



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 FEBRERO 2023



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 agradecimiento a todo el cuerpo docente de la Licenciatura en Tecnologías para la Información en Ciencias, por su tiempo, su paciencia y por todas sus enseñanzas tanto dentro como fuera del ámbito académico, especialmente a las Dras. Marisol Flores Garrindo y Adriana Menchaca Méndez por ayudar a no rendirme y siempre brindarme su apoyo en cada etapa de mi trayectoria universitaria.

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

Agradezco al proyecto **PAPIME PE106021** que me ayudó económicamente durante el desarrollo de este trabajo ya que, sin este apoyo, me hubiera sido imposible concluirlo.

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.

De la misma manera agradezco muchísimo al Dr. Heberto Ferreira 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 hicieron todos y cada uno de ellos para que yo pueda llegar hasta aquí. Agradezco a mi mamá y a mi papá quienes, aunque no entendían bien de qué trata mi carrera, no dejaron de creer 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 en mí.

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 (LTICs) 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 estudiantes y docentes de las diferentes licenciaturas, dentro de la ENES Morelia, con respecto a cuáles 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 objetivo 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.

Las prácticas realizadas a lo largo de este proyecto (como libretas de JupyterLab) se encuentran en el siguiente repositorio en la plataforma **GitHub**:

<https://github.com/MichellMonroy/Practicas-ML>

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 computing laboratory manual for undergraduate students is proposed in order to improve the knowledge acquisition of Machine Learning (ML) and its application onto a large scale Data Science (Big Data).

The project is focused on designing, building and implementing a manual of computer laboratory practices aimed at students from the sixth semester onwards of the Bachelor of Information Technology in Sciences (LTICs) and/or other degrees from ENES Morelia with basic knowledge of the ML.

To carry out this project, opinion of students and teachers of 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 subjects based on ML of the LTICs curriculum.

The objective of this project is to improve the academic training of students, within the LTICs, as well as to improve the quality of teaching of this knowledge area by the academic staff.

The practices carried out throughout this project (as JupyterLab notebooks format) are found in the following repository on the **GitHub** platform:

<https://github.com/MichellMonroy/Practicas-ML>

Índice general

Agradecimientos institucionales	I
Agradecimientos personales	II
Resumen	III
Abstract	IV
1. Introducción	1
1.1. Justificación	2
1.2. Hipótesis	3
1.3. Objetivo	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Particulares	3
1.4. Descripción general	5
2. Antecedentes	6
2.1. Tipos de Machine Learning	7
2.1.1. Aprendizaje Supervisado	8
2.1.2. Árboles de decisión.	9
2.1.3. k -vecinos más cercanos.	9
2.1.4. Redes neuronales artificiales (RNA).	9
2.1.5. Aprendizaje No Supervisado	12
2.1.6. Aprendizaje Por Refuerzo	13

2.2. Uso del ML en la actualidad	14
2.3. Diseño de la encuesta	15
2.4. Ejes y preguntas	16
3. Algoritmos de Machine Learning	24
3.1. Árboles de decisión	24
3.1.1. CHi-squared Automatic Interaction Detector (CHAID)	25
3.1.2. Classification And Regression Tree (CART)	26
3.1.3. Algoritmo C4.5	27
3.2. Modelos de Regresión	28
3.2.1. Regresión lineal	28
3.2.2. Regresión logística	29
3.3. k -vecinos más cercanos (KNN)	30
3.4. Clustering (k -medias)	33
3.5. Dask	36
3.6. Sistema de archivos HDFS	37
4. Métricas para Evaluar Modelos	39
4.1. RMSE	39
4.2. MAE	40
4.3. Matriz de confusión	40
4.4. Exactitud	41
4.5. Precisión	42
4.6. Recall (Sensibilidad)	43
4.7. Métrica F_β	44
4.7.1. Métrica F_1	44
4.7.2. Métricas $F_{0.5}$, F_1 y F_2	45
5. Metodología de las Prácticas	46
6. Resultados y Discusión de la encuesta	48
6.1. Prueba Piloto	48

Apéndices	51
.1. Prácticas	52
.2. Datos	52
.3. Artículo	53

Índice de figuras

1.1. Mapa mental del desarrollo del proyecto.	4
2.1. Diagrama del aprendizaje supervisado.	8
2.2. Ejemplo de los componentes de una RNA.	10
2.3. Ejemplo de una RNA completamente conectada, cada nodo es una neurona que realiza el procedimiento señalado en el texto. En ella se puede ver que la salida de una capa es la entrada de la siguiente capa.	11
2.4. Ejemplo de aprendizaje supervisado usado para clasificación de spam.	11
2.5. Ejemplo de aprendizaje no supervisado usado para clustering, basado en los atributos de los datos.	12
2.6. Diagrama de un algoritmo de aprendizaje por refuerzo.	13
2.7. Diagrama de los campos de estudio dentro de la IA.	15
2.8. Matriz de correlación de los resultados obtenidos en la encuesta, donde los encabezados son las preguntas realizadas en la encuesta ver Tabla 2.2.	19
2.9. Nivel de importancia de ejes de acuerdo a la población encuestada. .	20
2.10. Nivel de importancia de temas de acuerdo a la población encuestada.	21
2.11. Nivel de importancia de herramientas de acuerdo a la población encuestada.	22
3.1. Ejemplo de una recta calculada con regresión lineal donde $y = 2.2816 + 0.3464x$.	29
3.2. Función Sigmoide aplicada a la regresión lineal.	30
3.3. Nuevo punto O a clasificar.	31

3.4. Vecinos más cercanos al punto O (con menor distancia).	31
3.5. Vecinos más cercanos (con $k = 3$).	32
3.6. $k = 3$ centroides en un conjunto de datos.	34
3.7. Buscando el centroide más cercano.	34
3.8. Etiquetado del punto actual de acuerdo a su centroide más cercano. .	35
3.9. k grupos formados.	35
3.10. Sintaxis de Pandas (arriba) y sintaxis de Dask (abajo).	36
3.11. Sintaxis de NumPy (izquierda) y sintaxis de Dask (derecha).	37
3.12. Diagrama de un HDFS.	37
4.1. Matriz de confusión simple de dos valores (positivo/negativo).	41
4.2. Representación gráfica de la exactitud vs. precisión.	43
6.1. Resultados del Módulo I del Diplomado.	49

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 para la Información en Ciencias (LTICs), de la Escuela Nacional de Estudios Superiores Unidad Morelia (ENES Morelia), perteneciente a la Universidad Nacional Autónoma de México (UNAM) 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 tratar de poner en práctica la teoría aprendida, además de carecer de las herramientas para su aplicación en problemas reales.

En virtud de lo anterior, se observa la necesidad de que docentes y estudiantes de la LTICs conozcan nuevas fronteras en Inteligencia Artificial (IA), específicamente en la aplicación de modelos matemáticos del aprendizaje automático (*ML – Machine Learning*).

El 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 mejora su rendimiento en el futuro.

Por otro lado, el Internet de las Cosas (*IoT – Internet of Things*) 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 bancos en los cuales 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, abrir una cuenta, etc.

El asistente virtual del banco procesa texto y voz para mejorar la interacción con los clientes, hace uso de datos y del ML para procesar la información que convierte. El agente inteligente utiliza algoritmos de IA 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. 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 temas difíciles.

1.2. Hipótesis

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 volúmenes de datos 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.

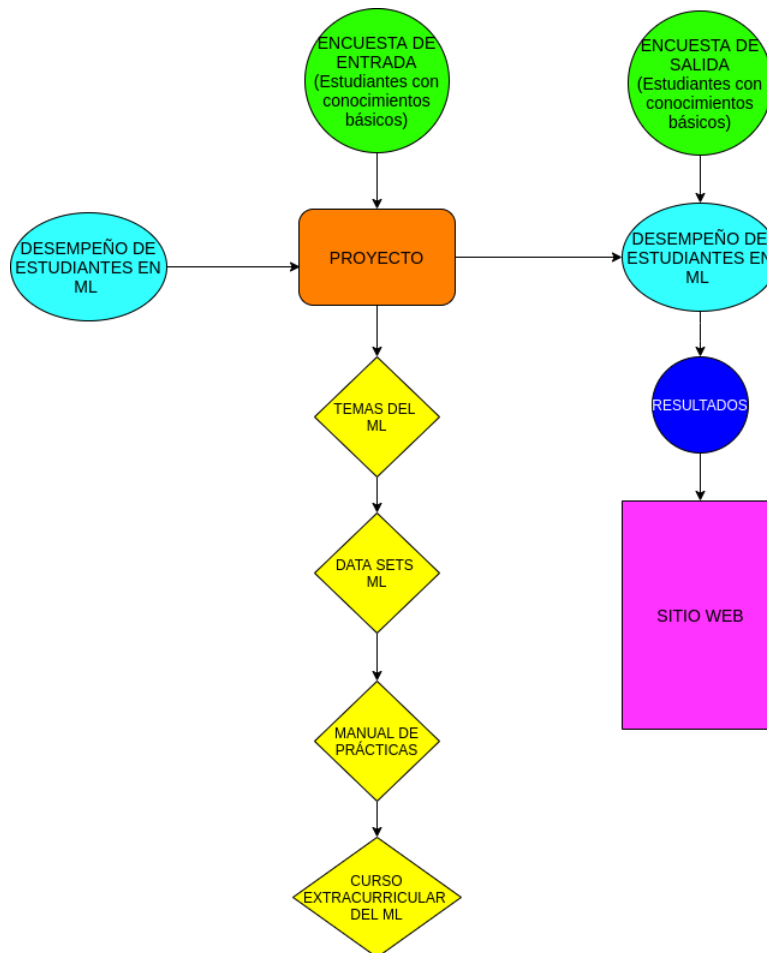


Figura 1.1: Mapa mental del desarrollo del proyecto.

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.
- En el Capítulo 3 se detalla de forma más técnica y teórica el funcionamiento de los algoritmos que se usarán en las prácticas (Capítulo 5). En éste 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 de archivos distribuidos (como el *HDFS – Hadoop File System*).
- El Capítulo 4 habla sobre las principales métricas de evaluación para modelos de ML. En éste se explica la razón de cada métrica, además de mostrar las fórmulas 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 éste 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 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 la LTICs y académicos con formación afín a éste.

