

Aplicación y optimización de modelos de Machine Learning para evaluar el rendimiento académico de estudiantes de secundaria

1st Francesco Carlevarino

Facultad de Ingeniería

Universidad del Pacífico

Lima, Peru

fe.carlevarinod@alum.up.edu.pe

2nd Fabrizio Montalvo

Facultad de Ingeniería

Universidad del Pacífico

Lima, Peru

fh.montalvop@alum.up.edu.pe

Resumen—El rendimiento académico de los estudiantes es un factor crucial en su desarrollo educativo y profesional. Este estudio emplea algoritmos de aprendizaje automático para predecir el rendimiento académico de estudiantes de secundaria, considerando una amplia gama de variables, incluidas las socioeconómicas, familiares y académicas. Utilizando un conjunto de datos obtenido de Kaggle que abarca dos cursos, Matemáticas y Portugués, se implementaron cuatro optimizadores: *Adam*, *AdamW*, *SGD* y *BFGS*. El objetivo principal de este trabajo fue optimizar los modelos predictivos mediante técnicas avanzadas de optimización, mejorando la precisión de las predicciones del rendimiento académico. Los resultados muestran que el optimizador *Adam* obtuvo el mejor desempeño en términos de precisión, con los valores más bajos de error cuadrático medio (RMSE) en los tres conjuntos de datos (entrenamiento, validación y prueba), lo que indica una excelente capacidad de generalización. En contraste, *BFGS* presentó un rendimiento significativamente inferior en todos los conjuntos. Este estudio destaca la efectividad de los optimizadores adaptativos y subraya la importancia de optimizar las funciones de costo para mejorar la predicción en problemas educativos complejos.

Index Terms—machine learning, aprendizaje automático, rendimiento académico, predicción educativa.

I. INTRODUCCIÓN

I-A. Contextualización

El rendimiento académico de los estudiantes es un indicador fundamental que influye en su desarrollo educativo y futuro profesional [8]. Este rendimiento puede evaluarse a través de distintos parámetros, como calificaciones generales, tasas de deserción escolar y desempeño en pruebas estandarizadas, tales como el SAT (Scholastic Aptitude Test) en Estados Unidos o la prueba PISA (Program for International Student Assessment) a nivel internacional.

El rendimiento académico no solo está influenciado por factores internos como la motivación y las habilidades cognitivas del estudiante, sino también por variables externas que incluyen aspectos socioeconómicos, características del entorno educativo y contextos familiares. Entre estas, el nivel socioeconómico es uno de los factores más estudiados debido a su impacto directo en las oportunidades y logros académicos. Por ejemplo, en Estados Unidos, el National Center for

Education Statistics (NCES) reporta que los estudiantes provenientes de familias de bajos ingresos tienen un desempeño significativamente menor en pruebas de matemáticas y lectura en comparación con aquellos de familias con ingresos más altos [6]. De manera similar, el informe PISA en Europa muestra diferencias de más del 30 % entre estudiantes de alto y bajo nivel socioeconómico en áreas como matemáticas y comprensión lectora [7].

En este contexto, las técnicas de aprendizaje automático (machine learning) se presentan como una herramienta poderosa para predecir y analizar el rendimiento académico, ya que permiten identificar patrones complejos a partir de grandes volúmenes de datos [1]. Estos modelos pueden integrar una amplia variedad de variables, no solo socioeconómicas, sino también demográficas, psicológicas, escolares y tecnológicas. Por ejemplo, el análisis de variables como el acceso a recursos educativos, la calidad de las instituciones, la frecuencia de uso de tecnologías digitales y el apoyo familiar puede proporcionar una perspectiva integral del rendimiento de los estudiantes.

La presente investigación busca implementar algoritmos de machine learning para predecir el rendimiento académico, integrando múltiples variables relevantes. De esta manera, no solo se analiza cómo afectan las variables socioeconómicas, sino que también se exploran otras dimensiones que contribuyen al desempeño estudiantil. Este enfoque permitirá no solo entender mejor los factores que impactan el aprendizaje, sino también proponer estrategias que puedan cerrar las brechas existentes entre los estudiantes y mejorar la equidad en los sistemas educativos.

