Implementación Manual de Regresión Lineal mediante Descenso de Gradiente

Fernando Daniel Brenes Reyes
Escuela de Ingeniería en Computación
Instituto Tecnológico de Costa Rica
Correo: 2020097446@estudiantec.cr

Jose Pablo Quesada Rodriguez
Escuela de Ingeniería en Computación
Instituto Tecnológico de Costa Rica
Correo: josepabloqr15@estudiantec.cr

Ashley Vasquez Concepcion
Escuela de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica
Correo: ashley19@estudiantec.cr

***Abstract*—This report presents a from-scratch implementation of a linear regression model using gradient descent, applied to the *Student Performance Prediction* dataset. The study aims to provide a comprehensive understanding of the mathematical and computational foundations of linear regression, while systematically examining the impact of data preprocessing, dataset partitioning, and model evaluation strategies. The analysis includes both model performance and the implications of employing manually implemented evaluation metrics, highlighting considerations for rigorous and reproducible predictive modeling.**

I. Introducción

La regresión lineal constituye uno de los pilares fundamentales en el campo del aprendizaje automático y la estadística aplicada [1]. A pesar de su aparente simplicidad, este modelo ha demostrado ser una herramienta poderosa para el análisis predictivo en múltiples disciplinas, desde las ciencias sociales hasta la ingeniería. Su relevancia radica en la capacidad de modelar relaciones lineales entre variables independientes y dependientes, proporcionando interpretaciones intuitivas y resultados cuantitativamente precisos.

Este trabajo se enfoca en construir paso a paso un modelo de regresión lineal utilizando el algoritmo de descenso de gradiente para ajustarlo. A diferencia de métodos que usan bibliotecas listas para usar como `scikit-learn`, este enfoque busca comprender realmente cómo funciona el modelo, incluyendo conceptos como la función de error, cómo se calculan los gradientes y cómo el modelo mejora hasta llegar a un resultado estable.

El conjunto de datos utilizado, llamado *Student Performance Prediction* y disponible en Kaggle, es ideal para aplicar este modelo porque contiene información relacionada con el desempeño académico de los estudiantes. Trabajar con este dataset permite no solo probar la precisión del modelo, sino también reflexionar sobre cómo la regresión lineal puede ser útil en contextos educativos.

En el documento se presentan de manera clara los pasos necesarios para construir el modelo: explorar los datos, tratar valores atípicos, dividir el dataset en grupos de entrenamiento y prueba usando diferentes métodos, implementar el descenso de gradiente y evaluar los resultados con métricas sencillas como el error cuadrático medio (MSE). Finalmente, se muestran los resultados obtenidos y se discuten sus puntos fuertes, limitaciones y posibles mejoras para el futuro.

II. Metodología

El presente trabajo se desarrolló con un enfoque experimental orientado a la implementación manual de un modelo de regresión lineal mediante descenso de gradiente. Este enfoque permitió comprender de forma práctica el impacto de las decisiones de preparación de datos, la división en subconjuntos y la elección de hiperparámetros sobre el desempeño del modelo. A continuación, se describen las etapas seguidas.

A. Conjunto de datos y exploración inicial

Se utilizó el conjunto de datos *Student Performance Prediction*, disponible públicamente en Kaggle[1], el cual contiene variables socioacadémicas y de rendimiento estudiantil. El dataset se cargó en un `DataFrame` de `Pandas`, identificándose un total de **10000** instancias y **6** atributos.

Se revisó la estructura (filas, columnas, tipos de variables) y se obtuvieron estadísticas descriptivas básicas como media, desviación estándar, valores mínimos y máximos. Este análisis permitió conocer la distribución y el rango de cada variable. La Fig. 1 muestra un ejemplo de la estructura del conjunto de datos y la distribución inicial de las variables más relevantes.

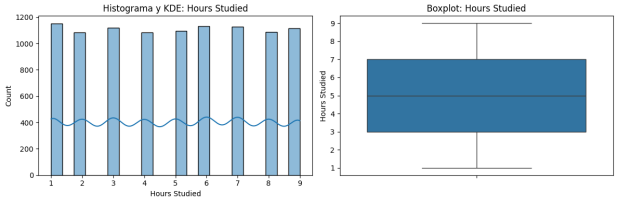

Fig. 1. Ejemplo de la estructura del conjunto de datos y distribución de variables.

B. Análisis exploratorio de datos (EDA)

Para comprender mejor las relaciones entre las variables y la variable objetivo, se realizó un análisis exploratorio que incluyó:

- Histogramas y diagramas de caja para examinar la distribución de variables numéricas.
- Diagramas de dispersión para evaluar relaciones entre predictores y variable objetivo.

[1]https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression/data

- Matriz de correlación representada mediante un mapa de calor para identificar la fuerza y dirección de las relaciones lineales.

Se prestó atención especial a la detección de valores atípicos; en caso de encontrarlos se documentó su tratamiento y, si no existían, se justificó su ausencia. La Fig. 2 presenta la matriz de correlación obtenida.