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.

Posteriormente 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 *aprendizaje automático* (*ML – Machine Learning*) cuyo fin es mejorar el rendimiento de los algoritmos a través de su experiencia. Esta técnica hace uso de herramientas como la estadística, la informática, las matemáticas y las ciencias computacionales, teniendo su fundamento en el análisis de los datos.

Por otra parte, la definición de Mitchell y cols. (1997) del ML es la siguiente:

Se dice que un programa de computadora aprende de una experien-

cia 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 Samuel (1959), 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 la computadora la habilidad de aprender, sin que esté explícitamente programada.

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 éstos 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. Tales enfoques se dividen en:

- Si están entrenados bajo supervisión humana o no. A los cuales se les denominan: **supervisado, no supervisado y por refuerzo.**
- Si pueden aprender sobre la marcha o no, 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**.

2.1.1. Aprendizaje Supervisado

El aprendizaje supervisado suele usarse cuando se cuenta con datos de los cuales 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.

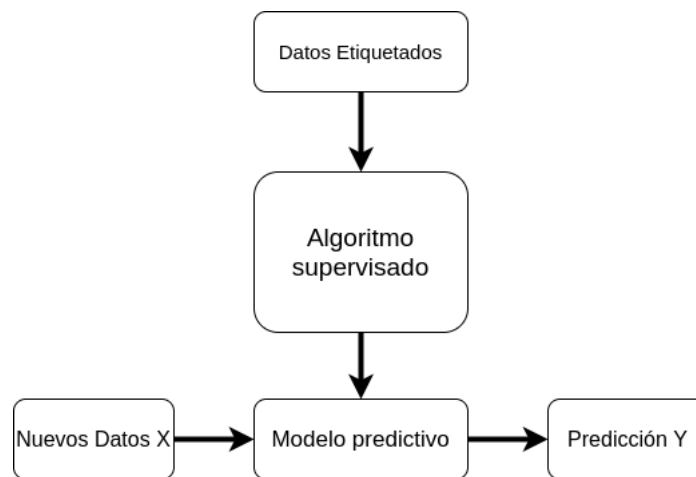


Figura 2.1: Diagrama del aprendizaje supervisado.

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

De acuerdo con Ayodele (2010) el aprendizaje supervisado tiene la tarea de resolver los siguientes problemas, los cuales no pueden resolverse con programación simple:

- **Regresión.**

Jagielski y cols. (2018) definen la regresión como un método en el cual se hace uso de variables numéricas, para realizar predicciones, las cuales se espera que cada vez tengan menor margen de error. Montgomery y cols. (2021) señalan que estas variables se estudian para encontrar correlación entre ellas y sus respectivas

etiquetas, para realizar predicciones de acuerdo a los patrones encontrados. Existen dos tipos principales de regresión: lineal y logística.

■ **Clasificación.**

Li y cols. (2001) indican que 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).

Para complementar lo anterior, Kotsiantis y cols. (2007) mencionan los siguientes algoritmos que se usan para clasificación (entre otros).

2.1.2. 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 se escoge el atributo cuyo valor es más adecuado para dividir el conjunto de datos, hasta que todos los datos sean clasificados.

2.1.3. k -vecinos más cercanos.

Song y cols. (2007) afirman que este algoritmo tiene la tarea de predecir la etiqueta de un dato (x_0) dados los k datos más cercanos, es decir, aquéllos 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.

2.1.4. 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, siendo la más común la función sigmoide.

Zou y cols. (2009) explica que una red neuronal se compone de los siguientes elementos, donde: x son los datos de entrada, w son los pesos que se le asigna a cada componente de x , f (una función de agregación) el nodo (neurona) que realiza operaciones a los datos de entrada, σ es la función de activación (Sigmoide, ReLu, etc.), ver la Figura 2.2.

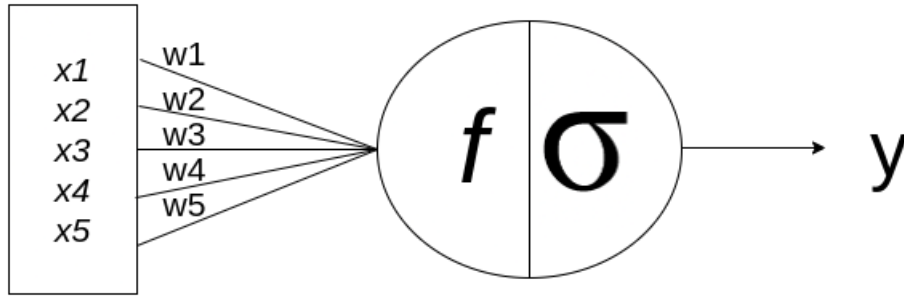


Figura 2.2: Ejemplo de los componentes de una RNA.

En cada iteración del entrenamiento de la red neuronal, f calcula la suma del producto de cada componente en x con su respectivo peso w , esto se pasa a la función de activación (σ) para obtener la salida y :

$$y = f \left(\sum_{i=0}^n w_i x_i - T \right) \quad (2.1)$$

Donde T es el umbral que determina si la salida es 0 o 1 (para clasificación binaria)

La decisión de si la salida actual pasa a ser 0 o 1 de acuerdo al umbral, es la función escalón:

$$y = \begin{cases} 0 & \text{if } \sum_{i=0}^n w_i x_i \geq T \\ 1 & \text{if } \sum_{i=0}^n w_i x_i < T \end{cases} \quad (2.2)$$

Este modelo se vuelve más complicado cuando se le agregan más neuronas o nodos. Al conjunto de varias neuronas conectadas entre sí, se les denomina *capa*. Una capa es un conjunto de neuronas cuyas entradas son el resultado de una capa anterior y cuyas salidas serán la entrada de una capa posterior.

Un ejemplo de una red neuronal multicapa se aprecia en la Figura 2.3

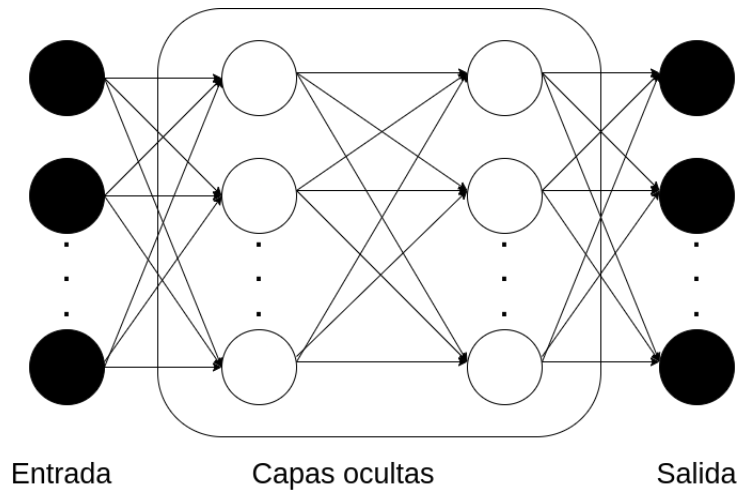


Figura 2.3: Ejemplo de una RNA completamente conectada, cada nodo es una neurona que realiza el procedimiento señalado en el texto. En ella se puede ver que la salida de una capa es la entrada de la siguiente capa.

En cada iteración de procesamiento o *época* se busca ajustar cada uno de los pesos de las entradas de todas las neuronas, para que la salida de cada capa se ajuste cada vez más a los datos que ya conocemos, de esta forma la red neuronal va aprendiendo.

Entre las aplicaciones de la clasificación, por ejemplo, está la detección de correo electrónico *spam* (correo no deseado), como se ve en la Figura 2.4.

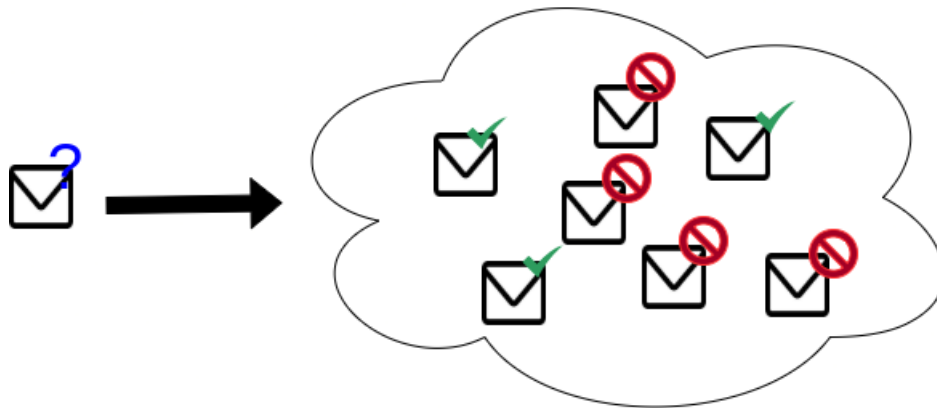


Figura 2.4: Ejemplo de aprendizaje supervisado usado para clasificación de spam.

2.1.5. 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, según señalan 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, 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 2.5.

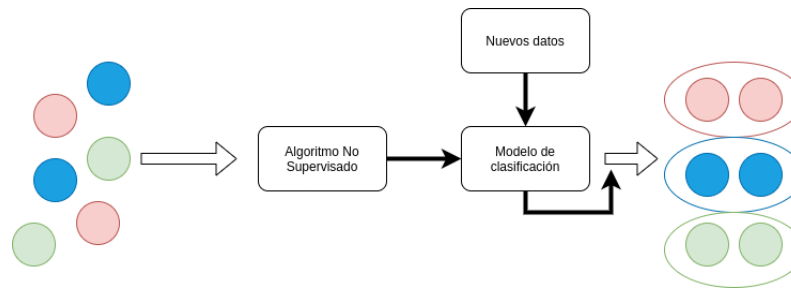


Figura 2.5: Ejemplo de aprendizaje no supervisado usado para clustering, basado en los atributos de los datos.

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 Prin-

cipales (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 éstos.

2.1.6. 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 2.6 se puede ver, de manera muy general, el comportamiento del aprendizaje por refuerzo.

