Inteligencia Artificial Avanzada para la Ciencia de Datos 1

Adrián Galván Díaz
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de Estudios Superiores de Monterrey
Querétaro, México
A01704076@tec.mx

***Abstract* — Este proyecto se enfoca en la implementación manual de un algoritmo de regresión logística utilizando gradiente descendiente, sin la asistencia de marcos de trabajo o bibliotecas externas. El objetivo principal fue desarrollar un modelo capaz de predecir con precisión si un hongo es comestible o venenoso, basándose en un conjunto de datos categóricos que describe 23 especies de hongos de las familias Agaricus y Lepiota. A través de un riguroso Análisis Exploratorio de Datos (EDA), se preparó y transformó el set de datos, lo que permitió la creación y evaluación del modelo de clasificación. Los resultados obtenidos demostraron un alto porcentaje de precisión, confirmando la eficacia del algoritmo en la tarea de clasificación. Este proyecto no solo ofrece una contribución significativa al entendimiento y aplicación de algoritmos de Machine Learning sin frameworks, sino que también establece una base sólida para la escalabilidad de técnicas de clasificación en conjuntos de datos categóricos complejos.**

Keywords* — *Regresión Logística, Gradiente Descendiente, Análisis Exploratorio de Datos (EDA), Machine Learning (ML), Inteligencia Artificial (IA), Split Test, Matriz de Confusión, Épocas/Epochs, Learning Rate, Parámetros, Hiper Parámetros, Python, Instancias, Clase(s), Features, Entropía Cruzada

I. Introducción

La inteligencia artificial (IA) ha evolucionado significativamente desde sus inicios a mediados del siglo XX [1]. Originalmente, la IA se centraba en la creación de sistemas que pudieran realizar tareas que normalmente requerirían inteligencia humana, como la resolución de problemas y el procesamiento del lenguaje natural. Con el tiempo, la IA se ha diversificado en varios subcampos, siendo el aprendizaje automático o Machine Learning (ML) uno de los más prominentes. ML se refiere a la capacidad de las máquinas para aprender de datos y mejorar su rendimiento sin ser programadas explícitamente para cada tarea específica [2].

El desarrollo del aprendizaje automático comenzó con el reconocimiento de patrones y se ha expandido para incluir una amplia gama de técnicas que permiten a los sistemas aprender y tomar decisiones basadas en datos. Hoy en día, ML se clasifica principalmente en tres categorías: aprendizaje supervisado, no supervisado y por refuerzo.

Aprendizaje supervisado: Involucra entrenar un modelo en un conjunto de datos etiquetado, donde la respuesta correcta es conocida. Es ideal para tareas de clasificación y regresión. [3]

Aprendizaje no supervisado: Se utiliza cuando los datos no están etiquetados y el objetivo es identificar estructuras ocultas o patrones en los datos, como en la agrupación o reducción de dimensionalidad. No necesita de intervención humana en el proceso de aprendizaje. [3]

Aprendizaje por refuerzo: Implica entrenar un modelo para tomar decisiones secuenciales a través de un sistema de recompensas y castigos, comúnmente utilizado en áreas como el control robótico y los videojuegos. [3]

Hoy en día, la IA y el ML tienen un impacto profundo en diversos sectores, desde la medicina hasta la industria, mejorando procesos, optimizando recursos y generando soluciones innovadoras.

A. Contexto general

En el contexto de este proyecto, los algoritmos de machine learning, en particular los de clasificación, son fundamentales para resolver problemas de decisión binaria. Este proyecto se enfoca en el uso de un algoritmo de regresión logística, una técnica de aprendizaje supervisado, para predecir si un hongo es comestible o venenoso basándose en sus características observables.

La regresión logística es uno de los métodos más comunes para problemas de clasificación binaria, mientras que la regresión lineal se utiliza generalmente para problemas de predicción continua. La capacidad de estos modelos para hacer predicciones precisas y basadas en datos los convierte en herramientas esenciales en la toma de decisiones en diversas áreas.

Aplicar machine learning al análisis de hongos, una rama de la biología, permite no solo avanzar en la identificación precisa de especies, sino también abrir la puerta a aplicaciones en otras áreas biológicas, como la clasificación de plantas, la detección de enfermedades en cultivos, y el análisis de patrones de comportamiento animal.

