

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro



TC3006C. Inteligencia artificial avanzada para la ciencia de datos I

Grupo 101

Momento de Retroalimentación :

“Módulo 2 Implementación de una técnica de aprendizaje máquina sin el uso de un framework. (Portafolio Implementación)”

Evidencia presentada por:

Emiliano Mendoza Nieto

A01706083

Profesor :

Benjamín Valdés Aguirre

Fecha de entrega :

Domingo 10 de Septiembre de 2023

I. Introducción

En este entregable se ha elegido implementar una regresión logística usando el método de gradiente descendente estocástico (SGD).

Se han definido funciones auxiliares como **sigmoid** para el **mapeo de valores a una escala entre 0 y 1** y **cost** para **calcular el costo de las predicciones del modelo**.

También se ha implementado la función **gradient_descent** que utiliza el método de gradiente descendente para ajustar iterativamente los parámetros del modelo y minimizar el costo.

II. Descripción Base de Datos:

Clasifique las plantas de iris en tres especies en este conjunto de datos clásico.

Acerca del conjunto de datos

El conjunto de datos Iris se utilizó en R.A. El artículo clásico de Fisher de 1936, El uso de medidas múltiples en problemas taxonómicos, también se puede encontrar en el Repositorio de aprendizaje automático de la UCI.

Incluye tres especies de iris con 50 muestras cada una, así como algunas propiedades de cada flor. Una especie de flor es linealmente separable de las otras dos, pero las otras dos no son linealmente separables entre sí.

Las columnas de este conjunto de datos son:

- Id
- SepalLengthCm
- SepalWidthCm
- PetalLengthCm
- PetalWidthCm
- Species

III. Implementación

Se implementó una función **manual_train_test_split** para dividir los datos en conjuntos de entrenamiento y prueba.

Se entrenó el modelo usando los datos de entrenamiento, y luego se probaron las predicciones en ambos conjuntos.

Se registró la precisión, recall, F1-score y accuracy del modelo usando el conjunto de prueba.

Se proporciona una visualización del costo del modelo a lo largo de las épocas para ambos conjuntos.

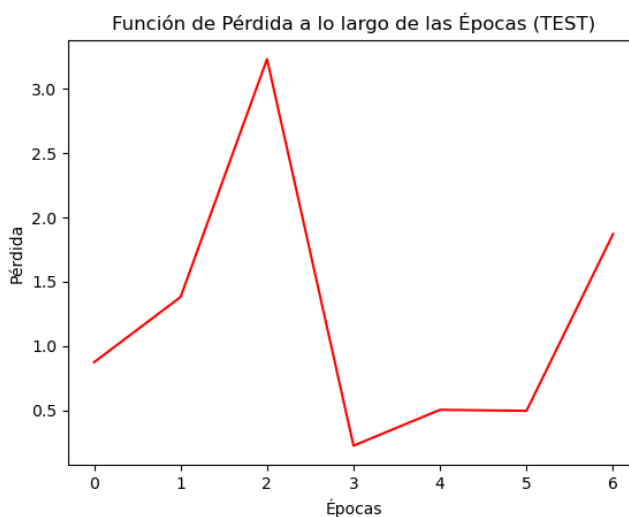
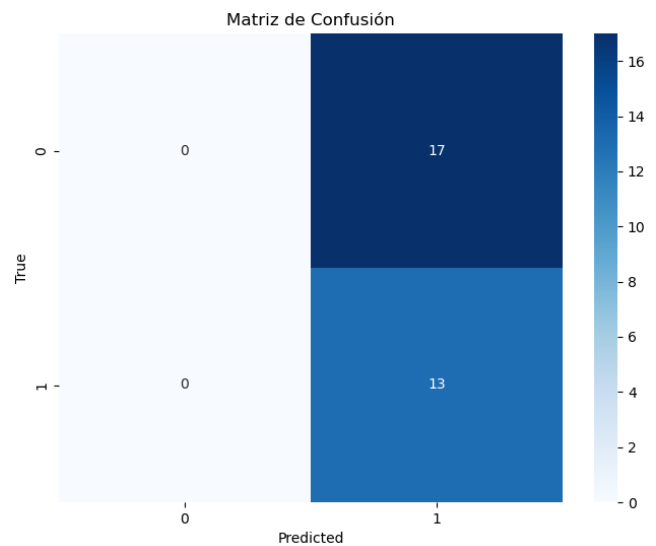
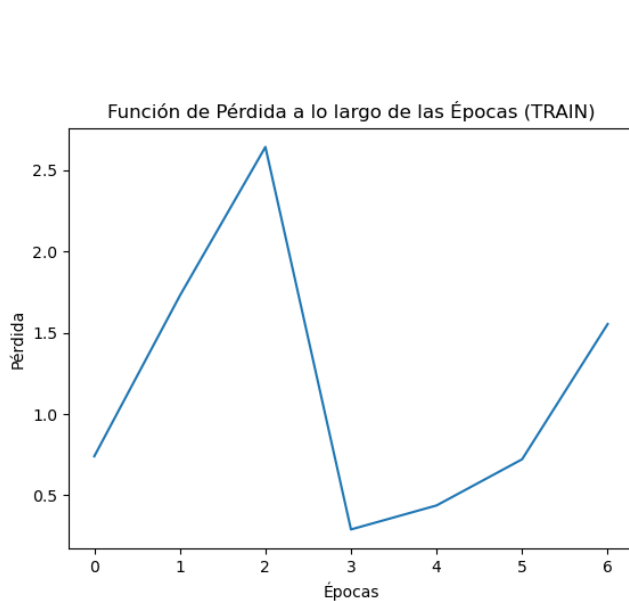
Se muestra una matriz de confusión para las predicciones del conjunto de prueba.