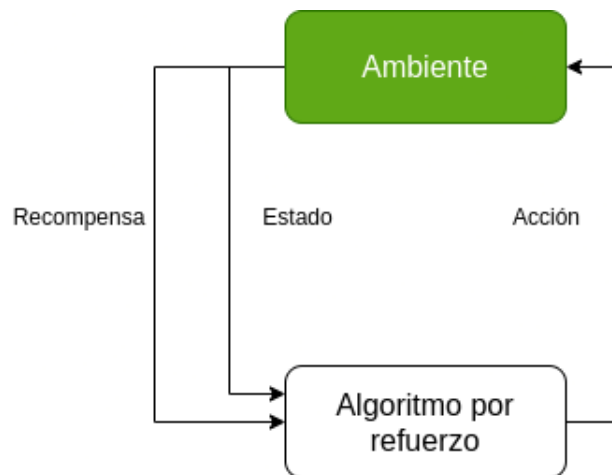


Figura 2.6: Diagrama de un algoritmo de aprendizaje por refuerzo.

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.
- *Felipe 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 del ML en la actualidad

En nuestros días, una de las principales aplicaciones del ML 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.

Por otro lado, Siau y Wang (2018) apuntan la utilidad del aprendizaje supervisado en el campo de la computación y las matemáticas, como el desarrollo de Google DeepMind y Alpha Go. Además de que 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 Figura 2.7 se puede apreciar un diagrama a grandes rasgos de que el ML es un campo de estudio dentro de la Inteligencia Artificial.

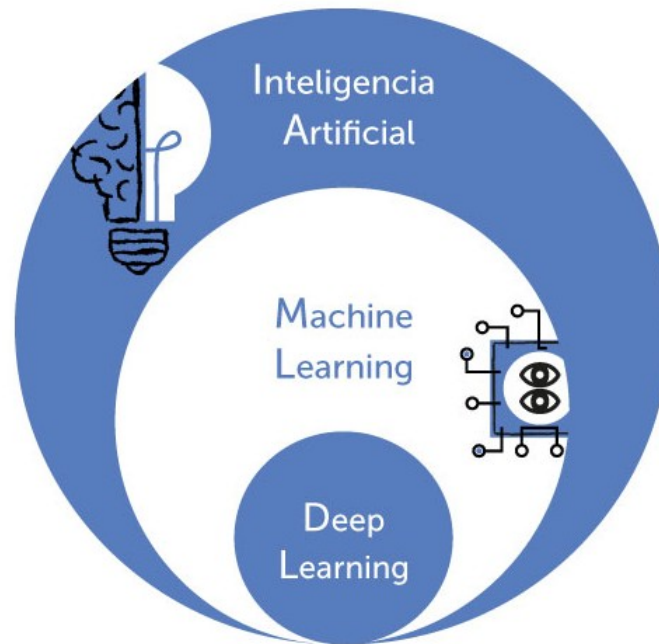


Figura 2.7: Diagrama de los campos de estudio dentro de la IA.

2.3. Diseño de la encuesta

Esta investigación se caracterizó por ser un estudio de tipo:

1. Exploratorio
2. Descriptivo
3. Correlacional
4. Pre – experimental

con la finalidad de contar con un estudio de caso a través de una sola medición. Para ello se generó una encuesta que fue aplicada a una población conformada por profesores, estudiantes e investigadores de la Universidad Nacional Autónoma de México campus Morelia, Michoacán.

La encuesta fue creada usando una plataforma en línea encuesta.com, (2021) y distribuida por correo electrónico (Gmail principalmente) y redes sociales a personas seleccionadas al azar entre la población de estudio de la universidad mencionada

anteriormente. En primer lugar, la muestra se calculó mediante el método de población finita con base a 600 personas. Esta muestra tiene un intervalo de confianza del 95 % y un margen de error del 10 %, como se muestra en la Tabla 2.1

Descripción	Valor
Tamaño de la población	600
Nivel de confianza	95 %
Margen de error	10 %
Tamaño de la muestra	83

Tabla 2.1: Muestra de la población.

La encuesta consistió en ocho preguntas generadas a partir de una revisión crítica de la literatura relacionada con la ciencia de datos de ML, DL y Big Data. Se consultaron a expertos y expertas en el área de ML, cuatro maestros titulares de la ENES Morelia, tres del Tecnológico Nacional de México campus Morelia, dos del Instituto de Investigaciones en Ecosistemas y Sustentabilidad (IIES, UNAM) y uno del Instituto de Materiales UNAM.

Ellas y ellos validaron las preguntas seleccionadas, mencionando que se deberían considerar los siguientes puntos:

- Dar diferentes opciones para responder.
- Poner respuestas dicotómicas.
- Usar escala de Likert.

2.4. Ejes y preguntas

Gracias a las observaciones, la encuesta se perfeccionó dividiéndola en cuatro ejes principales: *Eje I: Machine Learning*; *Eje II: Deep Learning*; *Eje III: Big Data*; y *Eje IV. Herramientas*.

La escala Likert aplicada para los primeros tres ejes fue: Muy importante (*VImp* - *Very Important*), Importante (*Imp* - *Important*), Neutral, Poco importante (*LImp* -

Less Important), Nada importante (*NImp* - *Nothing Important*), Lo desconozco (*dnK* - *Do not Know*); ver Tabla 2.2.

Mientras que para el cuarto eje se usó el nivel de habilidad que se tiene para diferentes herramientas de programación *bajo*, *medio*, *alto*.

Como se puede observar, cada escala se abrevió por sus siglas en inglés para que se pueda entender mejor.

#	Descripción de la pregunta	Eje	Tipo de pregunta
Q1	Nº de cuenta o empleado UNAM, tu nombre si no cuentas con ellos	-	Opción
Q2	Licenciatura que estudia; 2.1 Ciencias, 2.2 Agroforestales, 2.3 Ciencias Ambientales, 2.4 Ciencia de Materiales Sustentables, 2.5 Ecología, 2.6 Estudios Sociales y Gestión Local, 2.7 Geociencias, 2.8 Geohistoria, 2.9 Tecnologías para la Información en Ciencias, 2.10 Otro (Other)	-	Opción
Q3	Semestre que cursas (1-10), no aplica para profesores	-	Número
Q4	Consideras que los siguientes temas relacionados con el aprendizaje automático (Machine Learning) son: 4.1 Data Science, 4.2 Web Scraping, 4.3 Data Wrangling, 4.4 Machine Learning, 4.5 Data Mining, 4.6 Ensemble Learning, 4.7 Data visualization, 4.8 ML: supervised/unsupervised, 4.9 Binary and multiclass classification, 4.10 EDA, 4.11 Clustering, 4.12 ML model, 4.13 ML evaluation: underfitting, overfitting, 4.14 Cross validation, 4.15 Hyperparameters, regularization, feature engineering, 4.16 PCA	I	Opciones Likert
Q5	Consideras que los siguientes temas relacionados con el aprendizaje profundo (Deep Learning) son: 5.1 NN Shallow & Deep, 5.3 CNN, 5.3 RNN, 5.4 Transfer Learning & Fine-Tuning, 5.5 Dropout, 5.6 Data Augmentation, 5.7 Batch Normalization	II	Opciones Likert
Q6	Consideras que los siguientes temas relacionados con los macrodatos (Big Data) son: 6.1 Concepto, 6.2 Escalado de modelos, 6.3 Analítica a gran escala, 6.4 Sistema de archivos distribuidos, 6.5 Map-Reduce	III	Opciones Likert
Q7	La habilidad que tienes en el manejo de las siguientes herramientas es: 7.1 Tensorflow, 7.2 Spark, 7.3 Keras, 7.4 Fast.ai, 7.5 PyTorch, 7.6 HDFS, 7.7 Kafka, 7.8 Python, 7.9 Scikit-Learn	VI	Opciones Likert
Q8	Consideras importante incluir algunos temas adicionales relacionados con el ML, DL y el Big Data, no mencionados anteriormente:	-	Abierta

Tabla 2.2: Encuesta aplicada.

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 2.8.

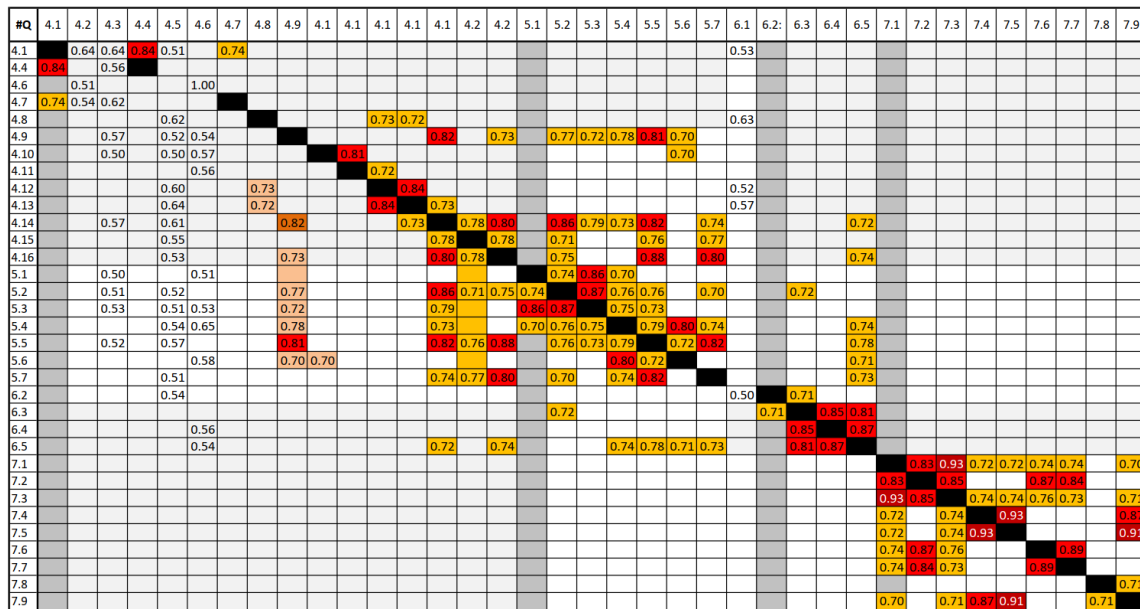


Figura 2.8: Matriz de correlación de los resultados obtenidos en la encuesta, donde los encabezados son las preguntas realizadas en la encuesta ver Tabla 2.2.

