Implementa algoritmos de búsqueda de caminos, ordenamiento e inteligencia artificial (como minimax).

Clustering-Visualizer, <https://clustering-visualizer.web.app/>. Es una página Web para visualizar algoritmos de clustering. Realmente, no tiene muchos implementados, pero las visualizaciones son muy descriptivas.

MLDemos, puede visitarse desde <https://basilio.dev/>. Es una herramienta de visualización de código abierto para algoritmos de aprendizaje automático creada específicamente para ayudar en el estudio y comprensión del funcionamiento de una gran cantidad de algoritmos. Entre ellos: clasificación, regresión, reducción de dimensionalidad y clustering.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

## Bibliografía

---

- [1] Álvaro Arnáiz González, José Francisco Díez Pastor, César I García Osorio, and Juan José Rodríguez Díez. Herramienta de apoyo a la docencia de algoritmos de selección de instancias. 2012.
- [2] Álvaro Arnaiz-González, Jose-Francisco Díez-Pastor, Ismael Ramos-Pérez, and César García-Osorio. Seshat—a web-based educational resource for teaching the most common algorithms of lexical analysis. *Computer Applications in Engineering Education*, 26(6):2255–2265, 2018.
- [3] Salim Dridi. Supervised learning - a systematic literature review. *ResearchGate*, 09 2021.
- [4] Salim Dridi. Unsupervised learning - a systematic literature review. *ResearchGate*, 12 2021.
- [5] Intelligent. Machine learning: qué es y cómo funciona, 2020. [Internet; descargado 27-octubre-2022].
- [6] javaTpoint. Unsupervised machine learning. [Online; accessed 15-November-2022].
- [7] Lukas Huber. A friendly intro to semi-supervised learning. [Online; accessed 15-November-2022].
- [8] Patrik Perháč and Slavomír Šimoňák. Interactive system for algorithm and data structure visualization. *Computer Science Journal of Moldova*, 30(1), 2022.

- [9] David Petersson. Supervised learning, 2021. [Internet; descargado 15-noviembre-2022].
- [10] Katarzyna Romanowska, Gurpreet Singh, M. Ali Akber Dewan, and Fuhua Lin. Towards developing an effective algorithm visualization tool for online learning. pages 2011–2016, 2018.
- [11] Neova Tech Solutions. Machine learning algorithms: Beginners guide part 1, 2018. [Internet; descargado 27-octubre-2022].
- [12] Pascual Parada Torralba. ¿qué es el machine learning? aprendizaje supervisado vs no supervisado, 2022. [Internet; descargado 27-octubre-2022].
- [13] Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, Feb 2020.
- [14] Wikipedia contributors. Training, validation, and test data sets — Wikipedia, the free encyclopedia, 2022. [Online; accessed 15-November-2022].