Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores
Unidad Morelia

Manual de prácticas para la mejora de la enseñanza del aprendizaje máquina aplicado a la Ciencia de Datos a gran escala

INFORME FINAL

que para obtener el título de

Licenciada en Tecnologías para la Información en Ciencias

P R E S E N T A

Mariana Michell Flores Monroy

TUTOR

Dr. Sergio Rogelio Tinoco Martínez

CO-TUTOR

Dr. Heberto Ferreira Medina

Morelia, Michoacán Abril 2022

Agradecimientos institucionales

Le agradezco principalmente a la Escuela Nacional de Estudios Superiores Unidad Morelia, a los Institutos de investigación que forman parte de la UNAM campus Morelia y a la Universidad Nacional Autónoma de México, por haberme dado la oportunidad de adquirir las habilidades y conocimientos necesarios para desarrollarme en los ámbitos profesional, ético, académico y laboral.

Mi mayor y más profundo a todo el cuerpo docente de la Licenciatura en Tecnologías para la Información en ciencias, por su tiempo, su paciencia y por toas sus enseñanzas tanto dentro como fuera del ámbito académico, especialmente a la Dra. Marisol Flores Garrindo, Dra. Adriana Menchaca Méndez por ayudarme a no rendirme y siempre brindarme su apoyo en cada etapa de mi trayectoria universitaria.

Mi más sincero agradecimiento a las y los docentes y académicos que tan amables aceptaron participar en la mesa sinodal.

Agradezco a los siguientes proyectos que me ayudaron económicamente durante el desarrollo de este trabajo: al proyecto **PAPIME PE106021** ya que sin este apoyo no hubiera podido concluir de este.

Agradezco infinitamente al Dr. Sergio Rogelio Tinoco Martínez por su tiempo, comprensión, guía y paciencia durante este proyecto, por haber fungido como mi asesor y por sus observaciones.

Al Dr. Heberto Ferreria Medina por haber compartido su experiencia y conocimientos conmigo para poder aplicarlos en este proyecto, así como haber fungido como co-asesor en este trabajo.

Agradecimientos personales

Primero quiero agradecer a mi familia por el infinito sacrificio que hizo todos y cada uno de ellos para que yo pueda llegar hasta aquí. Agradezco a mi mamá y papá quienes aunque no entendían bien de qué trata mi carrera, no dejaron de creen en mí.

A mis hermanos Jersain, Abigail y Monse por motivarme a seguir adelante y cumplir mis objetivos.

A mis mascotas, que a pesar de no entender qué pasa me han motivado a ser mejor y a esforzarme cada día.

También agradezco a mis compañeros de clase que hicieron mi trayectoria universitaria más divertida y productiva de lo que podía esperar, pero en especial agradecer a Bruce que sin su amor y ayuda no hubiera podido concluir este proyecto.

Agradezco a mi amiga y compañera Ruth, que a pesar de haber tomado un camino diferente su influencia sigue presente conmigo.

A mi amigo de toda la vida, Eric quien siempre ha creído en mí y en mi potencial.

A mi asesor, el Dr. Sergio Tinoco y a mi cos-asesor, el Dr. Heberto Ferreria por facilitar todos los recursos necesarios para llevar a cabo este proyecto

Resumen

Este trabajo forma parte del proyecto PAPIME titulado “Propuesta de mejora a la enseñanza del aprendizaje automático aplicado a la Ciencia de Datos a gran escala” con el número de proyecto PE106021. Se propone la elaboración de un manual de prácticas para estudiantes del nivel licenciatura, que ayude a mejorar el conocimiento del Machine Learning (ML) y su aplicación en la ciencia de datos a gran escala (Big Data).

El proyecto estará basado en diseñar, construir e implementar una guía de prácticas dirigida a los estudiantes a partir de sexto semestre en adelante de la Licenciatura en Tecnologías para la Información en Ciencias o de otras licenciaturas de la ENES Morelia que cuenten con los conocimientos básicos del ML.

Para realizar dicho proyecto, se tomará en cuenta la opinión de alumnos y docentes de las diferentes licenciaturas dentro de la ENES Morelia, con respecto a cuales temas son los que consideran de mayor importancia y que se deben impartir dentro de las materias que utilizan el aprendizaje automático dentro del plan de estudios de la LTICS.

El fin de este proyecto es el de mejorar la formación académica de los estudiantes dentro de la LTICS así como mejorar la calidad de la enseñanza de este tema por parte de los docentes.

Abstract

This work is part of the PAPIME project called "Proposal to improve the teaching of machine learning applied to large-scale Data Science" with PE106021 project number. The development of a practices manual for undergraduate students is proposed to help to improve knowledge of Machine Learning (ML) and its application in large-scale data science (Big Data).

The project is designing, building and implementing a practices guide aimed at students from the sixth semester onwards of the Bachelor of Information Technology in Sciences and/or other degrees from ENES Morelia whose have the basic knowledge of the ML.

To make this project, the opinion of students and teachers of the different degrees within the ENES Morelia will be taken into account, with respect to which topics are the ones that they consider most important and that should be taught within the subjects that use learning. automatic within the LTICS curriculum.

The purpose of this project is to improve the academic training of students within the LTICS as well as improve the quality of teaching of this subject by teachers.

Índice general

Índice de figuras

Capítulo 1

Introducción

El uso de herramientas que permiten el análisis de grandes volúmenes de datos ha permitido que las ciencias exactas jueguen un papel importante para la toma de decisiones en las organizaciones. En la Licenciatura en Tecnologías de la Información para las Ciencias (TICs) de la Escuela Nacional de Estudios Superiores (ENES) Morelia, México, existen asignaturas relacionadas con la Ciencia de Datos que se incluyen en el plan de estudios a partir del sexto semestre, conocidas como asignaturas del área de profundización y que representan un reto para los estudiantes a la hora de tratando de poner en práctica la teoría aprendida, además de carecer de las herramientas para su aplicación en problemas reales.

Se observa la necesidad de que docentes y estudiantes conozcan nuevas fronteras en Inteligencia Artificial (IA), específicamente en la aplicación de modelos matemáticos de Machine Learning (ML).

ML es la rama de la IA que se encarga de desarrollar técnicas, algoritmos y programas que brindan a las computadoras la capacidad de aprender. Una máquina aprende cada vez que cambia su estructura, programas o datos, en función de la entrada o en respuesta a información externa, de tal manera que mejor se espera rendimiento en el futuro.

Por otro lado, el Internet de las Cosas (IoT por sus siglas en inglés) y la industria

4.0 han requerido la introducción de dispositivos autónomos e inteligentes, además del uso de maquinaria en el sector industrial.

En el sector bancario, se tiene el caso de BBVA Bancomer, en el cual han integrado un asistente de chat virtual vía WhatsApp. Su objetivo es facilitar la interacción entre los usuarios y el banco para dar respuesta rápida acerca de la localización de sucursales, como abrir una cuenta, etc.

El Asistente Virtual de BBVA procesa texto y voz para mejorar la interacción, hace uso de datos y de ML para procesar la información que convierte. Este usa algoritmos de inteligencia artificial para entender y aprender los requerimientos y consultas de los clientes, lo que le permite ampliar sus capacidades para futuras consultas.

La principal contribución de este trabajo es presentar una propuesta para mejorar el aprendizaje de los temas de ML y Big Data con capacitación práctica enfocada en casos de uso de la vida real.

1.1. Justificación

En la actualidad existen diferentes métodos para el análisis de grandes volúmenes de datos, haciendo que las ciencias de la información tomen un papel relevante en nuestra sociedad. Debido a su importancia, dentro de la LTICs de la ENES Morelia existen materias orientadas a la ciencia de datos, en especial a los métodos del Machine Learning. Estas materias, al igual que las técnicas y herramientas que se utilizan en el ML, son de alta importancia para el estudiante ya que forman la base que se requiere para materias más complejas como redes neuronales.

Así también, actualmente en la LTICs, las asignaturas del área del ML se imparten de manera teórica y práctica, especialmente debido al cambio de paradigma motivado por la pandemia médica del SARS-CoV-2. No obstante lo anterior y debido a la complejidad de los temas abordados, el rendimiento estudiantil no es el ideal.

Aunado a lo antes mencionado, la aplicación de los métodos del aprendizaje automático sobre grandes volúmenes de datos no es considerado dentro de los temarios de las diferentes asignaturas en el área. Por todo lo anterior, una práctica complementaria del alumnado, aplicada a problemas reales sobre el Big Data, reforzará el estudio y la comprensión de estos difíciles temas.

1.2. Hipotesis

En la Licenciatura en Tecnologías para la Información en Ciencias (ENES Morelia, UNAM) existe una cantidad considerable de estudiantes cuyo desempeño en las asignaturas del área del ML ha sido bajo debido a la complejidad de sus temas. Con el uso del manual de prácticas se espera que su desempeño mejore después de la intervención de este proyecto.

1.3. Objetivo

1.3.1. Objetivo General

Desarrollar un manual de prácticas para la enseñanza del Machine Learning, aplicado a gran escala, dirigido a estudiantes a partir de sexto semestre de la Licenciatura en Tecnologías para la Información en Ciencias o de otras licenciaturas de la ENES Morelia que cuenten con los conocimientos básicos del Machine Learning.

1.3.2. Objetivos Particulares

1. Desarrollar una encuesta para diagnosticar los conocimientos previos e intereses del alumnado.
2. Establecer los temas que se van a cubrir en el manual de prácticas, relacionados al ML y con fundamento en la encuesta aplicada.
3. Elaborar el marco teórico del manual de prácticas con base en los temas seleccionados.

4. Determinar los ejemplos prácticos del ML que se abordarán en el manual de prácticas, conciliando con los docentes de la LTICs de las asignaturas del área del ML, la pertinencia de las prácticas propuestas enfocadas al Big Data.

5. Implementar los ejemplos prácticos usando el lenguaje Python.

6. Realizar una prueba piloto del manual de prácticas.

7. Realizar el diagnóstico de los resultados de la intervención a través de una encuesta de salida.

8. Publicar los resultados en la página web del proyecto.

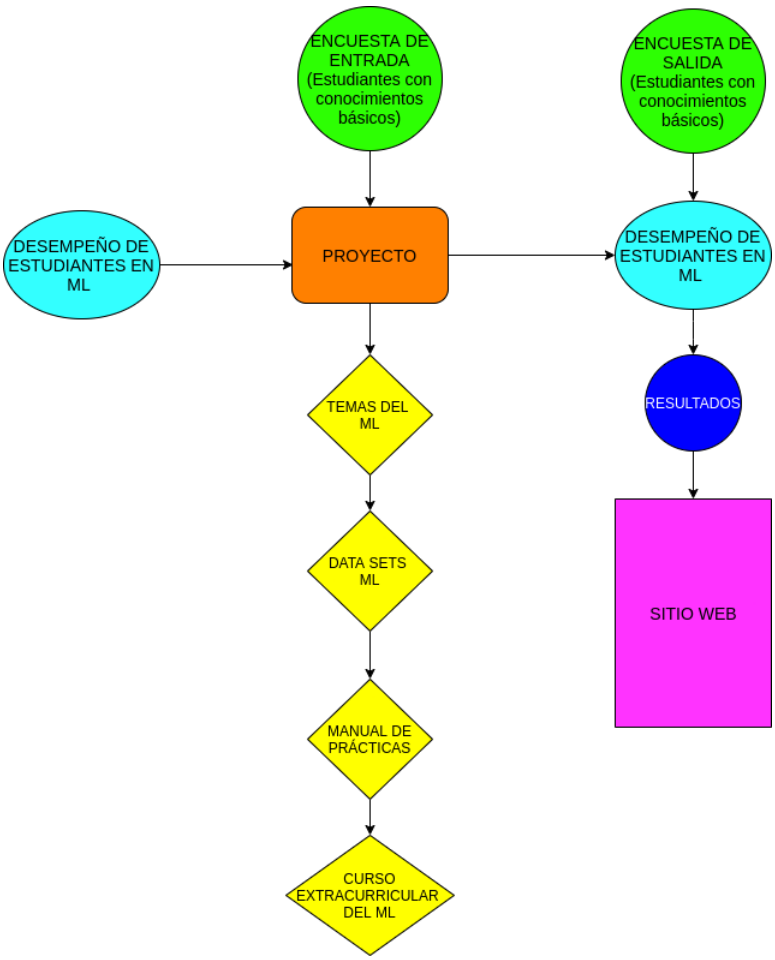

Figura 1.1: Mapa mental del desarrollo del proyecto.