De acuerdo al análisis de correlaciones, se identificaron áreas de oportunidad según porcentaje de importancia que respondieron los encuestados. En la Figura 2.9 se muestra esta importancia.

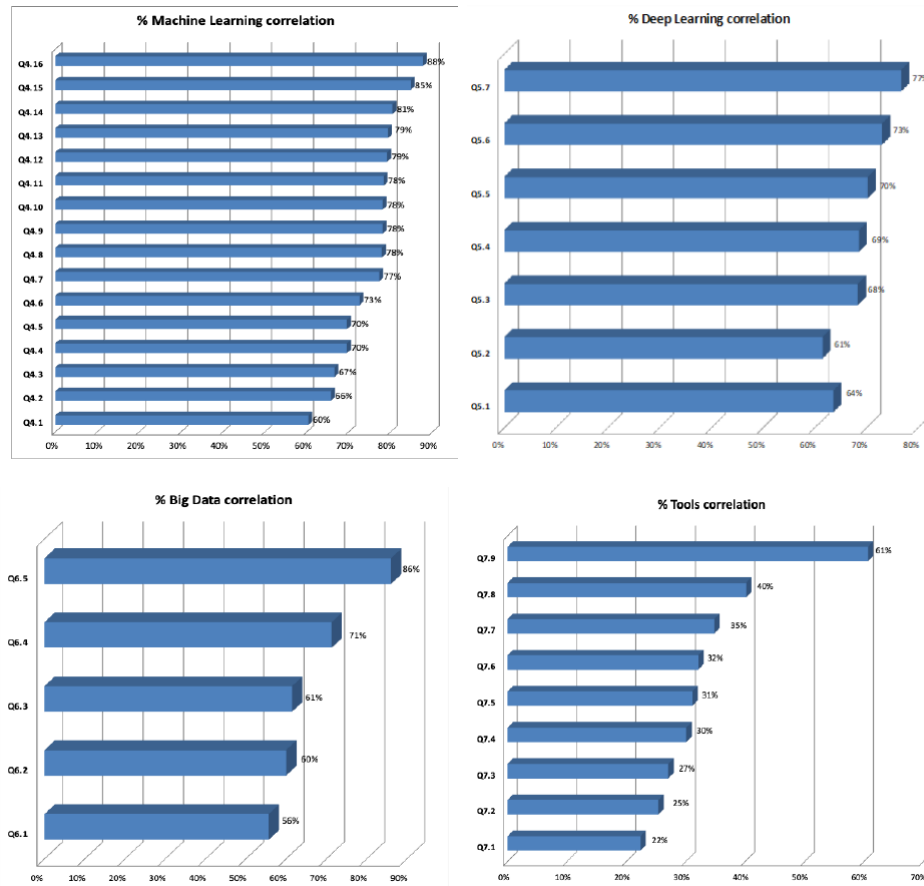


Figura 2.9: Nivel de importancia de ejes de acuerdo a la población encuestada.

De acuerdo con la encuesta desarrollada, se observó que los encuestados tenían cierto desconocimiento en algunos temas. En las Figuras 2.10 y 2.11 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 2.10: Nivel de importancia de temas de acuerdo a la población encuestada.



Figura 2.11: Nivel de importancia de herramientas de acuerdo a la población encuestada.

Como se observa en las Figuras, los temas que más desconocen las y los participantes (color verde) son los temas a desarrollar en el manual para obtener un mayor impacto académico, así mismo se tomó en cuenta la importancia que la población encuestada le dió a los temas, resultando en los siguientes temas.

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

De acuerdo a estos resultados se propone mejorar el aprendizaje con prácticas orientadas a reforzar estos temas. En la Tabla 5.1 se muestran las prácticas finalmente propuestas para mejorar en los temas que observamos como área de oportunidad. Cabe resaltar que se observó que las y los encuestadas (os) prefieren orientar la intervención hacia la aplicación práctica de los conocimientos.

En el siguiente capítulo se describen brevemente los temas correspondientes al ML determinado con fundamento en la encuesta aplicada, antes de proponer el esquema de las prácticas que conforman el manual

Debido a la demanda del conocimiento y manejo del ML es necesario tener una lista de temas a considerar para definir el número y la cantidad de prácticas a realizar. A continuación se muestran los principales temas a considerar.

Capítulo 3

Algoritmos de Machine Learning

En la Sección 2.1 se habló sobre los tipos de algoritmos del ML. En esta sección se profundizará en los algoritmos que se usaron en el manual de prácticas.

3.1. Árboles 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 inicie con un árbol vacío en el cual aún no hay nada de información acerca de los datos. Al ser un algoritmo avaricioso, éste busca cuál es el atributo que mejor divide el conjunto de datos y lo convierte en la raíz del árbol. Posteriormente el proceso se vuelve recursivo, dividiendo el conjunto de datos restante 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 la *prueba ji cuadrada* (χ^2). Pandis (2016) explica que esta prueba solo muestra si existe una asociación entre variables, es decir, mide qué tan dependiente es una variable de otra.

La forma de calcular χ^2 , mostrada en McHugh (2013), es la siguiente:

$$\chi^2 = \sum_i^j \frac{(O_i - E_i)^2}{E_i} \quad (3.1)$$

donde O_i son los valores observados y E_i es su frecuencia esperada.

Para calcular la frecuencia esperada (E) se emplea la fórmula siguiente:

$$E = \frac{M_R * M_C}{n} \quad (3.2)$$

donde M_R es la suma de la fila, M_C es la suma de la columna y n es el número total de datos.

La Ecuación 3.2 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. El método ayuda mucho cuando se trata de análisis estadísticos.

Por ejemplo, imagine que queremos conocer la frecuencia esperada del número de estudiantes que aprueban matemáticas, para ello tenemos:

	Aprobados	Reprobados
Español	35	6
Matemáticas	64	23
Historia	16	2

Tabla 3.1: Tabla con los datos crudos.

Lo primero que se tiene que hacer es calcular el total de cada fila y cada columna, como se muestra:

	Aprobados	Reprobados	Total
Español	35	6	41
Matemáticas	64	23	87
Historia	16	2	18
Total	115	31	146

Tabla 3.2: Cálculo de los totales por fila y por columna.

Después ubicamos los datos que nos interesan, en este caso, los estudiantes que aprueban matemáticas, es decir, la fila 2 y la columna 1.

Dicho esto, tenemos que $M_R = 87$ y $M_C = 115$ y $n = 146$. Por lo tanto la Ecuación 3.2 queda de la siguiente forma:

$$E = \frac{87 * 115}{146} = 68,52$$

De acuerdo con lo anterior la frecuencia esperada de que los estudiantes aprueben matemáticas es de 68.52. Este mismo procedimiento se repite para todas las filas y columnas, así se obtienen las frecuencias de cada elemento de la tabla, quedando como:

Frecuencia esperada	Aprobados	Reprobados
Español	32.29	8.70
Matemáticas	68.52	18.47
Historia	14.17	3.82

Tabla 3.3: Frecuencias esperadas por cada dato en la tabla.

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 ajustando un modelo simple de predicción en cada una de esas divisiones.

El algoritmo usa una herramienta extra de aprendizaje, llamada *poda*. Loh (2014) explica que el método de poda es una herramienta bastante útil, ya que se basa en el

concepto de eliminar al *eslabón más débil*.

Los valores de costo-complejidad (y *debilidad*) se pueden medir utilizando las métricas *Gini* y de *Entropía*. La definición de la primera se proporciona a continuación y, la de la segunda, se proporciona en la sección siguiente.

Métrica Gini: es la más usada en árboles de clasificación, Alvaredo (2011) explica que la métrica Gini es más sensible a las transferencias en el centro de las distribuciones de datos. Además, Timofeev (2004) menciona que Gini utiliza la siguiente formula de *impureza* para determinar cuál es el atributo de los datos óptimo para dividirlos:

$$i(t) = \sum_{k \times l} p(k|t)p(k|l) \quad (3.3)$$

3.1.3. Algoritmo 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. El árbol de decisión considera todas las posibles divisiones de los datos, con la finalidad de seleccionar aquella que genere la mayor ganancia de información posible.

Cintra y cols. (2013) señalan que los árboles de decisión C4.5 emplean la entropía y la ganancia de información como métricas para decidir cuáles son los atributos que mejor dividen al conjunto de datos.

Las definiciones matemáticas de la entropía y la ganancia de información, según Ahmad y cols. (2020), son las siguientes:

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

donde S es el subconjunto de datos a evaluar y p es la proporción de la clase.

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

donde S es el caso a evaluar, A es un atributo a evaluar, $|S_i|$ es el caso actual y $|S|$ 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 para estimar la relación entre variables. Dichos métodos son los más comunes en ML para la predicción de datos.

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

- Descriptivo: el método ayuda a analizar la *fuerza* que existe entre las variables independientes X y la variable dependiente y .
- Ajuste: el método es capaz de ajustarse para minimizar el error.

Existen dos principales modelos de regresión, la lineal y la 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 aplicar la regresión lineal en los problemas donde las variables son categóricas. 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:

$$y = \beta_0 + \beta_1 x \quad (3.6)$$

donde β_0 es la intercepción de la recta o *sesgo*, β_1 es la pendiente de la misma y la x es la variable independiente.

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

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

donde ϵ es el error (distancia) entre la recta calculada y las variables independientes x_i .