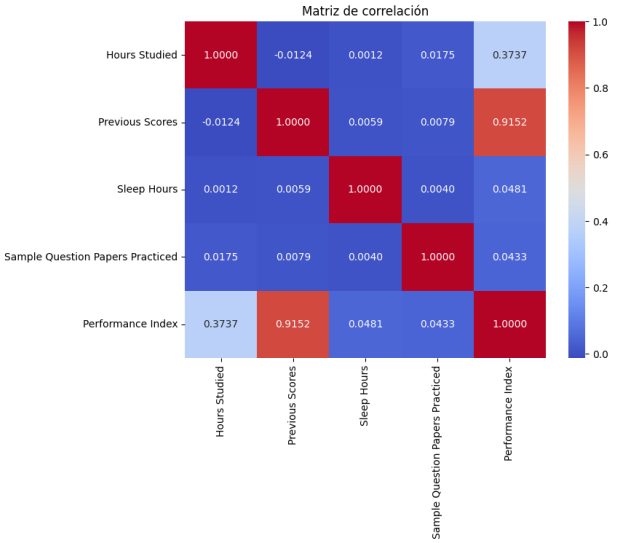

Fig. 2. Matriz de correlación entre predictores y variable objetivo.

C. Preprocesamiento de datos

En esta etapa se realizaron tres tareas principales:

1) Eliminación de instancias duplicadas para reducir redundancia y evitar sesgos en el entrenamiento.
2) Codificación binaria de la variable categórica *Extracurricular Activities* (Yes=1, No=0) para incluirla en el modelo lineal.
3) Normalización de variables numéricas mediante estandarización (*Z-score*), de modo que todas las características tuvieran la misma escala y ninguna dominara el proceso de optimización.

D. División del conjunto de datos

El dataset se dividió en tres subconjuntos: 70% para entrenamiento, 15% para validación y 15% para prueba. Se implementaron manualmente dos estrategias:

- *Random Sampling*: partición aleatoria simple.
- *Stratified Sampling*: partición estratificada para conservar la proporción de la variable objetivo en cada subconjunto.

Esto permitió comparar el impacto de la estrategia de partición en el desempeño del modelo.

E. Implementación del modelo de regresión lineal

El modelo se implementó manualmente en `NumPy` siguiendo la formulación estándar de regresión lineal. La optimización de parámetros se realizó con *Batch Gradient Descent*,técnica ampliamente utilizada en el entrenamiento de modelos predictivos [2]. Se actualizaron los coeficientes según:

$$\theta := \theta - \alpha \cdot \nabla L(\theta) \tag{1}$$

donde α es la tasa de aprendizaje y $\nabla L(\theta)$ es el gradiente de la función de costo. Se experimentó con distintos valores de α y número de iteraciones para observar la dinámica de convergencia. La Fig. 3 ilustra la evolución de la función de costo en cada iteración para los conjuntos de entrenamiento y validación.

Fig. 3. Evolución del error (MSE) en entrenamiento y validación durante el descenso de gradiente.

F. Evaluación del modelo

El rendimiento se evaluó mediante la métrica de Error Cuadrático Medio (MSE), calculada manualmente para cada conjunto. Además, se registró la evolución del costo por iteración y se compararon los resultados de entrenamiento y validación para detectar signos de sobreajuste.

III. Resultados y Discusión

En esta sección se presentan los resultados obtenidos tras la implementación del modelo de regresión lineal con descenso de gradiente. Se compararon dos variantes del algoritmo: *Batch Gradient Descent* y *Mini-Batch Gradient Descent*, evaluados en los conjuntos de entrenamiento, validación y prueba.

A. Curvas de aprendizaje

La Figura 4 muestra la evolución del error cuadrático medio (MSE) a lo largo de las épocas para el *Batch Gradient Descent*, mientras que la Figura 5 ilustra el mismo comportamiento para el *Mini-Batch Gradient Descent*. En ambos casos se observa una convergencia rápida y estable, sin indicios de sobreajuste, dado que las curvas de entrenamiento y validación presentan trayectorias similares.

B. Desempeño cuantitativo

En la Tabla I se reportan los valores finales de MSE para los conjuntos de entrenamiento, validación y prueba. Como se aprecia, ambas variantes alcanzan desempeños similares, con una ligera ventaja para el Mini-Batch, que logra menores valores en los tres subconjuntos.

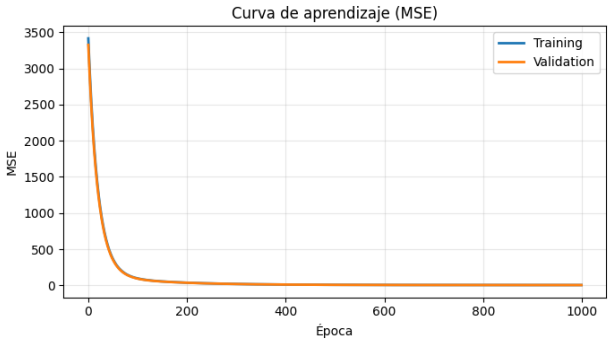

Fig. 4. Curva de aprendizaje (MSE) para Batch Gradient Descent.