1.4. Descripción general

Este documento está organizado de la siguiente manera:

- El capítulo 2 es una recopilación de los antecedentes más relevantes que existen sobre la inteligencia artificial y el Machine Learning. Dentro de este capítulo también se abarcan temas como los tipos más comunes de ML, una breve descripción de sus algoritmos más populares así como su uso en la actualidad en diferentes áreas del conocimiento.

- El capítulo 3, se habla de forma más técnica y teórica sobre el funcionamiento de los algoritmos que se usarán en las prácticas (capítulo 5). En este también se abarca la explicación sobre qué es y cómo funciona la librería Dask de Python, además de la aplicación y funcionamiento de los sistemas HDFS.

- El capítulo 4 se habla sobre las principales métricas de evaluación para modelos de ML. En este se explica la razón de cada métrica además de mostrar las formulas para calcularlas y dar la interpretación de las mismas dados los valores que pueden tomar.

- El capítulo 5 enlista y da un resumen de la metodología de las prácticas. En este se explica como se lleva a cabo la estructura de cada práctica, los datos que se van a usar, la métrica de evaluación a considerar además de explicar en un par de oraciones cuál es el objetivo de cada una de ellas.

- Finalmente, en el capítulo 6 se hace una recopilación de los datos obtenidos después de haber impartido el primer módulo del diplomado enfocado en el ML. Se muestran los resultados relacionados a la mejora del aprendizaje en ML en estudiantes de LTICs y académicos con formación afín de esta.

Capítulo 2

Antecedentes

Según Kaplan (2016) se le llama Inteligencia Artificial (IA) a la ciencia que se encarga de crear sistemas inteligentes que sean capaces de imitar el comportamiento inteligente de los humanos para la toma de decisiones y lograr metas.

Haenlein y Kaplan (2019) describen que en el año 1942 se creía que la inteligencia artificial era un fenómeno futurista gracias al escritor Isaac Asimov y su obra *Runaround*.

Unos años después, el británico Alan Turing (1950) publicó su artículo *"Computing Machinery and Intelligence"* donde describe por primera vez cómo crear inteligencia artificial mediante las máquinas de Turing, además de explicar cómo se realiza la prueba de Turing para diferenciar IA de la inteligencia humana.

Mitchell (1997) establece que la IA se utiliza para resolver problemas complejos que la programación convencional no puede.

En su artículo, Jordan y Mitchell (2015), explican que dentro de la IA existe una rama llamada Machine Learning (ML) cuyo fin es mejorar el rendimiento de los algoritmos a través de su experiencia. Está técnica hace uso de herramientas como estadística, informática, matemáticas y ciencias computacionales y se fundamenta en el análisis de los datos.

La definición de Mitchell y cols. (1997) es la siguiente: “Se dice que un programa de computadora aprende de una experiencia E, con respecto a una tarea T y una medida de desempeño P, si su desempeño con respecto a T, medido por P, mejora con la experiencia E”

Lee y cols. (2017) hacen mención a Arthur Samuel, pionero en el estudio del ML, quien lo definió de la siguiente manera

“El aprendizaje automático es el campo de estudio que le da a las computadoras la habilidad de aprender, sin que esté explícitamente programada” Samuel (1959)

El término Machine Learning se acuñó oficialmente alrededor del año 1960, según lo relatado por Liu (2020). Este nombre consiste en la palabra “Machine”, que hace alusión a cualquier dispositivo (robot, computadora) y la palabra “Learning” que hace referencia a la capacidad que se tiene de adquirir o descubrir patrones.

En la actualidad Mohri y cols. (2018) consideran al ML como la técnica de crear sistemas que sean capaces de aprender por sí mismos, utilizando grandes volúmenes de datos, haciendo que estos sean aptos para realizar análisis y, con ello, poder predecir futuros comportamientos.

2.1. Tipos de Machine Learning

El ML ha ganado importancia en las últimas décadas debido a su habilidad de realizar predicciones a partir de un conjunto de datos. Murdoch y cols. (2019) mencionan que los diferentes modelos de ML tienen la capacidad de adquirir conocimiento, relacionando características contenidas en los datos. A esto se le conoce comúnmente como “interpretaciones".

Géron (2019) explica que existen diferentes enfoques para el diseño de un sistema de ML. Estos se dividen en:

- Si están entrenados bajos supervisión humana o no. A estos se les denominan: **supervisado, no supervisado y de refuerzo.**
- Si pueden o no aprender sobre la marcha, denominados como **aprendizaje en línea.**
- Si detectan patrones de entrenamiento o si comparan nuevos datos con datos ya existentes. Este tipo de ML se cataloga como **aprendizaje basado en instancias o aprendizaje basado en modelos**

Aprendizaje Supervisado

El aprendizaje supervisado suele usarse cuando se cuentan con datos de los cuáles ya se sabe la respuesta que se desea predecir. Cunningham y cols. (2008) explican que este aprendizaje consiste en que el sistema pueda mapear entre los datos de entrada (input) y sus respectivas etiquetas (output) para después predecir las etiquetas dados nuevos datos no etiquetados, como se muestra en la Figura 2.1.

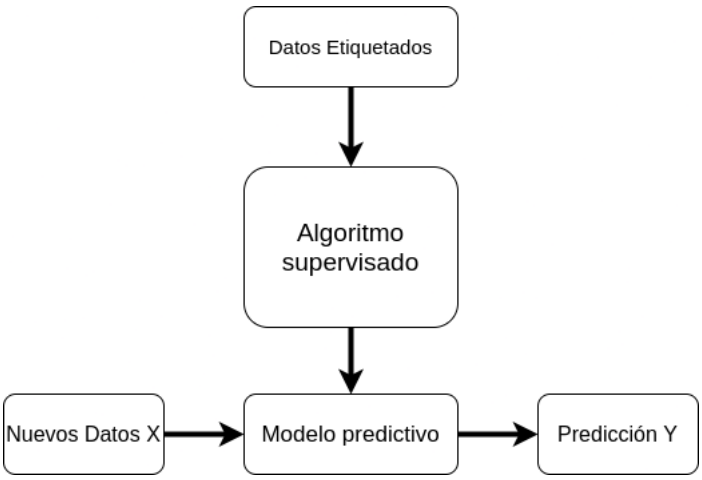

Figura 2.1: Diagrama del aprendizaje supervisado

Según El Naqa y Murphy (2015) el principal objetivo es que el sistema aprenda a distinguir las características de una etiqueta de otra. El aprendizaje supervisado

tiene la tarea de resolver los siguientes dos problemas Ayodele (2010), los cuales no pueden resolverse con programación simple:

- **Regresión.**

  Jagielski y cols. (2018) definen la regresión como un método en el cual se hacen uso de variables numéricas, para realizar predicciones, las cuales se espera que cada vez tengan menor margen de error. Estas variables se estudian para encontrar correlación entre ellas y sus respectivas etiquetas, para realizar predicciones de acuerdo a los patrones encontrados Montgomery y cols. (2021).
  Existen dos tipos principales de regresión: lineal y logística.

- **Clasificación.**

  La clasificación también se usa para hacer predicciones utilizando un conjunto de datos etiquetados, pero a diferencia de la regresión, la clasificación realiza predicciones discretas (llamadas clases) Li y cols. (2001).

  Kotsiantis y cols. (2007) mencionan los siguientes algoritmos que se usan para clasificación (entre otros):

  - Árboles de decisión.

    Este es un método muy utilizado debido a que es un algoritmo simple, fácil de comprender y porque no requiere de parámetros. Su y Zhang (2006) explican que el funcionamiento de los árboles de decisión consiste en un algoritmo recursivo en el que en cada iteración escoge el atributo cuyo valor es más adecuado para dividir el conjunto de datos, hasta que todos los datos sean clasificados.

  - k-vecinos más cercanos.

    Song y cols. (2007) explican que el algoritmo tiene la tarea de predecir la etiqueta de un dato (x_0) dados los k datos más cercanos, es decir, aquellos con menor distancia (euclidiana, distancia del coseno, etc.). Una vez que se tienen los k vecinos más cercanos, se revisan sus etiquetas y se le asigna a x_0 la etiqueta más repetida.

- Redes neuronales artificiales (RNA).

  Wang (2003) define a las RNA como un modelo que consiste en una capa de neuronas de entrada, algunas capas de neuronas ocultas y una capa de neuronas de salida. Cada conexión entre capas está asociada a un valor numérico (peso). También cuentan con funciones de activación, la más común es la función sigmoide.

Una de las aplicaciones de la clasificación, por ejemplo, es para detectar correo electrónico spam, como se ve en la Figura 2.2.

Figura 2.2: Un ejemplo de aprendizaje supervisado usado para clasificación de Spam

Aprendizaje No Supervisado

En el caso del aprendizaje no supervisado, los datos no están etiquetados, esto hace que el sistema tenga que aprender por sí mismo sin que se le indique si la clasificación es correcta o no Raschka y Mirjalili (2019)
El aprendizaje no supervisado, según Sathya y Abraham (2013), tiene la habilidad de aprender y organizar información, detectando patrones.

Para lograr clasificar los datos, se utiliza una técnica de agrupamiento o mejor conocida como clustering. Dayan y cols. (1999) proponen que el objetivo del clustering es agrupar datos cuyas características sean similares entre sí, como se ve en la Figura 2.3.

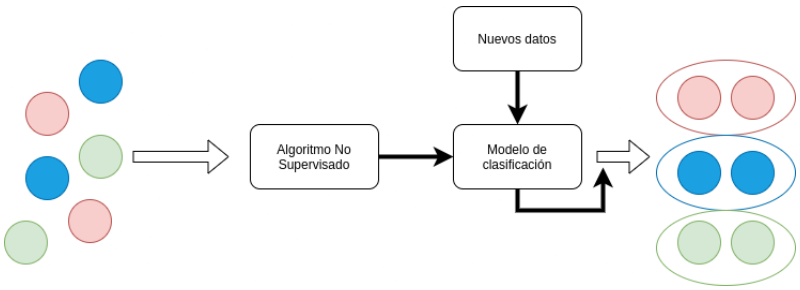

Figura 2.3: Un ejemplo de aprendizaje no supervisado usado para clustering, basado en los atributos de los datos.

Para implementar el clustering, Celebi y Aydin (2016) mencionan estas técnicas de aprendizaje no supervisado (entre otras):

- k-medias.

  Na y cols. (2010) explican que este algoritmo consiste en seleccionar aleatoriamente k centros, después calcular la distancia euclidiana (u otra métrica de distancia) de los demás datos para determinar cuál de los k centros es el más cercano y de esa forma clasificarlo en uno de los k grupos.

- Visualización y Reducción de Dimensiones.

  Vlachos y cols. (2002) mencionan que la reducción de dimensiones consiste en que un conjunto de datos reduzca su dimensionalidad sin la pérdida de información, para esto es común usar técnicas como Análisis de Componentes Principales (PCA en inglés) para que el conjunto de datos sea más fácil de procesar y de visualizar.

- Reglas de Asociación.

  De acuerdo con Aher y Lobo (2012), las reglas de asociación se usan comúnmente en minería de datos para encontrar de forma eficiente patrones o correlación en un gran conjunto de datos, para posteriormente obtener información de estos.

Aprendizaje Por Refuerzo

Wiering y Van Otterlo (2012) mencionan que el aprendizaje por refuerzo tiene como objetivo que el sistema aprenda en un entorno en el cual la única retroalimentación consiste en una recompensa escalar, la cual puede ser positiva o negativa (castigo).

La definición de Kaelbling y cols. (1996) es que el modelo recibe en cada iteración una recompensa r y el estado actual del entorno s, después el modelo toma una acción a de acuerdo con las entradas y eso es lo que se considera como la salida, la cual cambiará el estado s en la siguiente iteración.