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

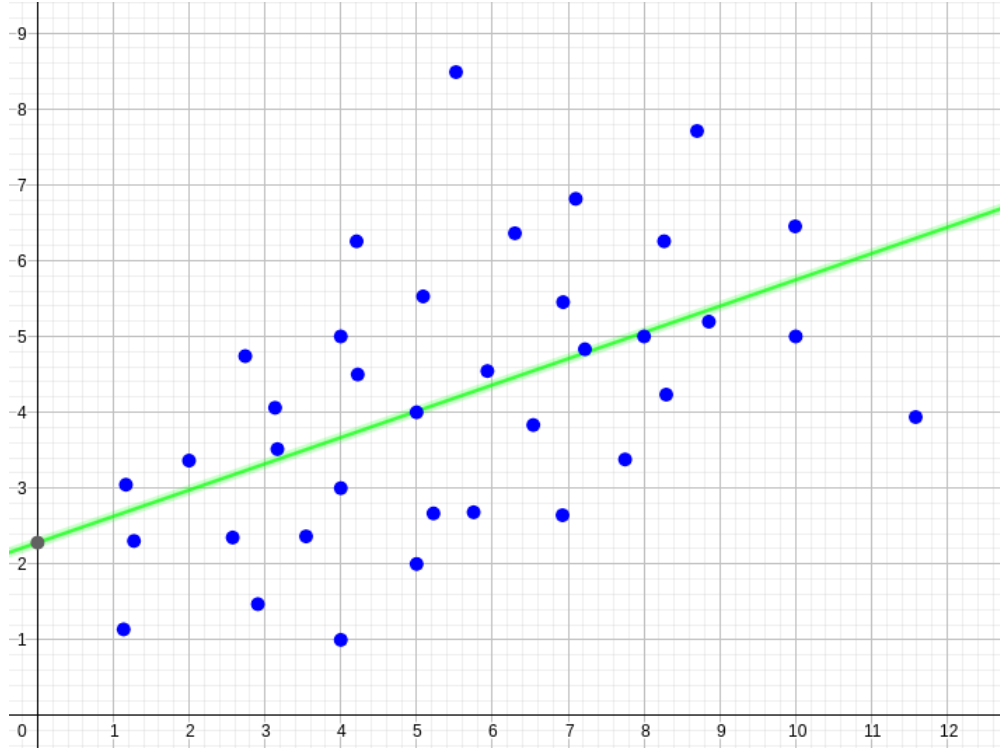


Figura 3.1: Ejemplo de una recta calculada con regresión lineal donde $y = 2.2816 + 0.3464x$.

3.2.2. Regresión logística

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

Harrell (2015) explica que el algoritmo consiste en generar la ecuación de la función sigmoide (Figura 3.2) que permita explicar la relación que existe entre las variables independientes y la variable dependiente (X y y).

$$y = \frac{1}{(1 + e^{-X})} \quad (3.8)$$

donde las X son todas las variables independientes representadas con la ecuación de una recta.

Para determinar la clasificación el modelo de regresión logística toma en cuenta la salida de la función sigmoide (Ecuación 3.8) y:

- Si la salida es ≤ 0.5 , entonces el algoritmo toma como resultado 0 (la clase negativa).
- Si la salida es > 0.5 , entonces el algoritmo toma como salida 1 (la clase positiva).

Lo anterior se puede apreciar en la Figura 3.2.

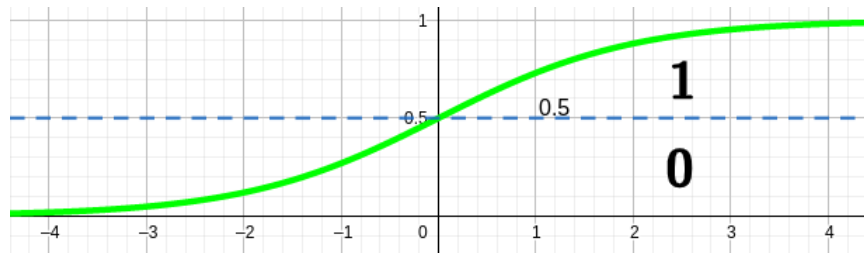


Figura 3.2: Función Sigmoide aplicada a la regresión lineal.

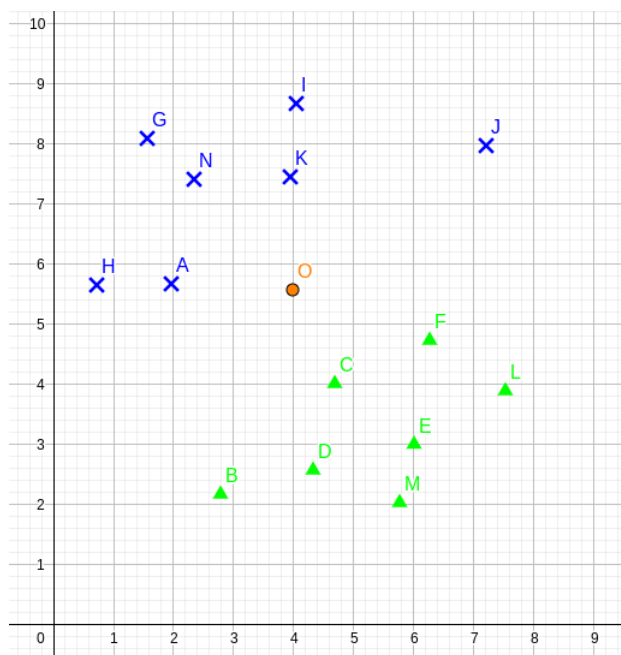
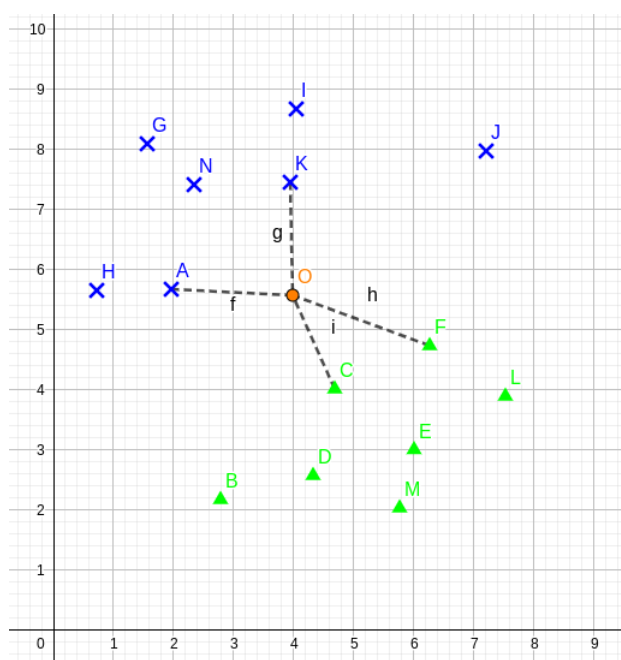
3.3. k -vecinos más cercanos (KNN)

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

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 (Ecuación 3.9) entre el *vecino* a clasificar y el resto de *vecinos* del conjunto de datos de entrenamiento.

Se seleccionan los k *vecinos* más cercanos, es decir, con menor distancia según la función de distancia que se emplee (euclidiana, coseno, etc.), ver Figuras 3.3 y 3.4.

Figura 3.3: Nuevo punto O a clasificar.Figura 3.4: Vecinos más cercanos al punto O (con menor distancia).

En la Figura 3.3 se muestran dos clases diferentes, la clase 1 se representa con cruces azules, mientras que la clase 2 está representada con triángulos verdes. También se muestra un nuevo punto a clasificar, representado con un círculo naranja.

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

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.9)$$

En la Figura 3.4 se están calculando las distancias entre el punto O (circulo naranja) y los puntos más cercanos a él. Se puede apreciar que se encontraron cuatro puntos, dos cruces azules y dos triángulos verdes.

Una vez que se obtienen los puntos con menor distancia, se seleccionan los k vecinos más cercanos (en este caso $k=3$) y la etiqueta que domine el conjunto (la etiqueta más frecuente) es la que decidirá la clasificación del punto actual.

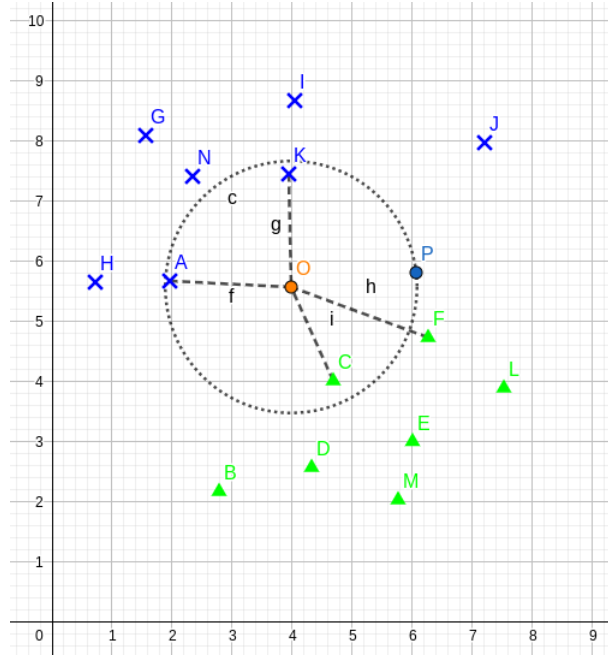


Figura 3.5: Vecinos más cercanos (con $k = 3$).

Como se puede ver en la Figura 3.5 hay dos puntos etiquetados como **cruz azul** y sólo uno etiquetado como **triángulo verde**, por lo tanto, el punto en cuestión se etiquetaría como una cruz azul.

3.4. Clustering (k -medias)

k -medias (k -Means en inglés) es un algoritmo de clasificación no supervisada que agrupa objetos (*clustering*) 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 llamados *centroides* (Figura 3.6), los cuales tendrán su respectivo subconjunto de puntos.

Después, para cada punto x_j en cada vecindario, se toma la distancia de éste a su respectivo centroide μ_i (o al más cercano), ver Figura 3.7. Por lo general, se utiliza la distancia euclidiana ($\|\bullet\|$) para obtener las distancias de cada punto al centro de su subconjunto:

$$Centroide(x_j) = \arg \min_{\mu_i} \|x_j - \mu_i\| \quad (3.10)$$

Se considera la menor distancia obtenida para que el punto actual se etiquete de acuerdo a su centroide más cercano (Figura 3.8).

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

1. Cuando el algoritmo ha llegado al número de iteraciones especificado al inicio de éste.
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 (Figura 3.9).