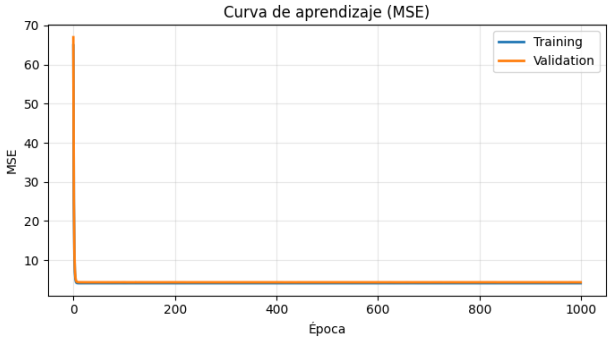

Fig. 5. Curva de aprendizaje (MSE) para Mini-Batch Gradient Descent.

C. Análisis de residuos

La Figura 6 presenta el gráfico de residuos frente a predicciones. Se observa una dispersión aleatoria de los residuos en torno a cero, lo cual confirma que el modelo no incurre en patrones sistemáticos de error y, por lo tanto, su ajuste es adecuado.

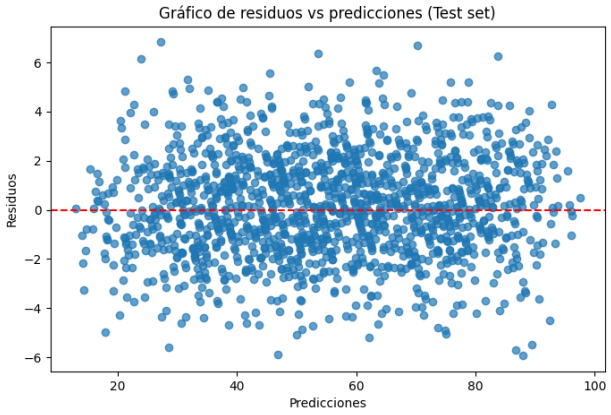

Fig. 6. Distribución de residuos frente a predicciones.

D. Discusión

Los resultados obtenidos confirman que la preparación del conjunto de datos (limpieza, codificación y normalización) y la división estratificada contribuyeron a curvas de aprendizaje estables y a la ausencia de sobreajuste. Tanto el *Batch Gradient Descent* como el *Mini-Batch Gradient Descent* convergieron rápidamente, alcanzando valores finales de MSE cercanos entre sí. La ligera ventaja del Mini-Batch (reducciones de entre 1 % y 2 % en los tres subconjuntos) sugiere que esta estrategia resulta más eficiente computacionalmente, especialmente en escenarios con mayor volumen de datos [3].

TABLE I
COMPARACIÓN DE MSE ENTRE BATCH Y MINI-BATCH GRADIENT DESCENT

Conjunto	Batch MSE	Mini-Batch MSE
Entrenamiento	4.2759	4.2213
Validación	4.2374	4.1904
Prueba	4.0964	4.0479

El análisis de residuos (Fig. 6) muestra una dispersión aproximadamente aleatoria en torno a cero, sin patrones sistemáticos, lo que indica que los supuestos del modelo lineal se cumplen razonablemente y que las predicciones no presentan sesgos evidentes.

En conjunto, estos hallazgos evidencian que la implementación manual de regresión lineal con descenso de gradiente es capaz de modelar de forma efectiva la relación entre las variables predictoras y el rendimiento estudiantil, manteniendo buena capacidad de generalización. Como trabajo futuro se propone aplicar esta misma metodología a conjuntos de datos más grandes y explorar la incorporación de términos no lineales o regularización para evaluar su impacto en el desempeño.

IV. CONCLUSIONES

El presente trabajo logró implementar de manera manual un modelo de regresión lineal optimizado mediante descenso de gradiente sobre el conjunto de datos *Student Performance Prediction*. A lo largo del proceso se realizó una preparación cuidadosa del dataset, que incluyó limpieza, codificación de variables categóricas, normalización y división controlada en subconjuntos de entrenamiento, validación y prueba, garantizando así la consistencia y confiabilidad de los resultados.

La comparación entre *Batch Gradient Descent* y *Mini-Batch Gradient Descent* mostró curvas de aprendizaje estables y convergencia rápida en ambos casos. Los valores de error cuadrático medio fueron similares entre los distintos subconjuntos, lo que indica que el modelo mantiene una buena capacidad de generalización y no presenta indicios de sobreajuste. Esta estabilidad respalda la validez del enfoque utilizado y confirma que cada etapa del procedimiento contribuyó al desempeño alcanzado.

En conjunto, los resultados obtenidos demuestran que es posible, mediante una implementación manual cuidadosamente planificada, aproximar de forma adecuada la relación entre las variables predictoras y el rendimiento estudiantil. Además, se confirma que la metodología seguida permitió cumplir los objetivos planteados y generar un modelo sencillo, reproducible y con resultados consistentes, cerrando satisfactoriamente el estudio realizado.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[3] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks," *arXiv preprint arXiv:1811.08888*, 2018.