En la Figura 2.4 se puede ver, de manera muy general, el comportamiento del aprendizaje por refuerzo.

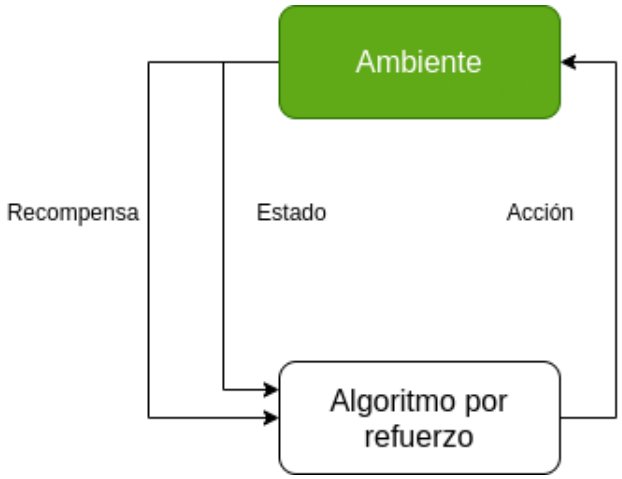

Figura 2.4: Diagrama de un algoritmo de aprendizaje por refuerzo.

En los últimos años, este algoritmo ha ganado terreno en el campo de investigación debido a sus aplicaciones mencionadas en Sutton y Barto (2018) tales como:

- Algoritmos que juegan ajedrez (Alpha Zero).
- Controlador adaptable de parámetros.
- Toma de decisiones.
- "Phil prepara su desayuno" el cual es un proceso de subtareas (como abrir el refrigerador, caminar a la estufa, romper un huevo, etc.) para lograr una tarea grande (preparar el desayuno).

2.2. Uso de ML en la actualidad

Una de las principales aplicaciones del ML en la actualidad es usar métodos supervisados para el análisis de grandes volúmenes de datos para obtener información sobre ellos. Por ejemplo en van Zoonen y Toni (2016) se muestra el caso de análisis de texto en redes sociales para entender la interacción entre usuarios.

Además de la utilidad del aprendizaje supervisado en el campo de la computación y matemáticas, como el desarrollo de Google DeepMinds y AlfaGo Siau y Wang (2018), existen otros usos en diferentes campos de la ciencia, por ejemplo:

- En Abbasi y Goldenholz (2019) se ve que debido a la gran utilidad de los algoritmos de ML para encontrar patrones, tienen un gran campo de aplicaciones tales como la detección de anomalías en electroencefalogramas.
- En biología, Libbrecht y Noble (2015) señalan que existen algoritmos encargados del análisis de grandes bases de datos de genomas, los cuales se entrenan para identificar potenciadores, nucleosomas, entre otras cosas.
- En medicina, Kourou y cols. (2015) indican que los algoritmos de ML se usan en la detección de tumores y su clasificación como benignos y malignos.
- Dentro de la psicología, Jiang y cols. (2020) los proponen como ayuda a la detección y predicción del riesgo de padecer algún trastorno mental.

En la imagen 2.5 se puede apreciar un diagrama a grandes rasgos de que el ML es un campo de estudio dentro de la Inteligencia Artificial

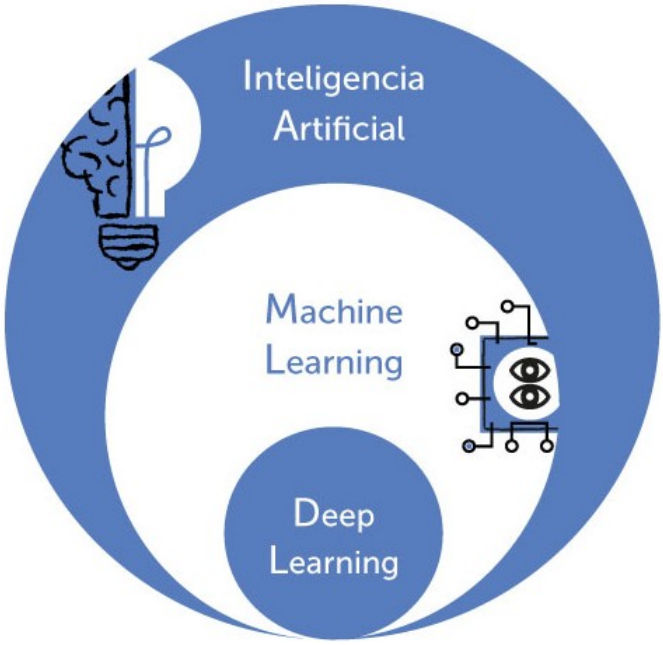

Figura 2.5: Diagrama de los campos de estudio dentro de la IA.

Capítulo 3

Algoritmos de Machine Learning

En la sección 5.1 se habló sobre los tipos de algoritmos del ML. En esta sección se profundizará en los algoritmos que se usarán en este manual de prácticas.

3.1. Árbol de decisión

Myles y cols. (2004) definen a un árbol de decisión como un algoritmo del tipo "divide y vencerás" usado comúnmente para hacer clasificación (aunque también puede usarse para regresión).
Su y Zhang (2006) proponen que el algoritmo inicia con un árbol vacío en el cual aún no hay nada de información acerca de los datos. Al ser un algoritmo avaricioso, este busca cual es el atributo que mejor divide el conjunto de datos y este se convierte en la raíz del árbol, después este proceso es recursivo, dividiéndose en subconjuntos que satisfacen la división de los datos.

A lo largo de los años, los investigadores han desarrollado diferentes algoritmos basados en árboles de decisión. Brijain y cols. (2014) explican que los modelos más importantes son los siguientes:

3.1.1. CHi-squared Automatic Interaction Detector (CHAID)

En su trabajo, Rodríguez y cols. (2016) mencionan que CHAID es un proceso que no hace suposiciones sobre los datos. El algoritmo determina cuál es la mejor forma de combinar las variables para predecir un resultado binario. Esto lo hace dividiendo cada variable en subconjuntos mutuamente excluyentes basados en la homogeneidad de los datos.

El criterio que se usa para determinar la división de los datos es *chi-squared test* (χ^2) Pandis (2016) explica que esta prueba solo muestra si existe una asociación entre variables, es decir, mide que tan dependiente es una variable de otra.
La forma de calcular χ^2, mostrada en McHugh (2013) es la siguiente:

$$\chi^2 = \sum_{i}^{j} \frac{(O_i - E_i)^2}{E_i} \tag{3.1}$$

Donde:
O = Los valores observados.
E = Frecuencia esperada.

Para calcular la frecuencia esperada (E) se usa la siguiente formula:

$$E = \frac{M_R * M_C}{n} \tag{3.2}$$

Donde:
M_R = Suma de la fila.
M_C = Suma de la columna.
n = Número total de datos.
La ecuación 2 se aplica para cada uno de los datos y con el resultado de cada uno de ellos, se calcula χ^2 también para cada dato. Este método ayuda mucho cuando se trata de análisis estadísticos.

3.1.2. Classification and regression tree (CART)

Según lo descrito por Loh (2011) estos modelos se obtienen mediante la división recursiva de los datos y ajustado un modelo simple de predicción en cada una de esas divisiones.

Este algoritmo usa una herramienta extra de aprendizaje, llamada "Poda". Loh (2014) explica que el método de poda es una herramienta bastante útil, ya que esta se basa en el concepto de eliminar al “eslabón más débil".
Estos valores de costo-complejidad se pueden medir usando los coeficientes de *Gini* y *Entropía*.

Gini: Este coeficiente es el más usado en árboles de clasificación, Alvaredo (2011) explica que este es más sensible a las transferencias en el centro de las distribuciones de datos.
Timofeev (2004) menciona que Gini utiliza la siguiente formula de impureza:

$$i(t) = \sum_{kl} p(k|t)p(k|l) \tag{3.3}$$

3.1.3. C4.5

El trabajo de Singh y Gupta (2014) menciona que el algoritmo C4.5 genera un árbol que divide recursivamente el conjunto de datos.
Este árbol de decisión considera todas las posibles divisiones de los datos para seleccionar el división que genere la mayor ganancia de información.

Cintra y cols. (2013) dicen que los árboles de decisión C4.5 usan la entropía y la ganancia de información como métricas para decidir cuales son los datos que mejor dividen el conjunto de datos.

Las definiciones matemáticas de la entropía y la ganancia de información, según

Ahmad y cols. (2020) es la siguiente:

$$Entropia(S) = \sum_{i=1}^{n} -p_i \times log_2 p_i \tag{3.4}$$

Donde:

S es la entropía

p es la proporción de la clase (output)

$$Ganancia(S, A) = Entropia(S) \sum_{i=1}^{n} \frac{|S_i|}{|S|} \times Entropia(S) \tag{3.5}$$

Donde:

S es el caso a evaluar

A es un atributo a evaluar

$|S_i|$ es el caso actual

$|S|$ es el número total de casos

3.2. Modelos de Regresión

Los modelos de regresión, son definidos por Maulud y Abdulazeez (2020) como métodos matemáticos utilizados estimar la relación entre variables. Es uno de los métodos más comunes en ML para la predicción de datos.

Actualmente, la regresión es una herramienta importante que ya ayuda a los analistas y estadistas a entender las relaciones que existen entre los datos, Kumari y cols. (2018) enlista las siguientes razones por las cuales este método es importante:

- Descriptivo: Este método ayuda a analizar la fuerza que han entre las variables dependientes X y la variable independiente y.
- Ajuste: El método es capaz de ajustarse para minimizar el error.

Existen dos principales modelos de regresión, la lineal y logística.

3.2.1. Regresión lineal

La regresión lineal, a pesar de ser uno de los métodos más utilizados en ML, no es aplicable para cualquier problema, por ejemplo, no se puede usar la regresión lineal a los problemas donde las variables son categóricas.
Por lo general, las variables a predecir tienen que ser numéricas, para que la regresión lineal pueda ser aplicable exitosamente.

En Montgomery y cols. (2021) se define la formula de regresión lineal en $\mathbb{R}^2$ de la siguiente forma:

$$y = \beta_0 + \beta_1 x \tag{3.6}$$

Donde
β_0 es la intercepción de la recta
β_1 es la pendiente
x son las variables dependientes

Para $\mathbb{R}^n$ la formula queda de la siguiente forma:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon_i \tag{3.7}$$

Donde
ϵ es el error (distancia) entre la recta calculada y x_i

Un ejemplo simple de regresión lineal en $\mathbb{R}^2$ es el que se muestra en la 3.1:

3.2.2. Regresión logística

La regresión logística, a diferencia de la lineal, es usada comúnmente para realizar predicciones binarias.