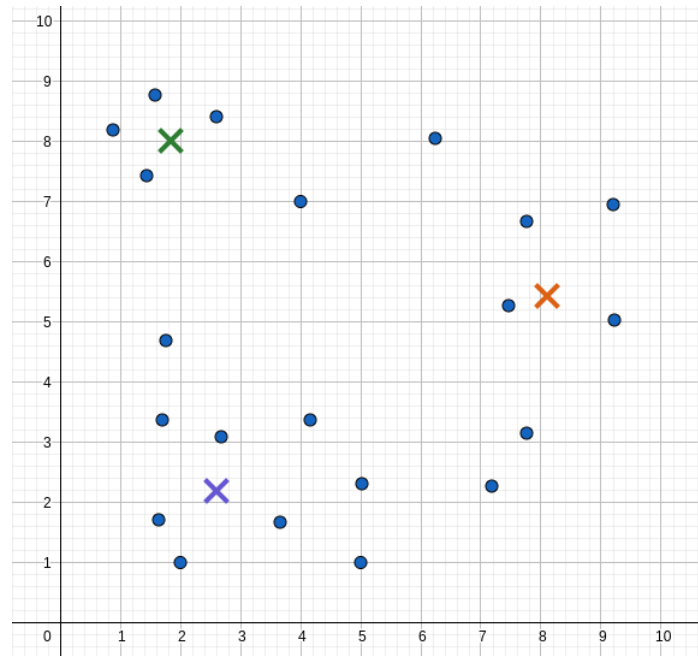


Figura 3.6: $k = 3$ centroides en un conjunto de datos.

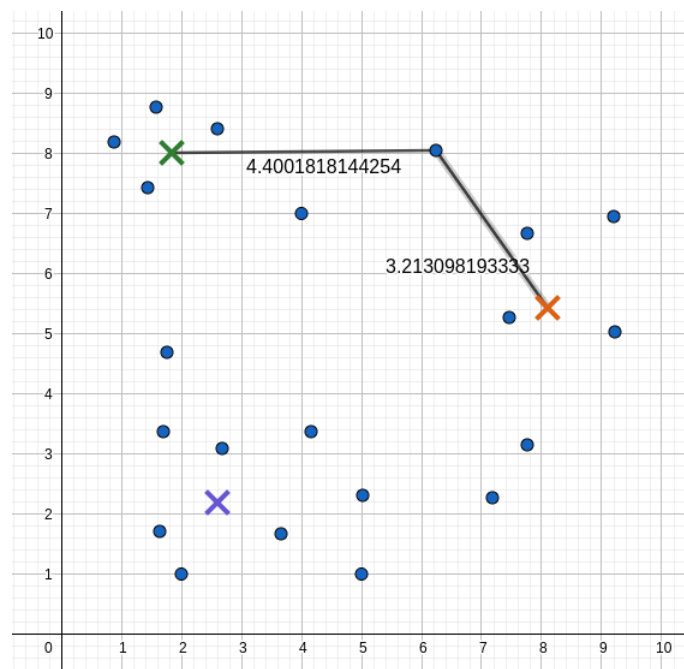


Figura 3.7: Buscando el centroide más cercano.

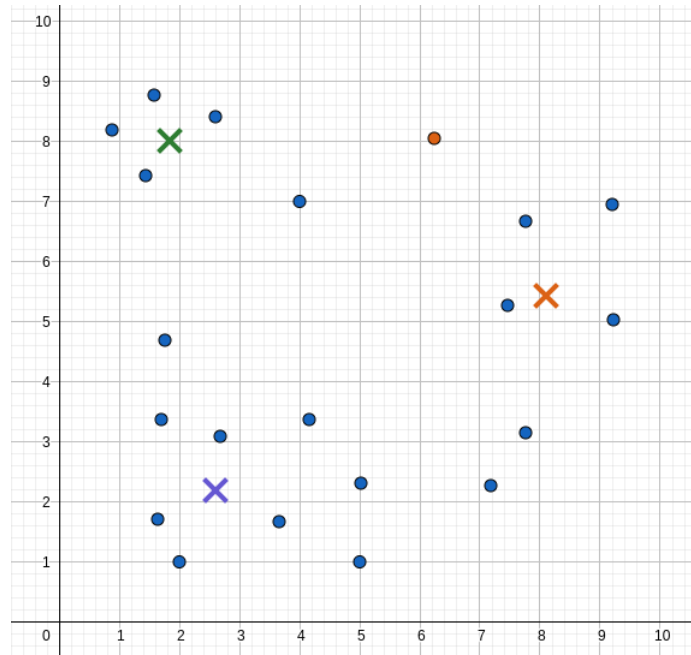


Figura 3.8: Etiquetado del punto actual de acuerdo a su centroide más cercano.

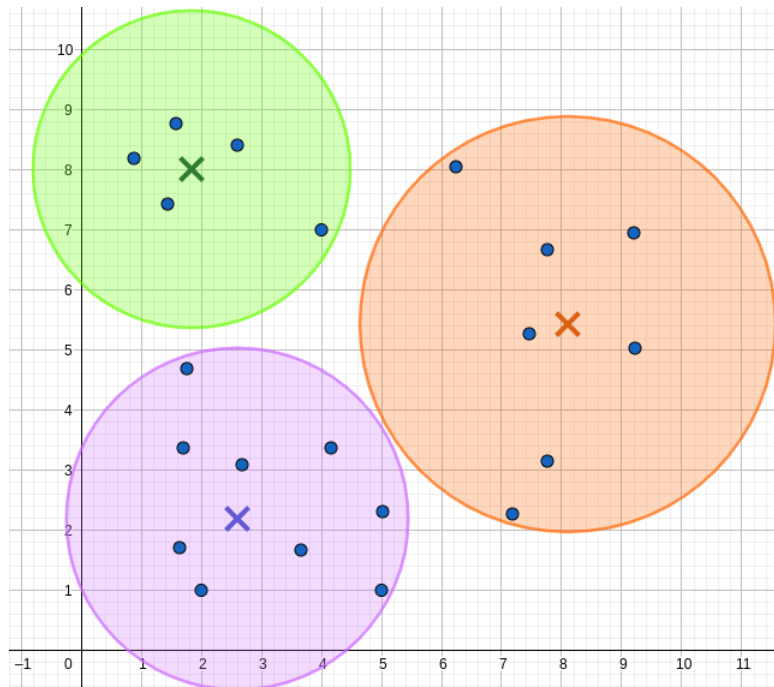


Figura 3.9: k grupos formados.

3.5. Dask

En muchos de los casos, se emplean herramientas del lenguaje de programación Python (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 pueden procesar?

Para ese tipo de casos, Python cuenta con una librería especial dedicada al procesamiento de grandes volúmenes de datos: *Dask* (<https://www.dask.org/>), esta simplifica las tareas de paralelización a la hora de hacer procedimientos de ML y DL.

La característica principal de Dask es que 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 en donde se tienen los comandos que dan el mismo resultado, es decir, devuelven un *dataframe* con los datos cargados desde un archivo en formato de valores separados por comas (.CSV).

Otra propiedad especial que tiene Dask, es que se pueden cargar varios archivos .CSV en una sola sentencia, siempre y cuando éstos compartan el mismo esquema (que tengan las 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) y sintaxis de Dask (abajo).

De igual forma, Dask tiene una sintaxis muy similar a la de NumPy, justo como se ve en la Figura 3.11. De esta forma, Dask puede realizar los mismos cálculos que 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 (izquierda) y sintaxis de Dask (derecha).

3.6. Sistema de archivos HDFS

El *Sistema de Archivos Distribuido de Hadoop* (*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) el HDFS tiene una gran tolerancia a los fallos ya que está 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 clusters en los cuales se crean subconjuntos de datos, lo que genera una arquitectura *maestro y trabajadores*.

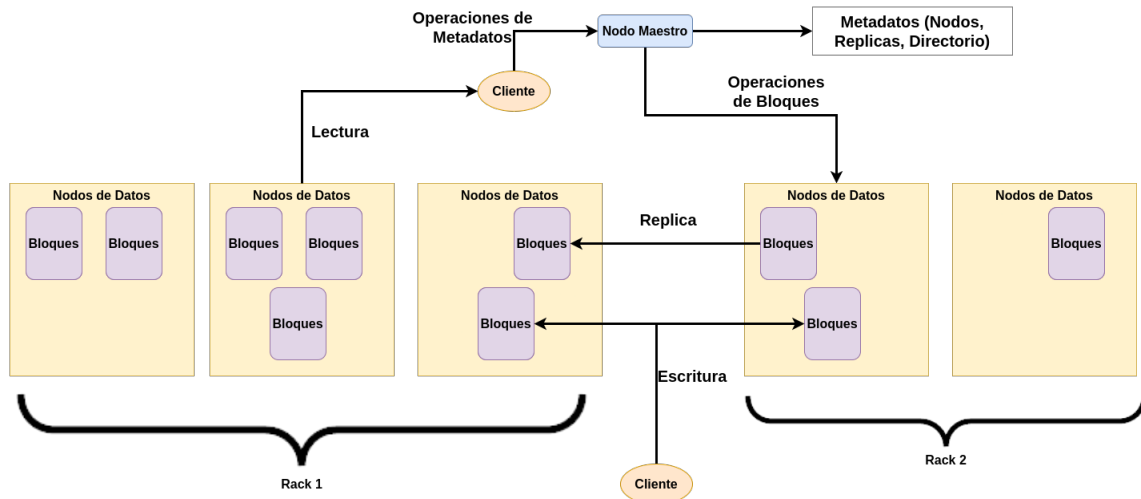


Figura 3.12: Diagrama de un HDFS.

Como se puede ver en la Figura 3.12, la arquitectura consiste de los siguientes elementos:

Nodo maestro: este nodo coordina el almacenamiento de todos los datos del sistema de archivos en un cluster. Además, también se encarga de almacenar todos sus metadatos.

Nodos de datos: son servidores de fragmentos de archivos (o de los archivos completos si son lo suficientemente pequeños), lo que significa que si uno de estos nodos falla, el archivo aún se encuentra disponible en cualquier momento, dado que cada fragmento se replica en varios de estos nodos (por omisión existen tres réplicas de cada fragmento, pero este número es configurable).