B. Base de Datos

El conjunto de datos utilizado en este proyecto proviene del UC Irvine Machine Learning Repository [4], un recurso ampliamente reconocido y de acceso público. Esta base de datos contiene descripciones de muestras hipotéticas de 23 especies de hongos pertenecientes a las familias Agaricus y Lepiota. En total, la base de datos cuenta con 8124 instancias y 22 atributos, todos nominalmente valorados. A continuación, se listan las variables:

Tabla 1. Tabla de Variables

Nombre de la Variable	Descripción	Valores Posibles
cap-shape	Forma del sombrero	bell (b), conical (c), convex (x), flat (f), knobbed (k), sunken (s)
cap-surface	Superficie del sombrero	fibrous (f), grooves (g), scaly (y), smooth (s)
cap-color	Color del sombrero	brown (n), buff (b), cinnamon (c), gray (g), green (r), pink (p), purple (u), red (e), white (w), yellow (y)
bruises	Presencia de moretones	bruises (t), no (f)
odor	Olor	almond (a), anise (l), creosote (c), fishy (y), foul (f), musty (m), none (n), pungent (p), spicy (s)
gill-attachment	Adherencia de las branquias	attached (a), descending (d), free (f), notched (n)
gill-spacing	Espaciado de las branquias	close (c), crowded (w), distant (d)
gill-size	Tamaño de las branquias	broad (b), narrow (n)
gill-color	Color de las branquias	black (k), brown (n), buff (b), chocolate (h), gray (g), green (r), orange (o), pink (p), purple (u), red (e), white (w), yellow (y)
stalk-shape	Forma del tallo	enlarging (e), tapering (t)
stalk-root	Raíz del tallo	bulbous (b), club (c), cup (u), equal (e), rhizomorphs (z), rooted (r), missing (?)
stalk-surface-above-ring	Superficie del tallo sobre el anillo	fibrous (f), scaly (y), silky (k), smooth (s)
stalk-surface-below-ring	Superficie del tallo bajo el anillo	fibrous (f), scaly (y), silky (k), smooth (s)
stalk-color-above-ring	Color del tallo sobre el anillo	brown (n), buff (b), cinnamon (c), gray (g), orange (o), pink (p), red (e), white (w), yellow (y)
stalk-color-below-ring	Color del tallo bajo el anillo	brown (n), buff (b), cinnamon (c), gray (g), orange (o), pink (p), red (e), white (w), yellow (y)
veil-type	Tipo de velo	partial (p), universal (u)
veil-color	Color del velo	brown (n), orange (o), white (w), yellow (y)
ring-number	Número de anillos	none (n), one (o), two (t)
ring-type	Tipo de anillo	cobwebby (c), evanescent (e), flaring (f), large (l), none (n), pendant (p), sheathing (s), zone (z)
spore-print-color	Color de la impresión de las esporas	black (k), brown (n), buff (b), chocolate (h), green (r), orange (o), purple (u), white (w), yellow (y)
population	Población	abundant (a), clustered (c), numerous (n), scattered (s), several (v), solitary (y)
habitat	Hábitat	grasses (g), leaves (l), meadows (m), paths (p), urban (u), waste (w), woods (d)

II. Análisis Exploratorio de Datos

El objetivo principal del Análisis Exploratorio de Datos (EDA) es conocer y comprender la estructura del conjunto de datos utilizado en este proyecto. Este proceso incluye la identificación de valores nulos, la visualización de la

distribución de las variables mediante gráficas, y la utilización de técnicas estadísticas como una matriz de correlación para tomar decisiones informadas sobre las variables más relevantes para el modelo. La finalidad del EDA es preparar los datos de manera adecuada para la construcción y evaluación del modelo de Machine Learning.

A. *Transformación de los Datos*

Para preparar los datos para el modelo de Machine Learning, se realizó una transformación de las variables categóricas utilizando One Hot Encoding [5]. Dado que las variables en el dataset no tenían una composición vectorial, sino que eran meramente clasificatorias (por ejemplo, colores, formas), el One Hot Encoding fue la técnica más adecuada para este caso. De esta forma se transformaron las variables donde cada una representa la presencia o ausencia de valor. Por ejemplo: bruises (1. bruises_t, 2. bruises_f).