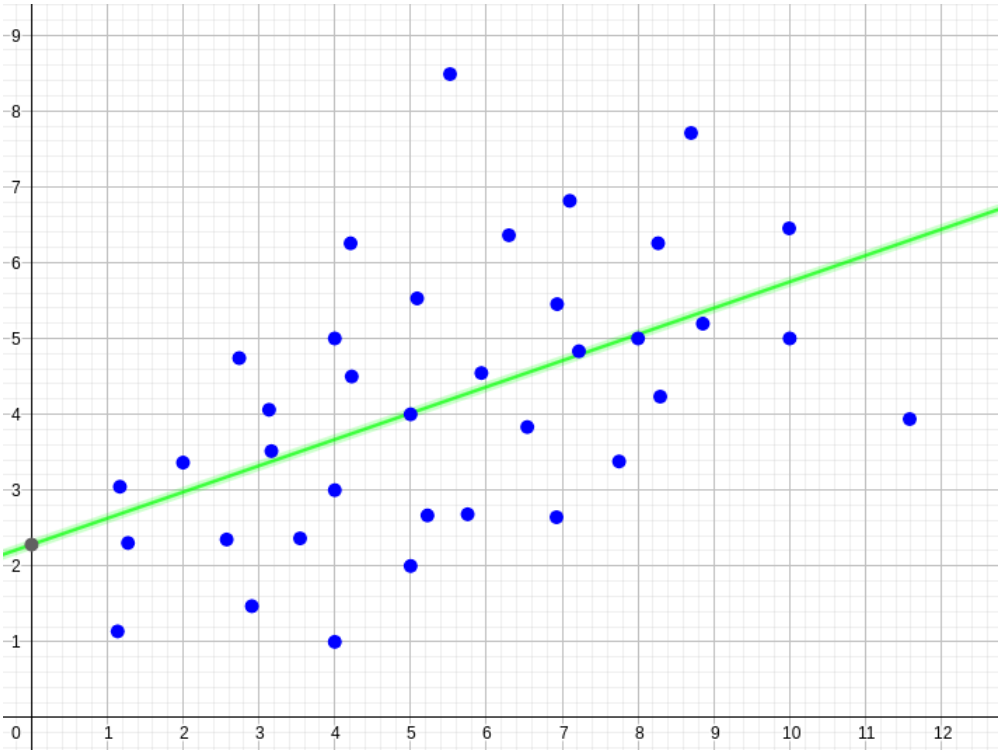

Figura 3.1: Ejemplo de una recta calculada con regresión lineal donde y = 2.2816+0.3464x

Harrell (2015) explica que el algoritmo consiste en generar la ecuación de la función sigmoide 3.8 que permita explicar la relación que existe entre las variables independientes y la variable dependiente (X y y).

$$y = \frac{1}{(1 + e^{-x})} \tag{3.8}$$

Donde:

x son todas las variables dependientes representadas como ecuaciones de una recta.

Para determinar la clasificación, el modelo de regresión logística se toma en cuenta la salida de la función de la recta (x) aplicada en la función sigmoide 3.8.

- Si la salida es <= 0.5, entonces el algoritmo lo toma como 0
- Si la salida es >0.5, entonces el algoritmo lo toma como 1

Lo anterior se puede apreciar en la imagen 3.2

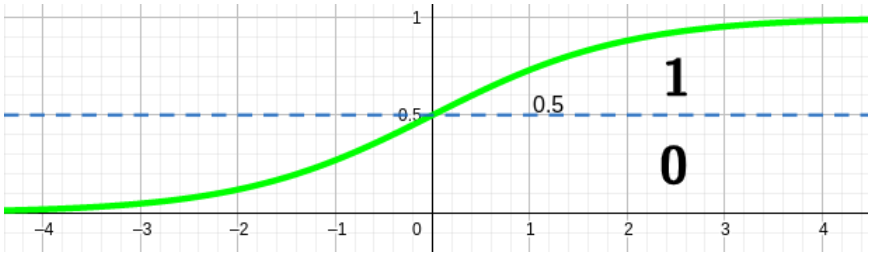

Figura 3.2: Función Sigmoide aplicada a la regresión lineal

3.3. *k*-vecinos más cercanos (KNN)

Como se mencionó en el capitulo 2, existen algoritmos de regresión y clasificación y uno de los algoritmos más importantes de clasificación es K-vecinos más cercanos o KNN por sus siglas en inglés (K-Nearest Neighbor).

En su trabajo, Zhang y cols. (2017) mencionan que KNN es un algoritmo supervisado que sirve para clasificar valores (y) buscando los puntos de datos "más similares" aprendidos en la etapa de entrenamiento.
El procedimiento consiste en calcular la distancia entre el "vecino" a clasificar y el resto de "vecinos" del dataset de entrenamiento. 3.3

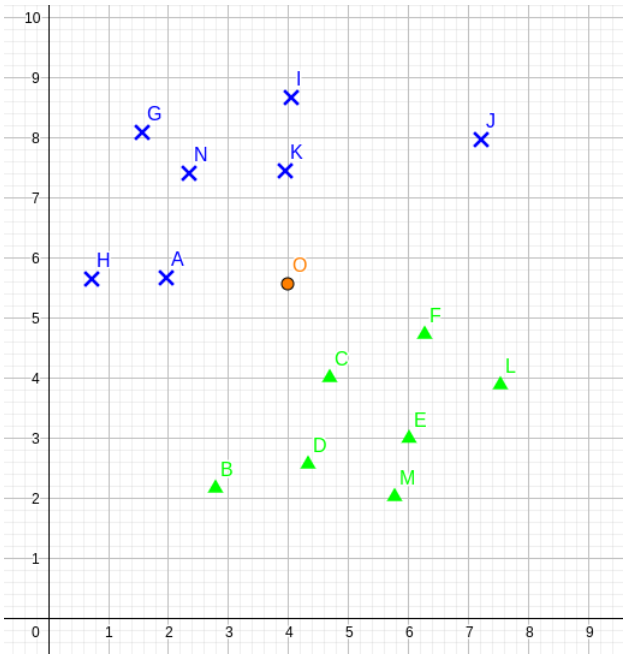

Figura 3.3: Nuevo punto a clasificar

Se seleccionan los "k vecinos" más cercanos, es decir, con menor distancia, según la función de distancia que se emplee (euclidiana, coseno, etc) 3.4.

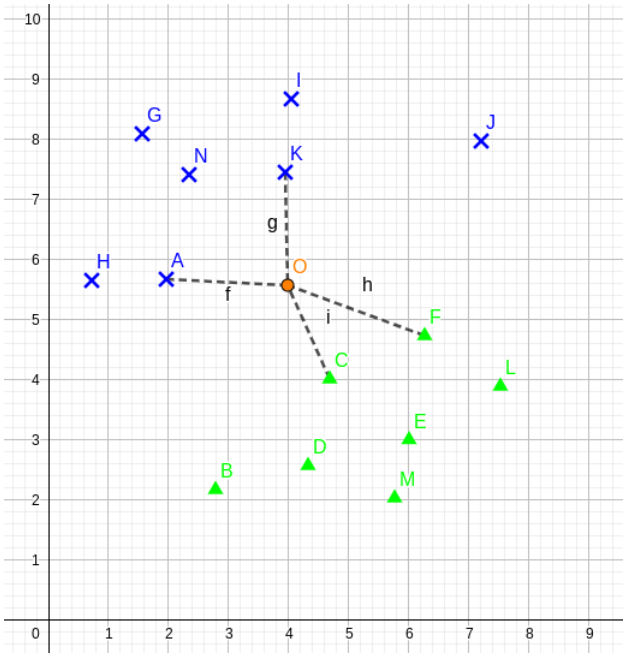

Figura 3.4: Vecinos más cercanos (con menor distancia)

En caso de usar la distancia euclidiana como métrica de distancia, Adithiyaa y cols. (2020) la definen de la siguiente manera 3.9:

$$D(x, y) = \sqrt{\sum_{i=i}^{n} (x_i - y_i)^2} \tag{3.9}$$

Una vez que se obtienen los puntos con menor distancia, se seleccionan los k vecinos más cercanos y la etiqueta que domine el conjunto (la etiqueta más frecuente) es la que decidirá la clasificación del punto actual.

Como se puede ver en la figura 3.5 hay 2 puntos etiquetados como cruz y solo uno etiquetado como triángulo, por lo tanto, el punto en cuestión se etiquetaría como una cruz azul.

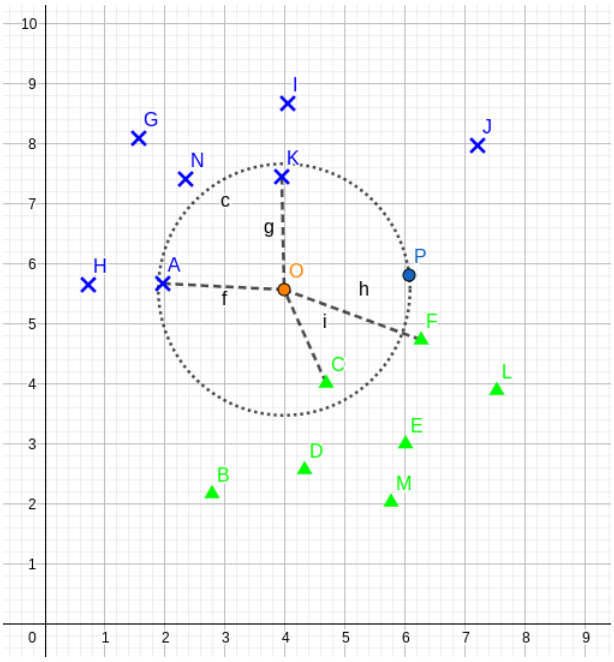

Figura 3.5: Vecinos más cercanos (con k = 3)

3.4. Clustering (K-means)

K-means es un algoritmo de clasificación no supervisada (clustering) que agrupa objetos en k grupos basándose en sus características.

K-means es definido por Kanungo y cols. (2002) como un algoritmo iterativo que se encarga de buscar la solución mínima local.

Para encontrar el mínimo local, del conjunto de datos se seleccionan aleatoriamente k puntos 3.6, llamados centroides, los cuales tendrán su respectivo vecindario de n puntos.

Después, para cada punto en cada vecindario, se toma la distancia de este a su respectivo centroide (o al más cercano) 3.7. Por lo general se usa la distancia euclidiana 3.10 para obtener las distancias de cada punto al centro de su subconjunto:

$$min_s E(\mu_i) = min_s \sum_{i=1}^{k} \sum_{x_j \in s_i} ||x_j - \mu|| \tag{3.10}$$

Se consideran la menor distancia obtenida, para que el punto actual se etiquete de

acuerdo a su centroide más cercano 3.8.

Pérez y cols. (2007) mencionan los casos en que algoritmo converge, por ejemplo:

1. Cuando el algoritmo ha llegado al número de iteraciones especificado al inicio del algoritmo.

2. Cuando la diferencia entre los centroides de dos iteraciones consecutivas es más pequeño que un umbral dado.

3. Cuando no hay intercambio de elementos entre los k grupos.

Después de Que el algoritmo termina su entrenamiento, se tienen k grupos cuyos puntos comparten características 3.9