Data Chunks: son bloques de datos que representan un archivo. Cada bloque es replicado y añadido a un nodo de datos.

Una vez que se ha descrito la terminología básica de los modelos de ML, los algoritmos principales que se emplean para la creación de dichos modelos, así como los medios mediante los cuales se procesan y almacenan los datos sobre los que éstos trabajan, el capítulo siguiente detalla las métricas con las que se evalúa su desempeño.

Capítulo 4

Métricas para Evaluar Modelos

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

4.1. RMSE

El *error de la raíz cuadrada promedio* o *RMSE* por sus siglas en inglés (*Root Mean Square Error*) lo define Barnston (1992) de la siguiente manera:

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

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

El RMSE se puede interpretar como la desviación estándar de la varianza. Además de que ésta nos da valores dentro de la misma escala de los datos. Un RMSE bajo indica un mejor ajuste del modelo y un valor alto indica que el modelo requiere modificaciones.

4.2. MAE

El *error absoluto promedio* o *MAE* por sus siglas en inglés (*Mean Absolute Error*) es una de las métricas más usadas para evaluar el desempeño de varios modelos de ML. Chai y Draxler (2014) mencionan que el MAE le da el mismo peso a todos los errores y lo definen de la siguiente manera:

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad (4.2)$$

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

Como se puede ver en la Ecuación 4.2, el MAE mide qué tan cerca se encuentra la predicción con 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 del MAE es el *NMAE* o MAE normalizado. Jannach y cols. (2010) muestran que el NMAE consiste en normalizar los valores del MAE con respecto a los valores con los que se trabaja, esto es:

$$NMAE = \frac{MAE}{r_{max} - r_{min}} \quad (4.3)$$

donde r_{max} es el valor máximo del conjunto de datos y r_{min} es el valor mínimo de éste.

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.

Como se puede ver en la Figura 4.1 existen dos ejes, los valores de predicción y

VALORES PREDICCIÓN	Positivo	Verdaderos positivos	Falsos Positivos
	Negativo	Falsos Negativos	Verdaderos Negativos
		Positivo	Negativo
		VALORES REALES	

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

los valores reales. Dados estos ejes, la matriz de confusión se compone de:

Verdaderos positivos (TP – *True Positive*): Son los valores clasificados como positivos cuando los valores reales también son positivos.

Falsos positivos (FP – *False Positive*): Son los valores clasificados como positivos cuando los valores reales son negativos.

Falsos negativos (FN – *False Negative*): Son los valores clasificados como negativos cuando los valores reales son positivos.

Verdaderos negativos (TN – *True Negative*): Son los valores clasificados como negativos cuando los valores reales también son negativos.

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

4.4. Exactitud

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

$$Exactitud = \frac{TP + TN}{N} \quad (4.4)$$

donde N es el número 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 cercanos están los valores predichos con respecto a los valores reales de los datos.

4.5. Precisión

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

$$Precision = \frac{TP}{TP + FP} \quad (4.5)$$

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.

Para explicar la Figura 4.2 se usará una analogía de *tiros a una diana*, donde el color azul representa lo alejado que se encuentra al centro, mientras que la zona amarilla está más cerca del centro, es decir, de la zona más importante. Se muestra de izquierda a derecha, de arriba hacia abajo.

- Cuando un método es preciso y exacto, se representa como muchos tiros que caen cerca unos de otros y al centro (zona amarilla), es decir, los resultados del método son muy parecidos entre sí y a los valores a los que se quiere llegar.
- Cuando un método es preciso pero no exacto, son aquellos tiros que caen cerca unos de otros, pero dando resultados poco favorables al quedar en la zona azul, los valores son similares entre sí, pero no se acercan a los reales.
- Cuando un método no es preciso pero sí es exacto, se muestran tiros muy ale-

jados unos de otros, cerca de la zona amarillal, es decir, los datos no tienen sentido entre sí, pero se acercan a los que se quiere llegar.

- Cuando un método no es preciso ni exacto, básicamente son tiros al azar.

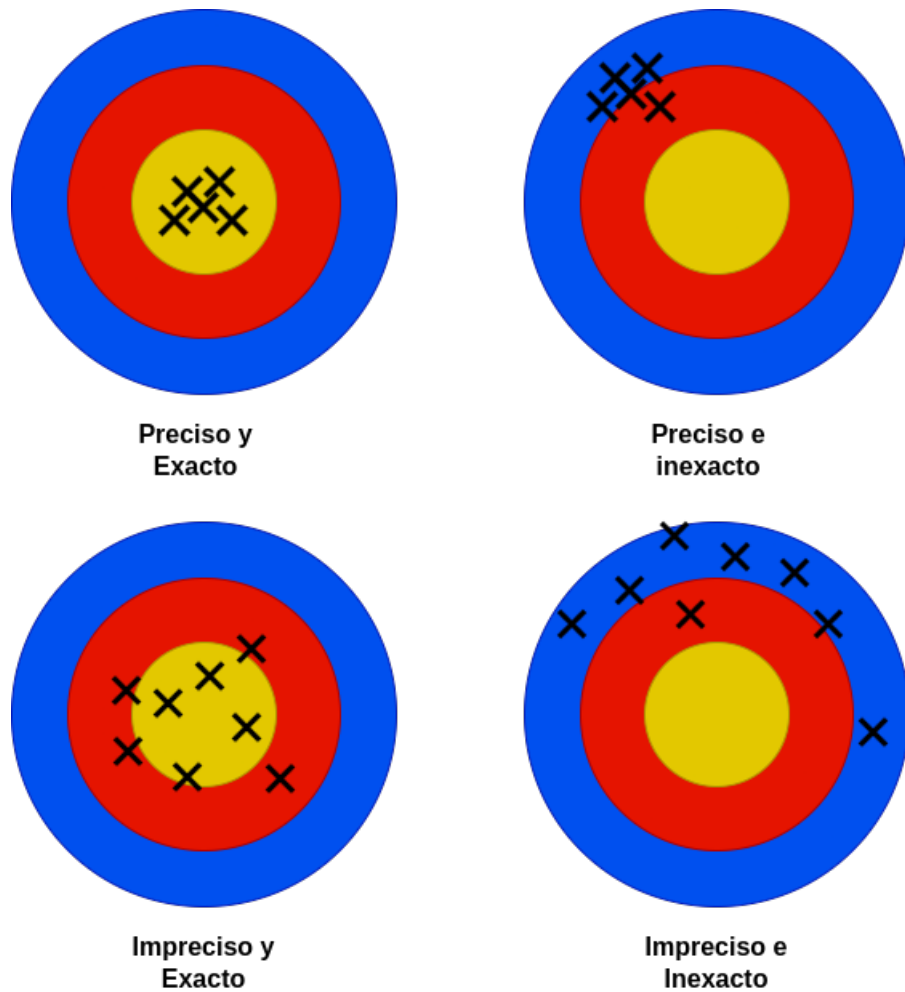


Figura 4.2: Representación gráfica de la exactitud vs. precisión.

4.6. Recall (Sensibilidad)

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:

$$Recall = \frac{TP}{TP + FN} \quad (4.6)$$

De acuerdo a la Ecuación 4.6 sabemos que si el recall obtiene un valor cercano a 1.0 (el 100%), indica que el modelo tiene un buen rendimiento; en caso de dar un valor cercano a 0.0, indica que el modelo necesita ajustes.

4.7. Métrica F_β

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

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

$$F_\beta = (1 + \beta^2) \frac{P \times R}{\beta^2 \times P + R} \quad (4.7)$$

donde P es el valor de la precisión, R es el valor del recall y β determina el balance entre ambas.

4.7.1. Métrica F_1

De acuerdo a Chicco y Jurman (2020) la métrica F_1 es la más común dentro de las métricas F_β .

La métrica F_1 es una media armónica entre la precisión y el recall, es decir, ambas métricas tienen el mismo porcentaje de importancia. Teniendo esto en consideración Huang y cols. (2015) definen a la métrica F_1 de la siguiente manera:

$$F_1 = 2 \frac{P \times R}{P + R} \quad (4.8)$$

Como ambas métricas tienen la misma importancia, la única forma de tener una

F_1 alta es que P y R tengan un valor alto.

4.7.2. Métricas $F_{0.5}$, F_1 y F_2

El problema del balance entre la precisión y el recall subyace en que no siempre se pueden presentar casos en que ambas sean altos debido a que, si el valor de una de ellas aumenta, la otra tiende a disminuir.

Para tratar este tipo de casos, el valor de β puede variar de acuerdo a las necesidades del modelo. Los valores más utilizados en la práctica son:

$\beta = 0.5$, **métrica $F_{0.5}$** : esta medición le da más importancia a la precisión.

$\beta = 1$, **métrica F_1** : representa el equilibrio entre precisión y recall.

$\beta = 2$, **métrica F_2** : el recall resulta más importante que la precisión.

Al término de la exposición en este trabajo de los antecedentes y la terminología necesaria para el desarrollo del manual de prácticas para apoyo de las asignaturas que emplean al ML como una herramienta, el paso siguiente consiste en detallar la estructura de cada una de estas prácticas al mismo tiempo que se enlistan los temas sobre los cuales se plantean. El capítulo siguiente se encarga de ambas cuestiones.

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 utilizar.
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.

En el caso de la práctica 5 y 6 (*Instalación y uso de Dask* e *Instalación y uso de HDFS* respectivamente) no se pueden aplicar los puntos 4d, 4e y 4f debido a que no se pueden construir modelos, ni hacer análisis de errores ni una implementación con datos. Solo explican el concepto de cada herramienta, su instalación y uso.

Así mismo, la Tabla 5.1 detalla las prácticas de laboratorio de cómputo contenidas en el manual desarrollado.

N°	Nombre	Datos	Métrica	Descripción
1	Clasificación usando árboles de decisión	Pasajeros del Titanic	Exactitud y/o Métrica F_β	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 casas-habitación en el este de California (década de 1990).
3	k -vecinos más cercanos	Pozos profundos del Lago de Cuitzeo	Precisión	Usar un conjunto de datos 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 de las transacciones de los clientes de un banco y así poder identificar a los diferentes clientes que existen (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 Dask 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 a paso como se realiza la instalación del HDFS y cuál es la utilidad de los comandos de éste.
7	Predicción del clima (Regresión Lineal)	RUOA de 2015 a 2021	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.