B. *Revisión de los Datos*

Durante la revisión de los datos, se identificó la presencia de valores nulos en la variable stalk_root, que representa la forma de la raíz del tallo. Esta variable contenía 2480 valores nulos, lo que representa una proporción significativa del total de 8124 instancias en el dataset. Los valores nulos estaban representados por un signo de interrogación (?). A continuación, se graficó la distribución de esta variable para entender mejor su comportamiento en el dataset..

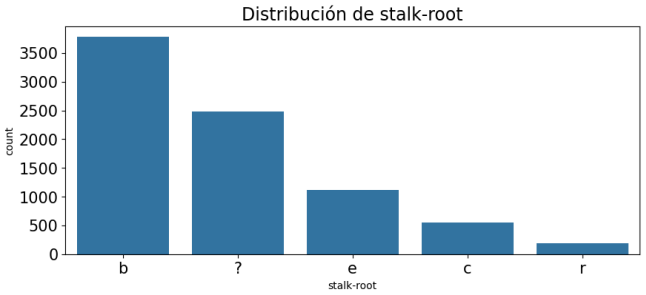

Fig. 1. Gráfico de barras para “stalk_root”

Como se observa en la Figura 1, una gran porción de los datos cuenta con un stalk_root indefinido. Inicialmente, se consideró la posibilidad de eliminar estas instancias, pero se optó por realizar un análisis más profundo a través de una matriz de correlación.

La matriz de correlación reveló una correlación positiva significativa de 0.784 entre los valores nulos en stalk_root y el color de las branquias negras (gill_color_b). Esta correlación sugiere que los valores nulos en stalk_root no son completamente aleatorios, sino que podrían estar asociados a características importantes de los hongos, posiblemente relacionados con morfologías específicas que afectan tanto la raíz como el color de las branquias. Por lo tanto, se decidió conservar estos valores nulos en el dataset, ya que podrían contener información útil para la predicción.

Como última observación, se encontró una correlación negativa fuerte de -0.78 entre la ausencia de olor (odor_n) y la comestibilidad de los hongos, indicando que los hongos sin olor tienden a ser comestibles. Esta información fue considerada valiosa y se mantuvo para su inclusión en el modelo.

III. MODELO

A. *Justificación*

a) *Regresión*

La regresión logística es un tipo de modelo de clasificación utilizado para predecir la probabilidad de que una observación pertenezca a una de dos categorías posibles [6]. A diferencia de la regresión lineal, que es adecuada para problemas de predicción continua, la regresión logística es ideal para situaciones donde la variable de resultado es binaria, como en este caso, donde queremos predecir si un hongo es comestible o venenoso.

La función central en la regresión logística es la función sigmoide, que mapea cualquier valor real a un rango entre 0 y 1, permitiendo interpretar el resultado como una probabilidad. La fórmula de la función sigmoide se muestra en la Figura 2 [6].

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fig. 2. Fórmula de una sigmoide

El uso de regresión lineal en este contexto no sería adecuado porque la regresión lineal asume una relación lineal entre las variables independientes y la variable dependiente. Sin embargo, en un problema de clasificación binaria, esta suposición no se sostiene, ya que el rango de salida de la regresión lineal no se limita a 0 y 1, como lo hace la función de una sigmoide (Figura 3) [6].

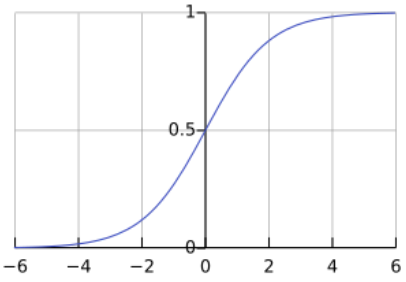

Fig. 3. Función representativa de una sigmoide

b) *Gradiente Descendiente*

El gradiente descendiente es un algoritmo de optimización utilizado para minimizar la función de pérdida, ajustando los parámetros del modelo (en este caso, los pesos **w** y el sesgo **b**). En cada iteración, el algoritmo calcula la pendiente de la función de pérdida con respecto a los parámetros y ajusta estos parámetros para reducir el error. El objetivo de este algoritmo es reducir la pendiente de la función de pérdida y acercarse a 0 de esta manera reduciendo el “costo” (Figura 4) [7].