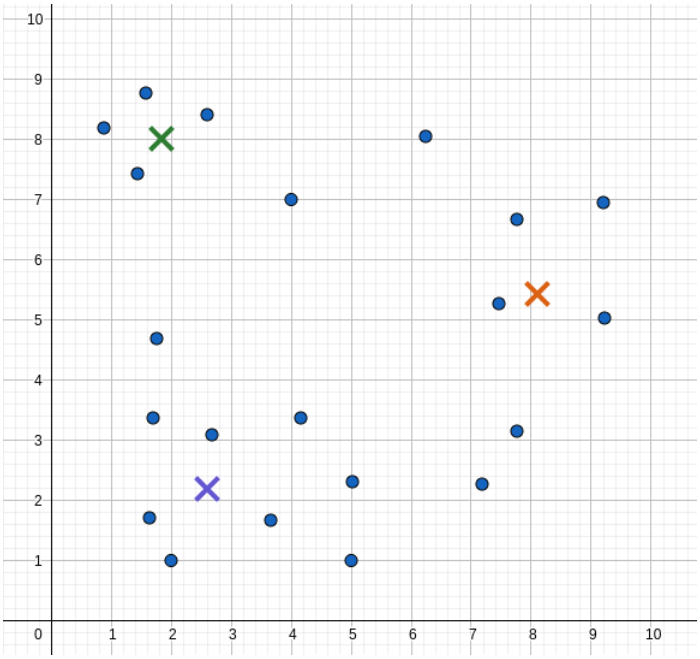

Figura 3.6: k = 3 centroides en un conjunto de datos

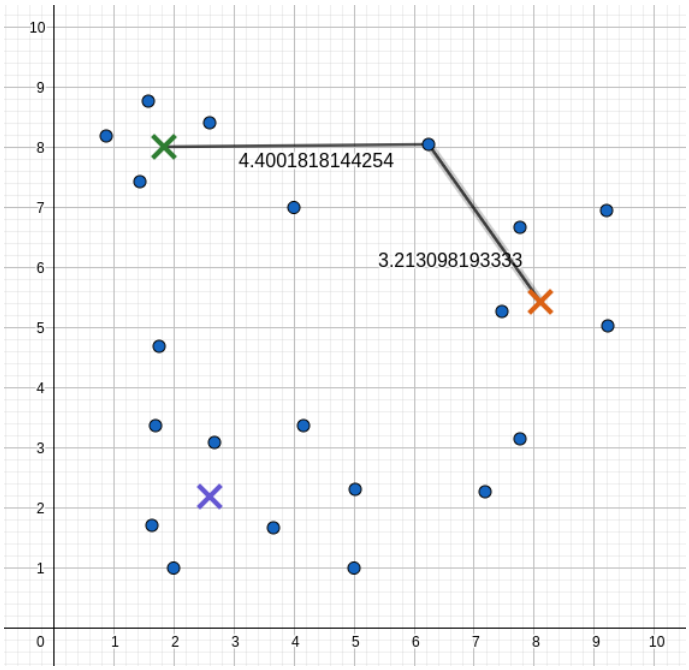

Figura 3.7: Buscando el centroide más cercano

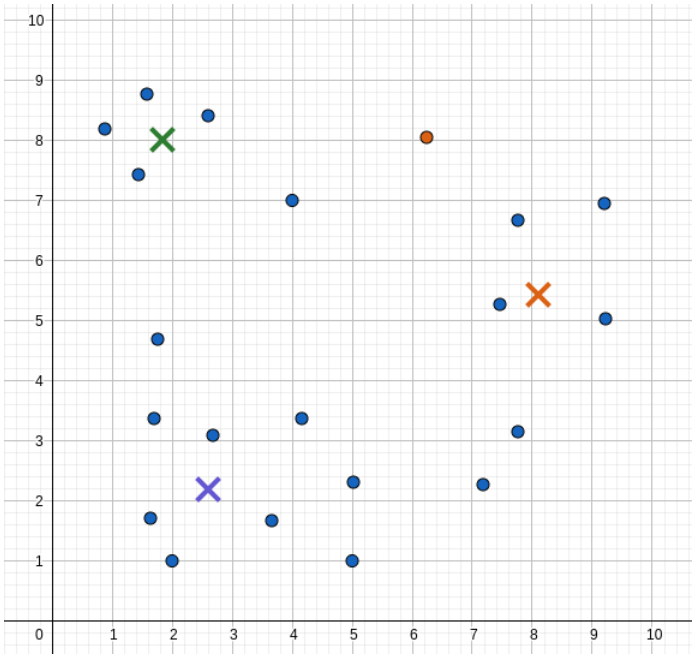

Figura 3.8: etiquetando el punto actual de acuerdo a su grupo

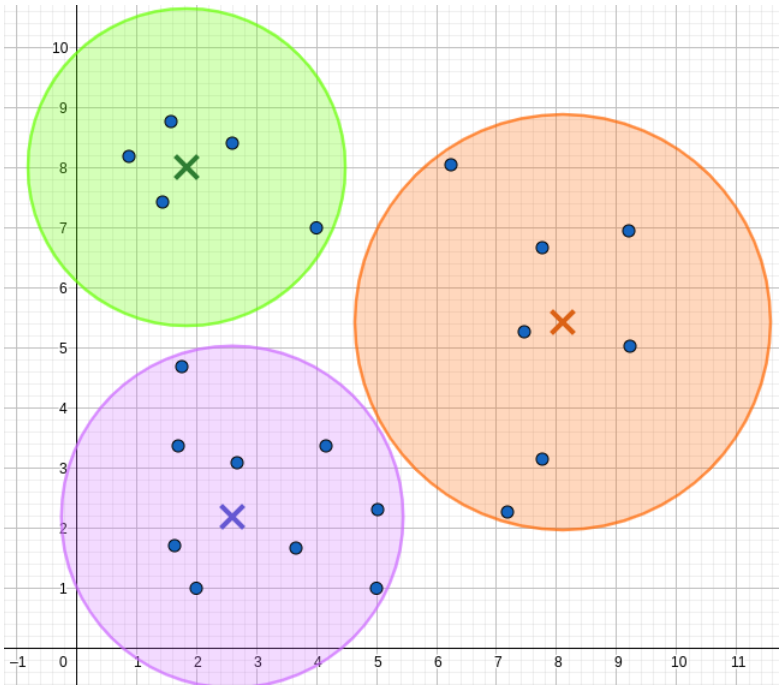

Figura 3.9: k grupos formados

3.5. Dask

En muchos de los casos, se usan herramientas de python como pandas, numpy, scikit-learn, etc para crear modelos de ML. Pero ¿qué se hace cuando el volumen de datos es más grande y pesado de lo que las librerías convencionales de python pueden procesar?

Para ese tipo de casos, python tiene una librería especial dedicada al procesamiento de grandes volúmenes de datos.

Como se muestra en su página https://www.dask.org/, dask simplifica las tareas de paralelización a la hora de hacer procedimientos de ML y DL.

Su característica principal es que dask está escrito sobre numpy y pandas, mientras que *dask-ml* está escrito sobre scikit-learn haciendo que la sintaxis de dask y pandas sea muy similar. Esto se puede ver en la figura 3.10 Ambos comandos dan el mismo resultado, es decir, regresan un dataframe con los datos cargados en un archivo csv.

Una propiedad especial que tiene dask, es que se pueden cargar varios archivos csv en una sola sentencia, siempre y cuando estos compartan esquema (mismas columnas).

```
import pandas as pd
df = pd.read_csv('2015-01-01.csv')
df.groupby(df.user_id).value.mean()
```

```
import dask.dataframe as dd
df = dd.read_csv('2015-*-*.csv')
df.groupby(df.user_id).value.mean().compute()
```

Figura 3.10: Sintaxis de Pandas (arriba) Sintaxis de Dask (abajo)

De igual forma, tiene una sintaxis muy similar a la de numpy, justo como se ve en la imagen 3.11. De esta forma, dask puede realizar los mismos cálculos que puede hacer numpy.

```
import numpy as np
f = h5py.File('myfile.hdf5')
x = np.array(f['/small-data'])

x - x.mean(axis=1)
```

```
import dask.array as da
f = h5py.File('myfile.hdf5')
x = da.from_array(f['/big-data'],
                  chunks=(1000, 1000))
x - x.mean(axis=1).compute()
```

Figura 3.11: Sintaxis de Numpy (izq) Sintaxis de Dask (der)

3.6. Sistema de archivos HDFS

Un Sistema de Archivos Distribuidos de Hadoop o HDFS por su nombre en inglés Hadoop Distributed File System tiene como función principal almacenar grandes volúmenes de datos de manera distribuida.

De acuerdo a Karun y Chitharanjan (2013) un HDFS tiene una gran tolerancia a los fallos ya que este ha sido diseñado para ser implementado en sistemas cuyo hardware no requiera un gran costo de procesamiento.

El manual de HDFS escrito por Borthakur y cols. (2008) muestra que la arquitectura de este sistema consiste en el uso de clúster en los cuales se crean subconjuntos de datos, dando una arquitectura de Maestro y trabajadores.

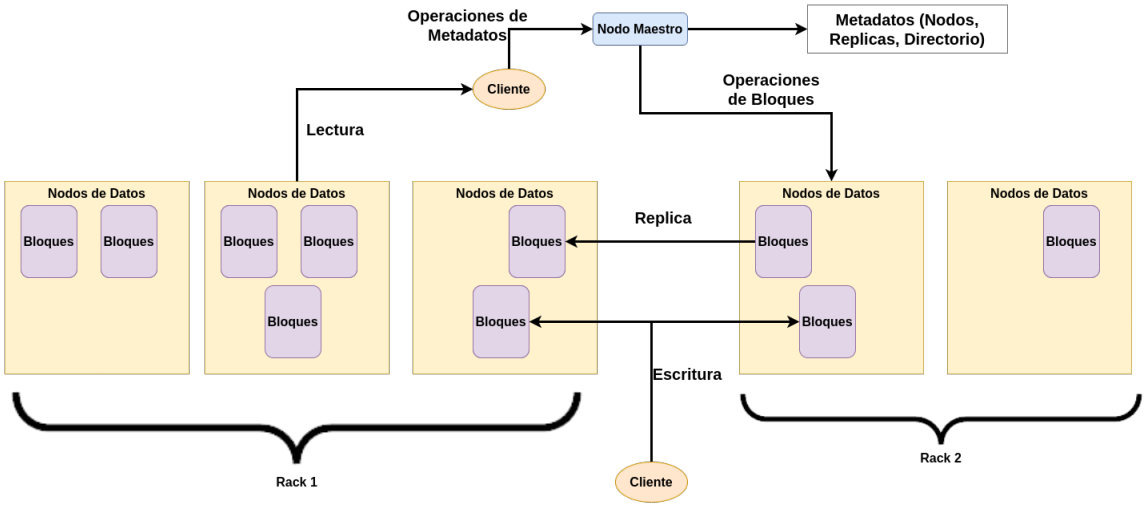

Figura 3.12: Diagrama de un HDFS

Como se puede ver en la figura 3.12 la arquitectura consiste en los siguientes elementos:

- Nodo Maestro, este almacena todos los datos del sistema de archivos en un clúster. Este también se encarga de almacenar todos los metadatos del clúster.
- Nodos de Datos son servidores de un solo archivo, lo que significa que si uno de estos nodos falla, el archivo aún se encuentra disponible en cualquier momento.
- Bloques de datos que representan un archivo. Cada bloque es replicado y añadido a un Nodo de Datos. Este proceso de replicación es rápido debido a que los bloques solo almacenan el nombre, ruta y permisos del archivo.

Capítulo 4

Métricas para Evaluar Modelos

Las métricas de evaluación ayudan a mejorar el rendimiento de modelos de ML. Esto se logra calculando la diferencia entre las variables predichas por el modelo y el valor real de estas.
El evaluar un modelo con más de una métrica puede dar mejores resultados en un modelo de ML.

4.1. RMSE

Error de Raíz Cuadrada Media o RMSE por sus siglas en inglés (Root Mean Square Error) es definida por Barnston (1992) de la siguiente manera:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{4.1}$$

Donde:
$\hat{y}_i$ es el valor predicho por el modelo.
y_i es el valor real.
n el número de elementos en el conjunto de prueba.

El RMSE se puede interpretar como la desviación estándar de la varianza. Además de que esta nos da valores dentro de la misma escala de los datos.

Un RMSE bajo indica un mejor ajuste del modelo y un valor alto indican que el modelo requiere modificaciones.

4.2. MAE

Error absoluto medio (MAE por sus siglas en inglés) es una de las métricas más usadas para evaluar el desempeño de varios modelos de ML. Chai y Draxler (2014) menciona que MAE le da el mismo peso a todos los errores

$$\frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{4.2}$$

Donde:

$\hat{y}_i$ es el valor predicho por el modelo.

y_i es el valor real.

n el número de elementos en el conjunto de prueba.

Como se puede ver en la fórmula (**cambiar**) MAE mide qué tan cerca está la predicción en relación al valor real del conjunto de datos. Como mide el promedio de las distancias entre los valores reales y las predicciones, un MAE "perfecto" es cuando el promedio de las distancias es 0. Es decir, las predicciones fueron iguales a los valores reales.

Una modificación que se tiene de MAE es NMAE, Jannach y cols. (2010) muestran que consiste en normalizar los valores del MAE con respecto a los valores con los que se trabaja.

$$\frac{MAE}{r_{max} - r_{min}} \tag{4.3}$$

Donde:

r_{max} es el valor máximo del conjunto de datos.

r_{min} es el valor mínimo del conjunto de datos.

4.3. Matriz de confusión

Una Matriz de confusión, según López y cols. (2018) es una representación gráfica de los resultados de un modelo. Esta representación es algo similar a lo mostrado en la figura 4.1

Figura 4.1: Matriz de confusión simple de 2 valores (positivo/negativo)

Como se puede ver en la figura 4.1 existen 2 ejes, los valores de predicción y los valores reales. Dados estos ejes, la matriz de confusión se compone de:

- Verdaderos positivos: Son los valores clasificados como positivo y el valor real también es positivo.
- Falsos positivos: Son los valores clasificados como positivo y el valor real es negativo.
- Falsos negativos: Son los valores clasificados como negativo y el valor real es positivo.
- Verdaderos negativos: Son los valores clasificados como negativo y el valor real también es negativo.