Tabla 5.1: Listado de prácticas de laboratorio de cómputo contenidas en el manual para el Módulo I.

Capítulo 6

Resultados y Discusión de la encuesta

De acuerdo con la experiencia de aplicar las prácticas desarrolladas en dos grupos de estudiantes y profesores de la UNAM, se ofrecieron cursos de acuerdo al diplomado, que se dividió en dos módulos Aprendizaje Automático y Aprendizaje profundo:

6.1. Prueba Piloto

MÓDULO I. Aprendizaje automático (ML). *Teoría y práctica para la mejora de la enseñanza del aprendizaje máquina (ML) aplicado a la Ciencia de Datos*. Prácticas descritas en la Tabla 5.1.

Al finalizar el primer curso de ML, se observó que de 40 asistentes el 50% presentó problemas de resolución de las prácticas, dicho porcentaje se muestra a continuación: Práctica I) 100%, Práctica II) 80%, Práctica III) 70%, Práctica IV) 50%, Práctica V) 50%, Práctica VI) 50% Práctica VII) 50%.

Como se muestra en la Figura 6.1 los resultados esperados (según la experiencia en cursos anteriores) frente a los resultados reales, como prácticas resueltas y entregadas fueron disminuyendo, de tal manera que las prácticas resueltas se fueron reduciendo, también muestra la eficiencia de la enseñanza de acuerdo a:

$$\%eficiencia = 100 * (prcticas\ resueltas - practicas\ esperadas) / practicas\ esperadas \quad (6.1)$$

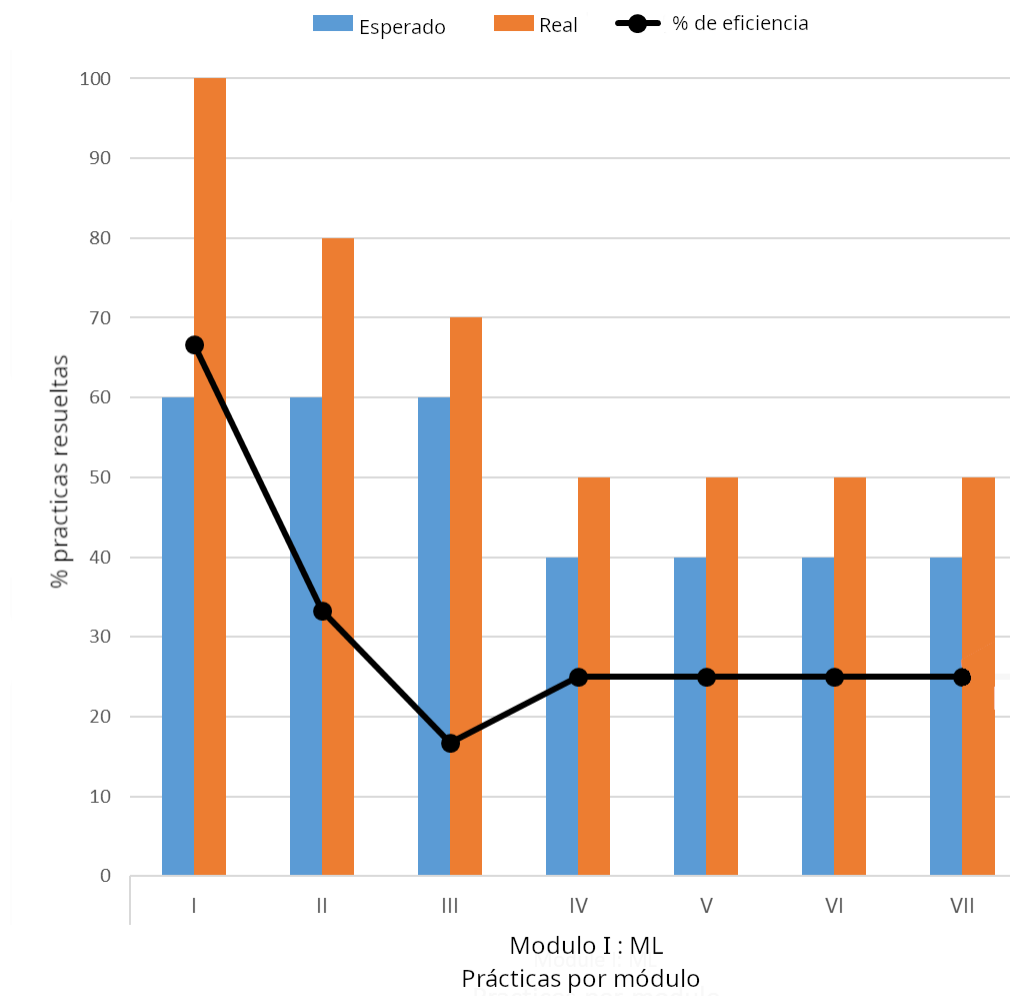


Figura 6.1: Resultados del Módulo I del Diplomado.

Además se observó que los estudiantes dejaron de trabajar en las prácticas más complejas. Las principales razones que se identificaron fueron el aumento de las tareas para el análisis de datos, además de tener que aplicar estadística y teoría matemática, utilizando un lenguaje de programación (Python).

En cursos similares a este, la enseñanza de ML se utiliza sólo como un caso de uso para abordar los temas pero no se realizan prácticas, que es la propuesta de mejora que se hace a través de este manual.

El manual de prácticas permite establecer un currículo más completo y amplio, en cuanto a los temas que se deben incluir en el ML, DL y Big Data y las herramientas informáticas asociadas con la Ciencia de Datos.

La propuesta referida en este informe aún está en desarrollo y, entre otras cuestiones, es necesario evaluar la eficiencia de la enseñanza de acuerdo con este enfoque práctico en Módulo II (*curso 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.

Apéndice

Los anexos consisten en:

- Código fuente en lenguaje Python (como libretas de JupyterLab) de las prácticas desarrolladas.
- El listado de enlaces a las fuentes originales de las cuales se tomaron los conjuntos de datos para las prácticas.
- La primera página del artículo publicado en las memorias del Congreso Internacional *IntelliSys 2022*.

.1. Prácticas

Como se mencionó al inicio de este documento, las practicas se encuentran en un repositorio publico de github: <https://github.com/MichellMonroy/Practicas-ML>

.2. Datos

Los conjuntos de datos que se usaron en las prácticas mostradas anteriormente, se pueden descargar de los enlaces siguientes:

- 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://acortar.link/eIeGDT>
- Online Retail K-means & Hierarchical Clustering.
<https://www.kaggle.com/hellbuoy/online-retail-customer-clustering>
- RUOA de 2015 a la actualidad.
<https://ruoa.unam.mx/index.php?page=estaciones&id=9>

- Información de datos públicos.

.3. Artículo



Al terminar el Módulo I del diplomado mencionado anteriormente, se realizó un artículo donde Tinoco-Martínez y cols. (2022) recabaron la información que se obtuvo de este. En seguida se muestra la primera página de este para que se aprecie el título, los autores y autora.

Puede revisar el artículo completo en el siguiente doi:

https://doi.org/10.1007/978-3-031-16075-2_1



How to Improve the Teaching of Computational Machine Learning Applied to Large-Scale Data Science: The Case of Public Universities in Mexico

Sergio Rogelio Tinoco-Martínez¹, Heberto Ferreira-Medina^{2,3}() ,
José Luis Cendejas-Valdez⁴() , Froylan Hernández-Rendón¹,
Mariana Michell Flores-Monroy¹, and Bruce Hiram Ginori-Rodríguez¹

¹ Escuela Nacional de Estudios Superiores Unidad Morelia, UNAM Campus Morelia,
58190 Morelia, Michoacán, Mexico

{stinoco, fherandez}@enesmorelia.unam.mx

² Instituto de Investigaciones en Ecosistemas y Sustentabilidad, UNAM Campus Morelia,
58190 Morelia, Michoacán, Mexico

hferreir@iies.unam.mx

³ Tecnológico Nacional de México, Campus Morelia, DSC, Morelia, Michoacán, Mexico

⁴ Departamento de TI, Universidad Tecnológica de Morelia. Cuerpo académico
TRATEC-PRODEP. Morelia, 58200 Morelia, Michoacán, México

luis.cendejas@ut-morelia.edu.mx

Abstract. Teaching along with training on Machine Learning (ML) and Big Data in Mexican universities has become a necessity that requires the application of courses, handbooks, and practices that allow improvement in the learning of Data Science (DS) and Artificial Intelligence (AI) subjects. This work shows how the academy and the Information Technology industry use tools to analyze large volumes of data to support decision-making, which is hard to treat and interpret directly. A solution to some large-scale national problems is the inclusion of these subjects in related courses within specialization areas that universities offer. The methodology in this work is as follows: 1) Selection of topics and tools for ML and Big Data teaching, 2) Design of practices with application to real data problems, and 3) Implementation and/or application of these practices in a specialization diploma. Results of a survey applied to academic staff and students are shown. The survey respondents have already taken related courses along with those specific topics that the proposed courses and practices will seek to strengthen, developing needed skills for solving problems where ML/DL and Big Data are an outstanding alternative of solution.

Keywords: Machine learning · Deep learning · Big data · Data science · Teaching skills

1 Introduction

The use of tools that allow the analysis of large volumes of data has allowed exact sciences to play an important role for decision-making in organizations [1]. In the Bachelor

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 Tecnologías 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., Treffer, 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.
- Tinoco-Martínez, S. R., Ferreira-Medina, H., Cendejas-Valdez, J. L., Hernández-Rendón, F., Flores-Monroy, M. M., y Ginori-Rodríguez, B. H. (2022). How to improve the teaching of computational machine learning applied to large-scale data science: The case of public universities in mexico. En *Intelligent systems and applications: Proceedings of the 2022 intelligent systems conference (intellisys) volume 3* (pp. 1–15).
- 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). Non-linear 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.
- Zou, J., Han, Y., y So, S.-S. (2009). Overview of artificial neural networks. *Artificial neural networks: methods and applications*, 14–22.