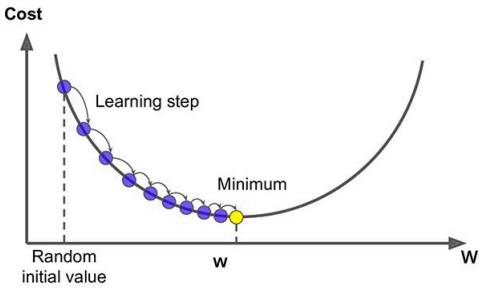

Fig. 4. Representación visual del funcionamiento del gradiente descendiente.

c) *Entropía Cruzada*

La entropía cruzada es una medida de la diferencia entre dos distribuciones de probabilidad. En el contexto de la regresión logística, se utiliza como función de pérdida para evaluar qué tan bien el modelo está prediciendo las probabilidades correctas, de esta manera la función de gradiente descendiente nos ayuda a disminuir el error de manera iterativa (Figura 6) [10]. La fórmula de la entropía cruzada para un modelo de clasificación binaria se puede ver en la Figura 5 [8, 9].

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i) \cdot \log(1-p(y_i))$$

Fig. 5. Fórmula de la entropía cruzada

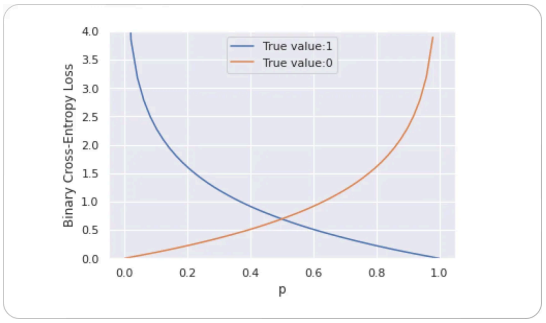

Fig. 6. Representación visual de la entropía cruzada.

B. *Preparación de los Datos*

En la preparación de los datos, es crucial separar el conjunto de datos en subconjuntos de entrenamiento y prueba. Esto es debido a que cuando se lleva a cabo un entrenamiento con el mismo conjunto de datos de prueba no sabrás cómo funcionará el algoritmo con datos con los que no se entrenó el modelo, es decir, los datos reales. El proceso de separar los datos, conocido como split test, implica dividir los datos en dos partes [11]:

1. **x_train, y_train**: Usados para entrenar el modelo.
2. **x_test, y_test**: Usados para evaluar el rendimiento del modelo.

C. *Descripción del modelo*

En el código se plantearon las funciones y métodos necesarios para llevar a cabo esta implementación. A continuación, describimos las funciones y parámetros utilizados en el modelo, organizándolos en una tabla para mayor claridad (Tabla 2).

TABLA 2. TABLA DE FUNCIONES

Nombre	**Función**	**Descripción**
sigmoid(z)	Mapea cualquier valor real a un rango entre 0 y 1 usando la función sigmoide.	Permite que el modelo prediga probabilidades entre 0 y1.
GD (x, y, w, b)	Implementa el gradiente descendiente para actualizar los pesos y el sesgo.	Optimiza los parámetros del modelo minimizando la función de pérdida a través de iteraciones.
loss (x, y, w, b)	Calcula la pérdida utilizando entropía cruzada. Los errores son guardados en un arreglo para posteriormente graficarlos.	Proporciona una medida del error del modelo en cada época, que es minimizada durante el entrenamiento.
predict(x, w, b)	Predice la clase de nuevas instancias basándose en los parámetros entrenados.	Utiliza la salida de la función sigmoide para clasificar las instancias como 0 (comestible) o 1 (venenoso).

Hiper-Parámetros:

Pesos (w): Inicializados en 0 y actualizados durante el entrenamiento.