Estos valores son útiles para calcular otras métricas de clasificación como la exactitud y la precisión.

4.4. Exactitud

Basados en la figura 4.1 existen verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN) y estos se usan para calcular la exactitud y la precisión.

La exactitud es definida por Borja-Robalino y cols. (2020) como:

$$exactitud = \frac{TP + TN}{N} \tag{4.4}$$

Donde:

TP Son las predicciones positivas predichas correctamente.

TN Son las predicciones negativas predichas correctamente.

N Es el total de predicciones tanto correctas como incorrectas.

La exactitud se interpreta como la proporción de predicciones acertadas con respecto al total de datos.

Debido a esto, la exactitud representa qué tan cercano están los valores predichos con el valor real de los datos.

4.5. Precisión

La precisión mide el grado de proximidad o cercanía de los resultados entre sí y esta es definida también por Borja-Robalino y cols. (2020) como:

$$precision = \frac{TP}{TP + FP} \tag{4.5}$$

Donde:

TP Son las predicciones positivas predichas correctamente.

FP Son las predicciones negativas predichas como positivas.

La precisión se interpreta como la proporción de verdaderos positivos reales con respecto a los valores positivos predichos.

Kellman y Hansen (2014) mencionan que la exactitud se refiere a los errores sistemáticos del modelo, generando sesgo en los datos, mientras que la precisión se relaciona con algún componente aleatorio el cual genera ruido. En la figura 4.2 se da un ejemplo sobre predicciones/clasificaciones que carecen de exactitud y/o precisión.

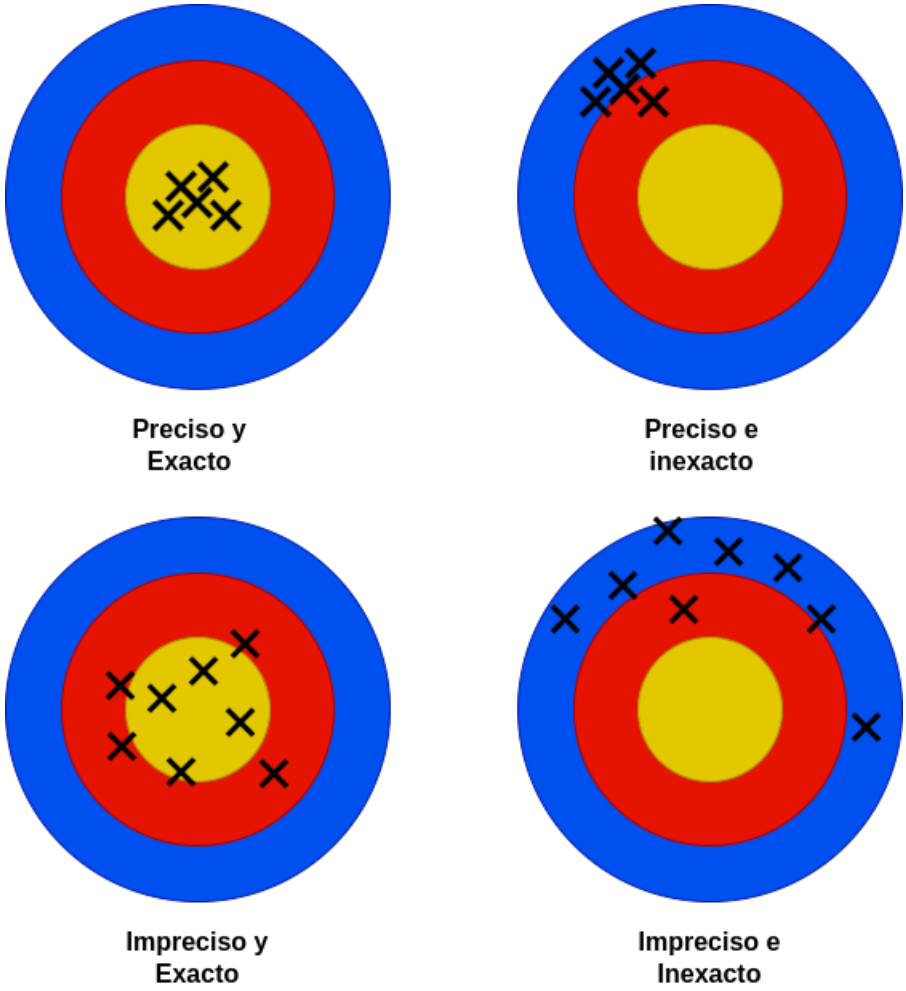

Figura 4.2: Representación gráfica de la exactitud vs precisión

4.6. Recall

El recall o sensibilidad muestra la proporción de verdaderos positivos predichos con respecto al número total de valores positivos.
En su trabajo, Davis y Goadrich (2006) definen el recall como

$$recall = \frac{TP}{TP + FN} \tag{4.6}$$

Donde:

TP Son las predicciones positivas predichas correctamente.

FN Son las predicciones positivas predichas como negativas.

De acuerdo a la ecuación 4.6 sabemos que si el recall da un valor cercano a 1, indica que el modelo tiene un buen rendimiento, en caso de dar un valor cercano a 0, indica que el modelo necesita ajustes

4.7. Fβ

Como se mencionó anteriormente, la precisión y el recall son métricas para cuantificar la calidad de clasificación de un modelo. De acuerdo con Derczynski (2016), estás métricas se pueden equilibrar de una forma proporcional, de acuerdo al objetivo deseado.

La Fβ es definida por Goutte y Gaussier (2005) de la siguiente manera:

$$F\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{4.7}$$

Donde:

P Es el valor de la Precisión.

R Es el valor del Recall.

β Determina el balance entre la precisión y el recall.

4.8. F1 score

De acuerdo a Chicco y Jurman (2020) el F1 es la métrica más común dentro del Fβ score.
F1 es una media armónica entre la precisión y recall, es decir, ambas métricas tienen el mismo porcentaje de importancia. Teniendo esto en consideración Huang y cols. (2015) define F1 de la siguiente manera:

$$F1 = 2\frac{PR}{P+R} \tag{4.8}$$

Como ambas métricas tienen la misma importancia, la única forma de tener un F1 alto es que P y R tengan un valor alto.

El problema de esto yace en que no siempre se pueden presentar casos en que la precisión y el recall sean altos debido a que si el valor de uno de ellos aumenta, el otro tiende a disminuir.

Para tratar este tipo de casos el valor de β puede variar de acuerdo al caso.
Los valores más utilizados en la práctica son:

- F1 con $\beta = 1$ representa el equilibrio entre precisión y recall.
- F0.5 con $\beta = 0{,}5$ le da más importancia a la precisión.
- F2 con $\beta = 2$ le da más importancia al recall.

Capítulo 5

Metodología de las Prácticas

Cada una de las prácticas presentadas en el anexo contienen las siguientes fases, en el orden en que se muestran:

1. Objetivo de la práctica.
2. Conceptos.
3. Herramientas a usar.
4. Desarrollo.
    a) Entender el Problema.
    b) Definir un criterio de evaluación.
    c) Preparar los datos.
    d) Construir el modelo.
    e) Análisis de errores.
    f) Implementación.

N° Práctica	**NOMBRE**	**DATASET**	**CRITERIO DE EVALUACIÓN**	**DESCRIPCIÓN**
1	Clasificación usando árboles de decisión	Pasajeros del Titanic	Exactitud y/o Métrica Fbeta	Construir un árbol de decisión para clasificar si los pasajeros del Titanic sobreviven o no dadas características como el sexo, edad, status, etc.
2	Predicción del costo de casa-habitación (Regresión Lineal)	California Housing	RMSE y/o MAE	Construir un modelo de predicción para el costo de las casa habitación en el este de California (década de 1990).
3	*k-vecinos más cercanos*	Pozos profundos del lago de Cuitzeo	Precisión	Usar un dataset de pozos para hacer un modelo de KNN que agrupe elementos conforme al volumen de extracción de pozos en Michoacán (supervisado).
4	Aprendizaje no supervisado (k-means)	Online Retail K-means & Hierarchical Clustering	No aplica	Diseñar un modelo de K-means para hacer clasificación las transacciones de los clientes de un banco y así poder identificar a los diferentes clientes que hay (no supervisado).
5	Instalación y uso de Dask	3 nodos virtuales con CPU y GPU c/u	No aplica	Mostrar cómo se realiza la instalación de Spark y cómo se usa para la manipulación de grandes volúmenes de datos.
6	Instalación y uso de HDFS	3 nodos virtuales con CPU y GPU c/u	No aplica	Enseñar paso por paso como se realiza la instalación de Hadoop y cuales la utilidad de los comandos de este mismo.
7	Predicción del clima (Regresión Lineal)	RUOA de 2015 a la actualidad	RMSE y/o MAE	Usar los datos climáticos obtenidos por la RUOA para hacer un modelo de regresión lineal capaz de predecir el clima de los siguientes días.

Capítulo 6

Resultados y Discusión

Para conocer los temas más relevantes sobre ML para los alumnos, se elaboró una encuesta la cual cuantificaba diferentes temas y herramientas de interés para los estudiantes.

Usando la información obtenida de la encuesta aplicada con anterioridad, se generaron análisis descriptivos donde se realizó el estudio de confiabilidad aplicando el Alfa de Cronbach obteniendo como resultado 0.956 y demostrando que la información obtenida es consistente.

También se aplicó un estudio de correlaciones utilizando la bivariada de Pearson y seleccionando únicamente las correlaciones obtenidas en los niveles alto y muy alto [0,7-0,93], que se muestran en la figura 6.1.

De acuerdo al análisis de correlaciones, se identificaron áreas de oportunidad según porcentaje de importancia que respondieron los encuestados. En la figura 6.2 se muestra esta importancia.

De acuerdo con la encuesta desarrollada, se observó que los encuestados tenían cierto desconocimiento en algunos temas. En la figura 6.3 los temas se muestran por eje, ordenados por nivel de desconocimiento: No sé (dnK), Nada importante (NImp), Menos importante (LImp), Neutro Importante (Imp), Muy importante (VImp) y para las herramientas, la escala es: No sé (dnK), Corta, Media y Alta.