VI. ANALISIS:

En el Código “Gradient_Descent_X.py”, se asignaron los siguientes valores a los hiperparámetros:

- ➔ Learnign Rate: 0.4
- ➔ Epochs: 7

Y se tuvieron estos resultados:



Accuracy (Train)	52.85 %
Error (Train)	47.14 %
Accuracy (Test)	43.33 %
Error (Test)	56.66 %
Score Accuracy (Sklearn)	0.43
Score Precision (Sklearn)	0.43
Score Recall (Sklearn)	1.0
Score F1 (Sklearn)	0.60

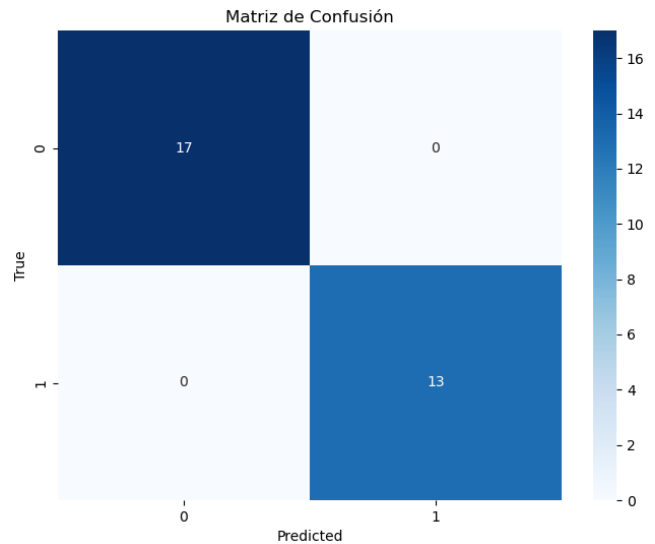
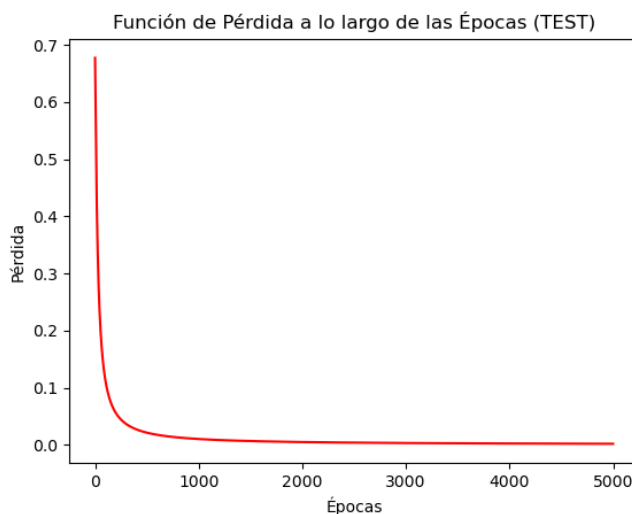
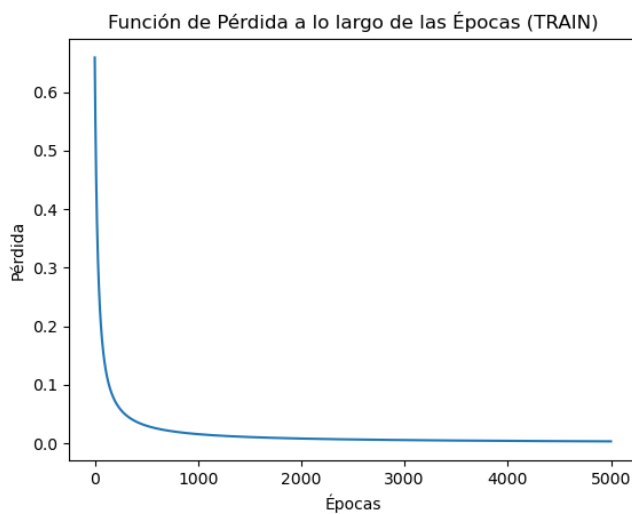
La exactitud en el conjunto de entrenamiento y de prueba no es muy alta, lo que indica que el modelo no está funcionando muy bien.

El recall perfecto de 1.0 sugiere que el modelo puede estar clasificando muchas instancias como positivas, lo que podría explicar la precisión más baja.

En el Código “Gradient_Descent_100.py”, se asignaron los siguientes valores a los hiperparametros:

- Learnign Rate: 0.05
- Epochs: 5, 000

Y se tuvieron estos resultados:



Accuracy (Train)	100.0 %
Error (Train)	0.0 %
Accuracy (Test)	100.0 %
Error (Test)	0.0 %
Score Accurary (Sklearn)	1.0
Score Precision (Sklearn)	1.0
Score Recall (Sklearn)	1.0
Score F1 (Sklearn)	1.0

El modelo tiene una **precisión del 100%** tanto en el conjunto de entrenamiento como en el de prueba, **lo cual es impresionante pero puede ser sospechoso de sobreajuste**. Sin embargo, dada la naturaleza del dataset de Iris al ser muy simple o sencillo, esta alta precisión no es inesperada.