Sesgo (b): Inicializado en 0 y actualizado junto con los pesos.

Parámetros:

Learning Rate: Tasa de aprendizaje, que controla qué tan grandes son los pasos del gradiente descendiente.

Epochs: Número de iteraciones para ajustar los pesos del modelo. Una época equivale a una vuelta completa a la base de datos.

D. Resultados

El modelo fue entrenado en dos configuraciones diferentes, utilizando 3000 y 6000 epochs. Los resultados muestran que con un mayor número de epochs, el modelo logra una mejor precisión en el conjunto de prueba.

Precisión en el conjunto de prueba (3000 epochs): 97.85%

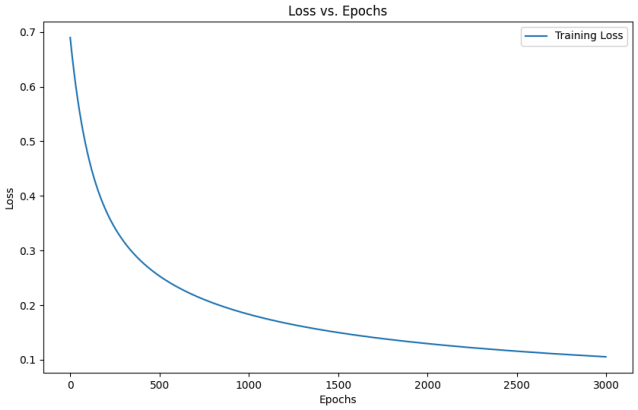

Fig. 7. Gráfica de pérdida a lo largo del tiempo 3000 epochs

Precisión en el conjunto de prueba (6000 epochs): 98.34%

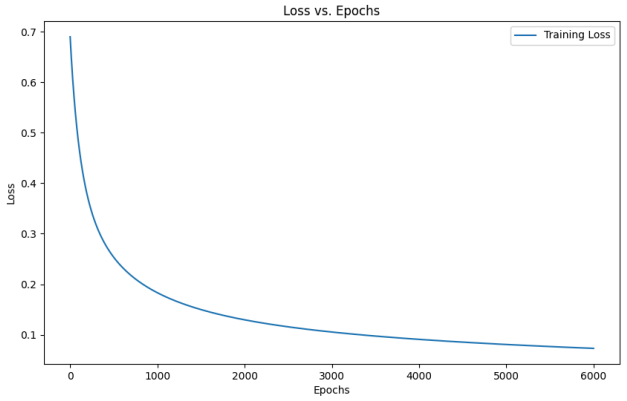

Fig. 8. Gráfica de pérdida a lo largo del tiempo 6000 epochs

Para la evaluación del modelo se decidió graficar una matriz de confusión para ambas pruebas de entrenamiento [12].