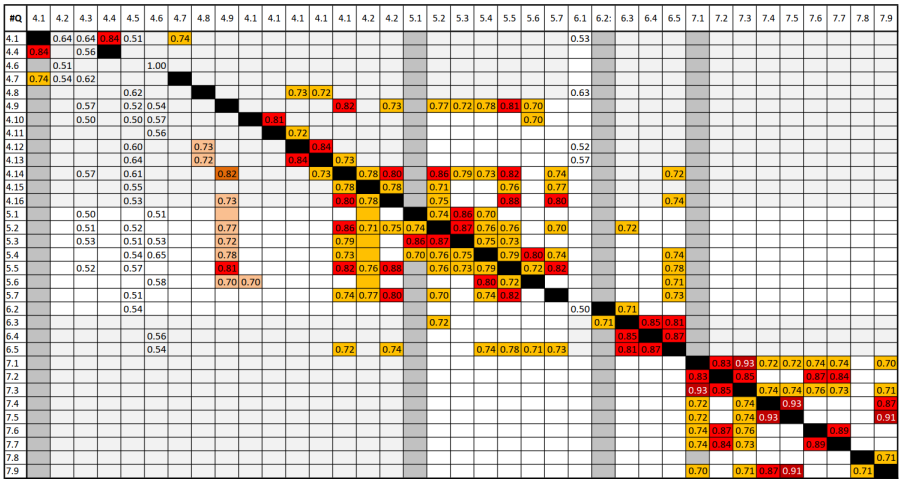

Figura 6.1: Matriz de Correlación de los resultados obtenidos en la encuesta

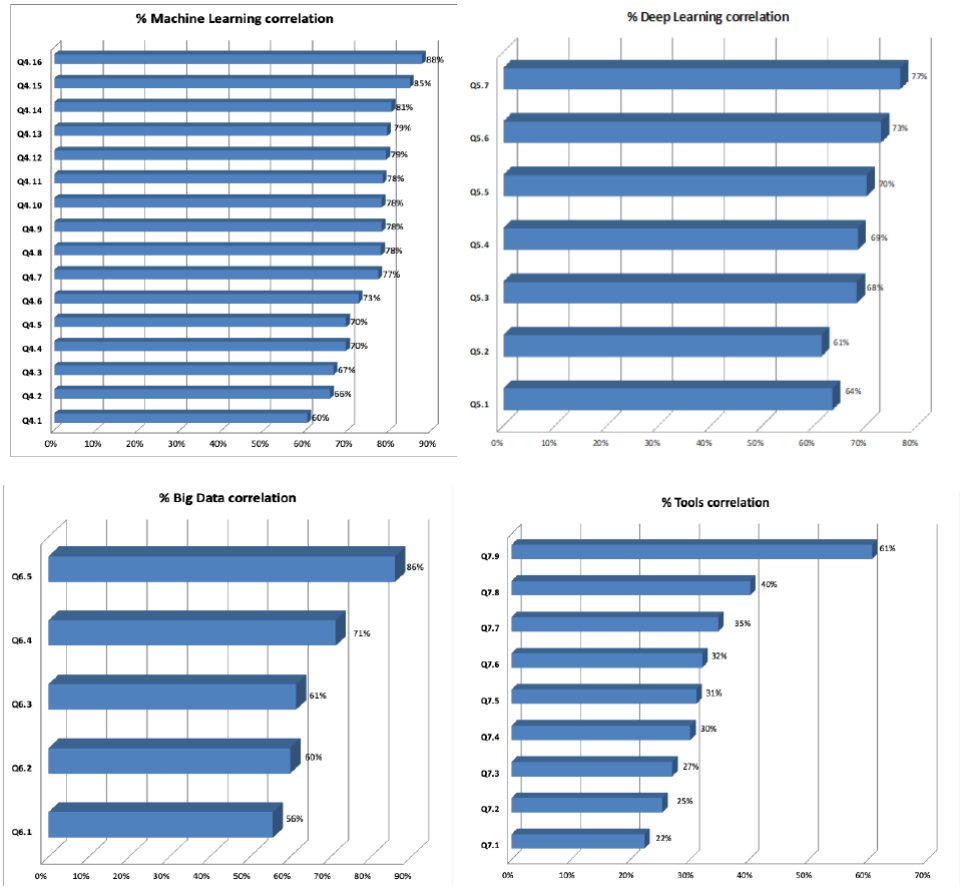

Figura 6.2: Nivel de importancia de acuerdo a los encuestados

Por la experiencia presentada durante el diplomado práctico dirigido a una mezcla de estudiantes y docentes de la Morelia campus de la universidad UNAM, divididos

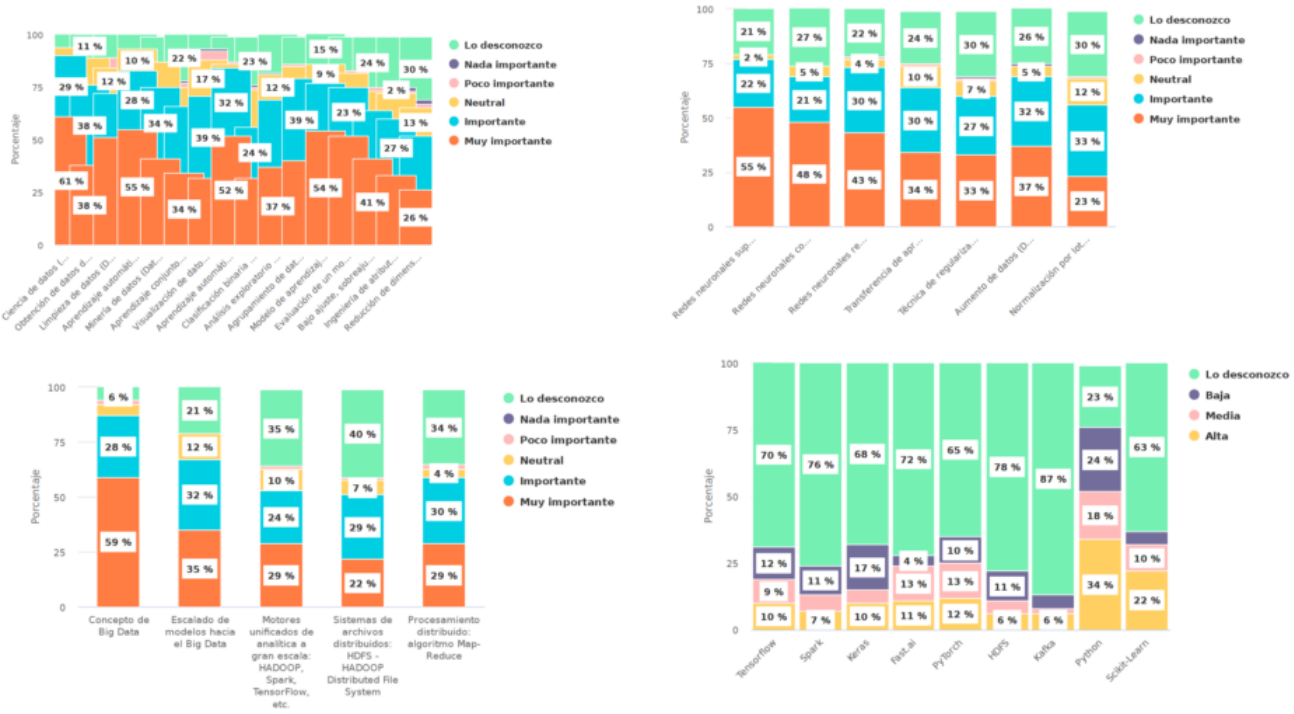

Figura 6.3: Niveles de desconocimiento

en dos grupos heterogéneos, en cuanto a la aplicación de las prácticas propuestas, se ofrecieron dos cursos de acuerdo al diplomado descrito a continuación:
MÓDULO I. Aprendizaje automático (ML). “Teoría y Práctica para la Mejora de la Enseñanza del ML Aplicado a la Ciencia de Datos"Ver tabla 5

1. Árboles de decisión
2. Métodos de regresión
3. Métodos de clasificación
4. Métodos de predicción
5. Clustering
6. Sistemas de archivos distribuidos (HDFS)

Las herramientas utilizadas en el diplomado fueron: Anaconda Python, Scikit-Learn, Matplotlib, Dask, HDFS, entre otros.

Al finalizar el primer curso, donde se realizó la intervención en ML, se observó que el 50 % de los asistentes, de un total de 40, tenían diversos problemas de práctica resueltos.

Estos problemas se muestran como porcentajes de prácticas resueltas en la Fig 6.4. En esta figura, se comparan los resultados esperados (según nuestras experiencias en cursos anteriores) frente a los resultados reales, como prácticas resueltas y entregadas por los asistentes. Además, la figura muestra la eficiencia de la enseñanza según el siguiente criterio:

$$\%eficiencia = 100 * (prcticas\ resultas - practicas\ esperadas)/practicas\ esperadas \tag{6.1}$$

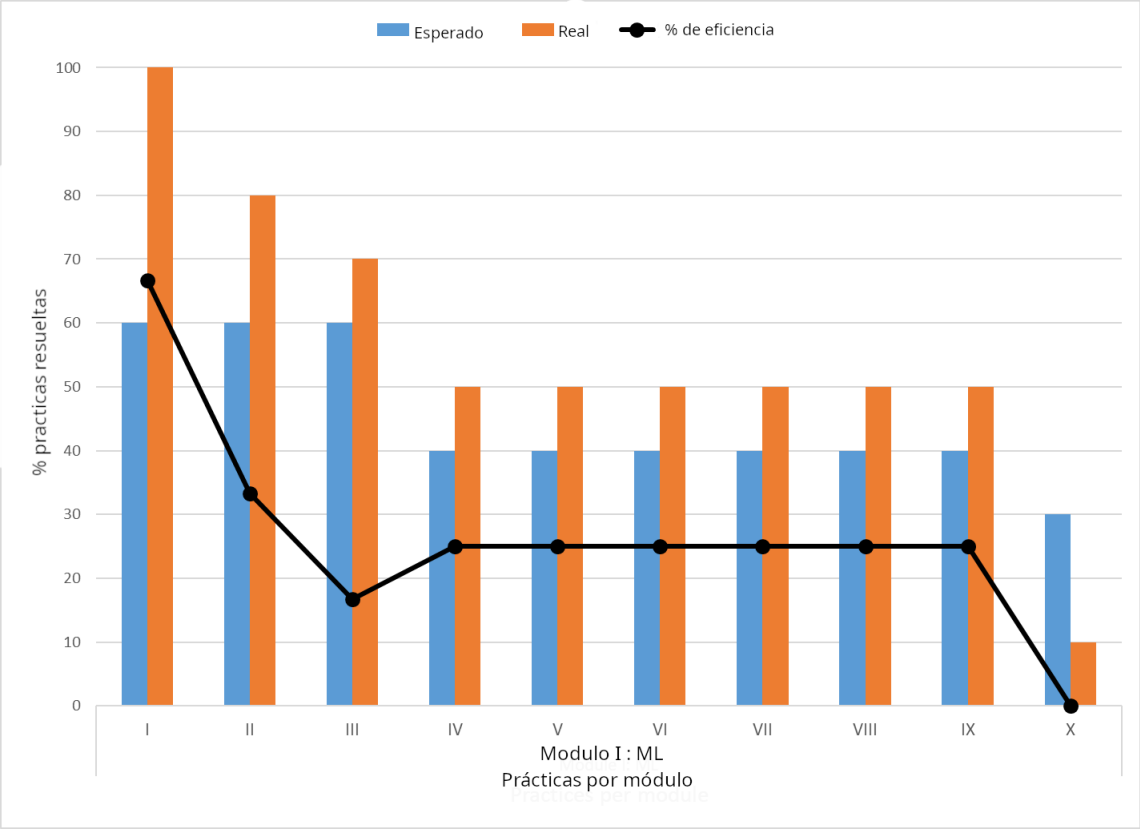

Figura 6.4: Resultados del Módulo I

También se observó que los estudiantes dejaron de trabajar en las prácticas más complejas de ML. Las principales razones que se identificaron fueron el aumento de las tareas de análisis de datos, además de tener que aplicar estadística y teoría matemática utilizando un lenguaje de programación (Python). La solución a estos problemas es dar mayor prioridad a la práctica con datos reales que a la teoría abstracta.

En trabajos similares a este, la enseñanza de ML se utiliza solo como un caso de uso para abordar educación en otros temas en casos reales; de hecho, con el enfoque de IA, pero no se realizan mejoras en la enseñanza de ML en sí, ni experiencias sobre cómo mejorar la enseñanza de la ciencia de datos.

La propuesta práctica en este trabajo permite establecer un currículo más completo y amplio, en cuanto a que incluye no solo el ML sino también el DL, el Big Data y las herramientas informáticas asociadas con la ciencia de datos.

Esta propuesta aún está en desarrollo y entre otras cuestiones, es necesario evaluar la eficiencia de la enseñanza de acuerdo con este enfoque práctico en un curso de DL, así como incluir otras herramientas que pueden ayudar a facilitar el aprendizaje de la ciencia de datos. Además de incluir plataformas online orientadas a la enseñanza así como otras herramientas que pueden ayudar facilitar el entendimiento de la ciencia de datos, como las plataformas de aprendizaje colaborativo en la nube.

Apéndices

Los anexos consisten en:

- Prácticas
- Datos
- Artículo

.1. Prácticas

Las prácticas se encuentran en Repositorio GitHub

.2. Datos

Los datasets que se usaron en las prácticas mostradas anteriormente, se pueden descargar de los siguientes links.