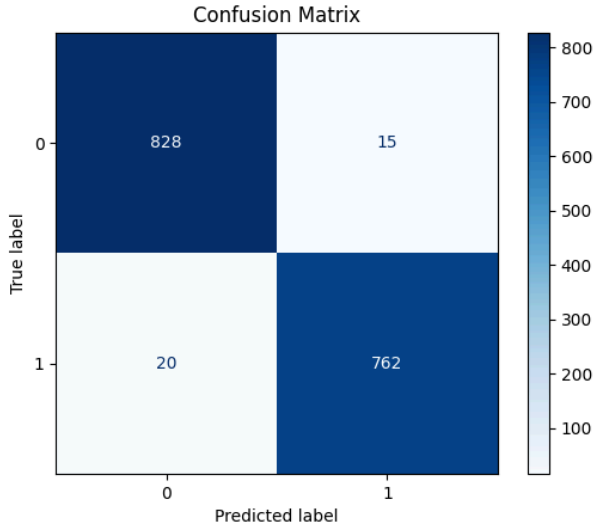

Fig. 9. Gráfica de matriz de confusión 3000 epochs

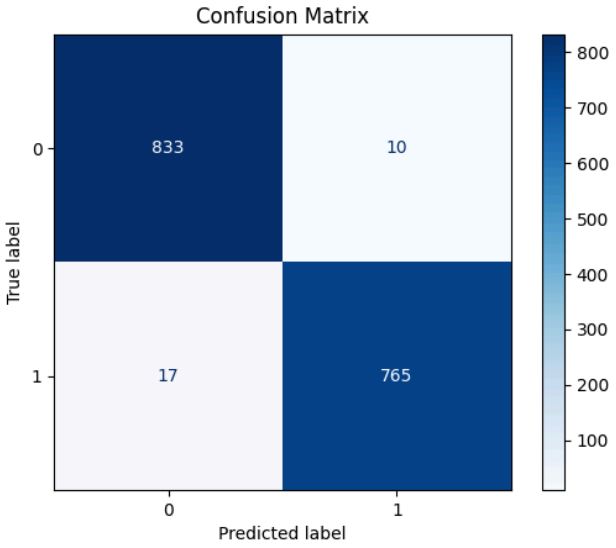

Fig. 10. Gráfica de matriz de confusión 6000 epochs

1. **3000 Epochs**:
   - TP (True Positives): 828
   - TN (True Negatives): 762
   - FP (False Positives): 15
   - FN (False Negatives): 20
2. **6000 Epochs**:
   - TP (True Positives): 833
   - TN (True Negatives): 765
   - FP (False Positives): 10
   - FN (False Negatives): 17

El aumento en las epochs resulta en una ligera reducción de errores, como lo demuestran las matrices de confusión, con menos falsas predicciones cuando se utilizan 6000 epochs. Sin embargo, la mejora es mínima, lo que sugiere que el modelo podría estar acercándose a su límite de rendimiento con este conjunto de datos. De igual manera, el

modelo tiene un muy buen desempeño con ambos conjuntos de epochs, con más de 97% de precisión en ambos casos.

References

[1] Evolución de la Inteligencia Artificial. (s. f.). Iberdrola. https://www.iberdrola.com/innovacion/evolucion-inteligencia-artificial

[2] ¿Qué es machine learning? | Definición, tipos y ejemplos | SAP. (s. f.). SAP. https://www.sap.com/latinamerica/products/artificial-intelligence/what-is-machine-learning.html#:~:text=El%20machine%20learning%20es%20un,experiencia%20sin%20ser%20programadas%20expl%C3%ADcitamente.

[3] Richardson, D. (2024, 8 mayo). La IA y el ML y su importancia para las empresas. Blog de Red Hat. https://www.redhat.com/es/blog/what-aiml-and-why-does-it-matter-your-business#:~:text=La%20inteligencia%20artificial%2C%20el%20aprendizaje,en%20las%20funciones%20del%20sistema.

[4] UCI Machine Learning Repository. (s. f.). https://archive.ics.uci.edu/dataset/73/mushroom

[5] Novack, G. (2024, 30 abril). Building a One Hot Encoding Layer with TensorFlow - Towards Data Science. Medium. https://towardsdatascience.com/building-a-one-hot-encoding-layer-with-tensorflow-f907d686bf39

[6] ¿Qué es la regresión logística? - Explicación del modelo de regresión logística - AWS. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/logistic-regression/

[7] Andrés, D., & Andrés, D. (2023, 18 enero). Gradiente descendente - ML Pills. ML Pills - Machine Learning Pills. https://mlpills.dev/machine-learning-es/gradiente-descendente/

[8] The cross-entropy error function in neural networks. (s. f.). Data Science Stack Exchange. https://datascience.stackexchange.com/questions/9302/the-cross-entropy-error-function-in-neural-networks

[9] Binary Cross Entropy: Where to use log loss in model Monitoring. (2023, 9 junio). Arize AI. https://arize.com/blog-course/binary-cross-entropy-log-loss/

[10] Shah, D. (2024, 2 julio). Cross entropy loss: intro, applications, code. V7. https://www.v7labs.com/blog/cross-entropy-loss-guide

[11] How To Choose The Right Test Options When Evaluating Machine Learning Algorithms. (s. f.). Machine Learning Mastery. https://machinelearningmastery.com/how-to-choose-the-right-test-options-when-evaluating-machine-learning-algorithms/

[12] Markham, K. (2024, 4 junio). Simple guide to confusion matrix terminology. Data School. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/