- Pasajeros del Titanic.
  https://www.kaggle.com/c/titanic/data
- California Housing.
  https://www.kaggle.com/fedesoriano/california-housing-prices-data-extra-features
- Pozos profundos del lago de Cuitzeo
  https://drive.google.com/file/d/19WVDYOM1xbF2hvbo75MDGe7OMN1clGhK/view?usp=shar
- Online Retail K-means & Hierarchical Clustering.
  https://www.kaggle.com/hellbuoy/online-retail-customer-clustering
- Anime Recommendation Database 2020.
  https://www.kaggle.com/hernan4444/anime-recommendation-database-2020
- RUOA de 2015 a la actualidad.
  https://ruoa.unam.mx/index.php?page=estaciones&id=9

- 100,000 UK Used Car Data set.
  https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes
- Información de datos públicos.
  Por decidirse

.3. Artículo

Heberto Ferreira Medina <hferreira@cieco.unam.mx>

Acceptance Notification : Intelligent Systems Conference (IntelliSys) 2022

IntelliSys Conference <IntelliSys@saiconference.com> 24 de febrero de 2022, 5:27
Para: Heberto Ferreira Medina <hferreira@iies.unam.mx>, stinoco@enesmorelia.unam.mx, luis.cendejas@ut-morelia.edu.mx, fhernandez@enesmorelia.unam.mx, michellmonroy18@gmail.com, bruginrod@gmail.com

Dear Heberto Ferreira Medina, Sergio Rogelio Tinoco-Martínez, José Luis Cendejas-Valdez, Froylán Hernández-Rendón, Mariana Michell Flores-Monroy, Bruce Hiram Ginori-Rodríguez,

Congratulations, your submitted paper "How to Improve the Teaching of Computational Machine Learning Applied to Large-Scale Data Science: The Case of Public Universities in Mexico" has been reviewed and accepted for presentation at the Intelligent Systems Conference (IntelliSys) 2022, to be held from 1-2 September 2022 in Amsterdam, The Netherlands.

We are of course aware that the situation regarding COVID-19 is a cause for apprehension - virtual participation (with reduced registration) is available, for anyone who cannot or chooses not to travel.

Intelligent Systems Conference (IntelliSys) 2022 will focus on areas of intelligent systems and artificial intelligence and how it applies to the real world. IntelliSys provides a leading international forum that brings together researchers and practitioners from diverse fields with the purpose of exploring the fundamental roles, interactions as well as practical impacts of Artificial Intelligence.

In order to attend, present and publish your paper, please register online at https://saiconference.com/IntelliSys2022/Register (Registration closes March 15, 2022). We also accept fees via bank/wire transfer.

IntelliSys 2022 proceedings will be published in the Springer series "Lecture Notes in Networks and Systems" and submitted for consideration to Web of Science, SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH and SCImago.

The Conference Board has decided that the reviewers' feedback and invitation letter for visa applications (if necessary) will be emailed to the author(s) after the registration process. If you would like to receive the reviewer feedback for representation at your university/organization, please feel free to contact us.

Again, congratulations and I look forward to your participation!

Regards,
Kohei Arai
Program Chair
IntelliSys Conference

View IntelliSys 2021 Recap

Referencias

Abbasi, B., y Goldenholz, D. M. (2019). Machine Learning Applications in Epilepsy. En (Vol. 60, pp. 2037–2047). Wiley Online Library.

Adithiyaa, T., Chandramohan, D., y Sathish, T. (2020). Optimal prediction of process parameters by gwo-knn in stirring-squeeze casting of aa2219 reinforced metal matrix composites. *Materials Today: Proceedings*, *21*, 1000–1007.

Aher, S. B., y Lobo, L. (2012). A Comparative Study of Association Rule Algorithms for Course Recommender System in E-learning. *International Journal of Computer Applications*, *39*(1), 48–52.

Ahmad, M., Al-Shayea, N. A., Tang, X.-W., Jamal, A., M Al-Ahmadi, H., y Ahmad, F. (2020). Predicting the pillar stability of underground mines with random trees and c4. 5 decision trees. *Applied Sciences*, *10*(18), 6486.

Alvaredo, F. (2011). A note on the relationship between top income shares and the gini coefficient. *Economics Letters*, *110*(3), 274–277.

Ayodele, T. O. (2010). Types of Machine Learning Algorithms. *New Advances in Machine Learning*, *3*, 19–48.

Barnston, A. G. (1992). Correspondence among the correlation, rmse, and heidke forecast verification measures; refinement of the heidke score. *Weather and Forecasting*, *7*(4), 699–709.

Borja-Robalino, R., Monleón-Getino, A., y Rodellar, J. (2020). Estandarización de métricas de rendimiento para clasificadores machine y deep learning. *Revista Ibérica de Sistemas e Tecnologias de Informação*(E30), 184–196.

Borthakur, D., y cols. (2008). Hdfs architecture guide. *Hadoop apache project*, *53*(1-13), 2.

Brijain, M., Patel, R., Kushik, M., y Rana, K. (2014). A Survey on Decision Tree Algorithm for Classification.

Celebi, M. E., y Aydin, K. (2016). *Unsupervised Learning Algorithms*. Springer.

Chai, T., y Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, *7*(3), 1247–1250.

Chicco, D., y Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1–13.

Cintra, M. E., Monard, M. C., y Camargo, H. A. (2013). A fuzzy decision tree algorithm based on c4. 5. *Mathware & soft computing*, *20*(1), 56–62.

Cunningham, P., Cord, M., y Delany, S. J. (2008). Supervised Learning. En *Machine Learning Techniques For Multimedia* (pp. 21–49). Springer.

Davis, J., y Goadrich, M. (2006). The relationship between precision-recall and roc curves. En *Proceedings of the 23rd international conference on machine learning* (pp. 233–240).

Dayan, P., Sahani, M., y Deback, G. (1999). Unsupervised Learning. *The MIT Encyclopedia of The Cognitive Sciences*, 857–859.

Derczynski, L. (2016). Complementarity, f-score, and nlp evaluation. En *Proceedings of the tenth international conference on language resources and evaluation (lrec'16)* (pp. 261–266).

El Naqa, I., y Murphy, M. J. (2015). What is Machine Learning? En *Machine Learning in Radiation Oncology* (pp. 3–11). Springer.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Goutte, C., y Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. En *European conference on information retrieval* (pp. 345–359).

Haenlein, M., y Kaplan, A. (2019). A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *California Management Review*, *61*(4), 5–14.

Harrell, F. E. (2015). Ordinal logistic regression. En *Regression modeling strategies* (pp. 311–325). Springer.

Huang, H., Xu, H., Wang, X., y Silamu, W. (2015). Maximum f1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(4), 787–797.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., y Li, B. (2018). Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. En *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 19–35).

Jannach, D., Zanker, M., Felfernig, A., y Friedrich, G. (2010). *Recommender systems: An introduction*. Cambridge University Press.

Jiang, T., Gradus, J. L., y Rosellini, A. J. (2020). Supervised Machine Learning: A Brief Primer. *Behavior Therapy*, *51*(5), 675–687.

Jordan, M. I., y Mitchell, T. M. (2015). Machine Learning: Trends, Perspectives, and Prospects. *Science*, *349*(6245), 255–260.

Kaelbling, L. P., Littman, M. L., y Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., y Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, *24*(7), 881–892.

Kaplan, J. (2016). *Artificial Intelligence: What Everyone Needs to Know*. Oxford University Press.

Karun, A. K., y Chitharanjan, K. (2013). A review on hadoop—hdfs infrastructure extensions. En *2013 ieee conference on information & communication technologies* (pp. 132–137).

Kellman, P., y Hansen, M. S. (2014). T1-mapping in the heart: accuracy and precision. *Journal of cardiovascular magnetic resonance*, *16*(1), 1–20.

Kotsiantis, S. B., Zaharakis, I., y Pintelas, P. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*(1), 3–24.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., y Fotiadis, D. I. (2015). Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal*, *13*, 8–17.

Kumari, K., Yadav, S., y cols. (2018). Linear regression analysis study. *Journal of the practice of Cardiovascular Sciences*, *4*(1), 33.

Lee, A., Taylor, P., Kalpathy-Cramer, J., y Tufail, A. (2017). Machine Learning Has Arrived. *Ophthalmology*, *124*(12), 1726–1728.

Li, W., Han, J., y Pei, J. (2001). CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. En *Proceedings 2001 IEEE International Conference on Data Mining* (pp. 369–376).

Libbrecht, M. W., y Noble, W. S. (2015). Machine Learning Applications in Genetics and Genomics. *Nature Reviews Genetics*, *16*(6), 321–332.

Liu, Y. H. (2020). *Build Intelligent Systems Using Python, TensorFlow2, PyTorch and Scikit-learn*. Packt Publishing Ltd.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14–23.

Loh, W.-Y. (2014). Fifty years of classification and regression trees. *International Statistical Review*, *82*(3), 329–348.

López, F. J. A., Avi, J. R., y Fernández, M. V. A. (2018). Control estricto de matrices de confusión por medio de distribuciones multinomiales. *Geofocus: Revista Internacional de Ciencia y Tecnología de la Información Geográfica*(21), 6.

Maulud, D., y Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, *1*(4), 140–147.

McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica*, *23*(2), 143–149.

Mitchell, T. (1997). Does Machine Learning Really Work? *AI Magazine*, *18*(3), 11–11.

Mitchell, T., y cols. (1997). Machine Learning. McGraw-hill New York.

Mohri, M., Rostamizadeh, A., y Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT Press.

Montgomery, D. C., Peck, E. A., y Vining, G. G. (2021). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., y Yu, B. (2019). Definitions, Methods, and Applications in Interpretable Machine Learning. *Proceedings of the National Academy of Sciences*, *116*(44), 22071–22080.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., y Brown, S. D. (2004). An Introduction to Decision Tree Modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), 275–285.

Na, S., Xumin, L., y Yong, G. (2010). Research on K-means Clustering Algorithm: An Improved K-means Clustering Algorithm. En *2010 Third International Symposium on Intelligent Information Technology and Security Informatics* (pp. 63–67).

Pandis, N. (2016). The Chi-square Test. *American journal of orthodontics and dentofacial orthopedics*, *150*(5), 898–899.

Pérez, J., Henriques, M., Pazos, R., Cruz, L., Reyes, G., Salinas, J., y Mexicano, A. (2007). Mejora al algoritmo de agrupamiento k-means mediante un nuevo criterio de convergencia y su aplicación a bases de datos poblacionales de cáncer. *Liver-2do Taller Latino Iberoamericano de Investigacion de Operaciones "la IO aplicada a la solución de problemas regionales*, 1–7.

Raschka, S., y Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.

Rodríguez, A. H., Avilés-Jurado, F. X., Díaz, E., Schuetz, P., Trefler, S. I., Solé-Violán, J., ... others (2016). Procalcitonin (PCT) Levels for Ruling-out Bacterial Coinfection in ICU Patients with Influenza: a CHAID Decision-Tree Analysis. *Journal of infection*, *72*(2), 143–151.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Sathya, R., y Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence*, *2*(2), 34–38.

Siau, K., y Wang, W. (2018). Building Trust in Artificial Intelligence, Machine Learning, and Robotics. *Cutter Business Technology Journal*, *31*(2), 47–53.

Singh, S., y Gupta, P. (2014). Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, *27*(27), 97–103.

Song, Y., Huang, J., Zhou, D., Zha, H., y Giles, C. L. (2007). Iknn: Informative k-Nearest Neighbor Pattern Classification. En *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 248–264).

Su, J., y Zhang, H. (2006). A Fast Decision Tree Learning Algorithm. En *AAAI* (Vol. 6, pp. 500–505).

Sutton, R. S., y Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Timofeev, R. (2004). Classification and regression trees (cart) theory and applications. *Humboldt University, Berlin*, 1–40.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433.

van Zoonen, W., y Toni, G. (2016). Social Media Research: The Application of Supervised Machine Learning in Organizational Communication Research. *Computers in Human Behavior*, *63*, 132–141.

Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., y Koudas, N. (2002). Nonlinear Dimensionality Reduction Techniques for Classification and Visualization. En *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 645–651).

Wang, S.-C. (2003). Artificial Neural Network. En *Interdisciplinary Computing in Java Programming* (pp. 81–100). Springer.

Wiering, M., y Van Otterlo, M. (2012). Reinforcement Learning. *Adaptation, Learning, and Optimization*, *12*(3).

Zhang, S., Li, X., Zong, M., Zhu, X., y Cheng, D. (2017). Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), 1–19.