I-B. Estado del arte

El impacto de los factores socioeconómicos en el rendimiento académico ha sido ampliamente documentado a nivel mundial, especialmente en contextos de alta desigualdad. Diversos estudios internacionales han demostrado que las condiciones socioeconómicas ejercen una influencia decisiva en la trayectoria educativa de los estudiantes, generando notables disparidades en el acceso y la calidad de la educación.

Por ejemplo, en un estudio realizado en Pensilvania, Chen y Ding (2023) aplican técnicas de aprendizaje automático para revelar que el estatus socioeconómico es uno de los factores más relevantes que afecta el rendimiento académico. Los estudiantes de familias de bajos ingresos suelen enfrentar condiciones menos favorables para el aprendizaje, como menor estabilidad familiar y un acceso limitado a oportunidades educativas de calidad. Estos hallazgos subrayan la importancia de abordar las desigualdades socioeconómicas para mejorar el desempeño escolar a nivel global [3].

En esa misma línea, Castrillón et al. (2022), en Colombia, emplearon técnicas de inteligencia artificial para predecir el rendimiento académico de estudiantes de educación superior en función de factores socioeconómicos, familiares y de hábitos. Su metodología permite clasificar a los estudiantes en cinco categorías con una precisión del 91.7%, facilitando la identificación de aquellos con mayores dificultades y permitiendo orientar acciones preventivas y de apoyo. Esto evidencia el potencial de estas técnicas para mejorar el rendimiento académico en contextos vulnerables.

Asimismo, Falait y Piscova en el año 2022 utilizaron modelos de aprendizaje supervisado para predecir el promedio de calificaciones de los estudiantes, considerando factores psicológicos, sociológicos y académicos. Este estudio destaca la utilidad de las metodologías de aprendizaje automático para identificar y cuantificar los factores que afectan el rendimiento académico en distintos contextos, facilitando el desarrollo de intervenciones orientadas a cerrar las brechas de rendimiento.

Por otro lado, Li y Dockery (2015) examinan el impacto del nivel socioeconómico en el rendimiento académico universitario en Australia, señalando que los estudiantes provenientes de escuelas con bajo estatus socioeconómico enfrentan barreras adicionales para acceder y sobresalir en la educación superior. Incluso tras su ingreso a la universidad, estas barreras continúan afectando su desempeño, evidenciando que las desventajas en etapas escolares tienen efectos duraderos en su éxito académico [5].

En conclusión, el análisis de diversas investigaciones sugiere que los factores socioeconómicos constituyen una barrera significativa para el rendimiento académico. La falta de recursos, la segregación escolar y las oportunidades educativas desiguales limitan el desarrollo académico de los estudiantes. Además, los hallazgos de estos estudios respaldan la necesidad de políticas educativas que promuevan la equidad, amplíen el acceso a recursos y brinden apoyo continuo a estudiantes de bajos ingresos para mitigar los efectos de estas desigualdades.

II. PLANTEAMIENTO DEL PROBLEMA

Los estudiantes de secundaria se encuentran en una etapa crítica de su desarrollo académico y personal, durante la cual diversas variables pueden influir significativamente en su rendimiento escolar. Factores como el entorno familiar, las condiciones socioeconómicas y las prácticas de estudio desempeñan un papel fundamental en su éxito o fracaso académico [4]. En los últimos años, el uso de algoritmos de machine learning ha permitido a los investigadores predecir el

rendimiento escolar, centrándose en factores como el apoyo familiar (por ejemplo, el nivel de ingresos de los padres) y las condiciones del hogar (acceso a internet y recursos educativos). No obstante, estos estudios suelen excluir variables adicionales, como los hábitos de salud (consumo de alcohol, horas de sueño) y otros factores familiares (nivel educativo de los padres), lo cual podría limitar la precisión y efectividad de los modelos predictivos.

Dado el potencial de los algoritmos de machine learning para analizar múltiples variables simultáneamente y captar patrones complejos en el rendimiento académico [?], es esencial evaluar cómo la inclusión de una gama más amplia de variables podría mejorar la precisión de las predicciones y ofrecer una visión más integral de los factores determinantes.

Para mejorar la precisión predictiva, en este proyecto integramos técnicas avanzadas de optimización. Estos métodos buscan mejorar la optimización de la función de costos a través de el uso de diferentes optimizadores. La aplicación de estrategias de optimización refinará el modelo, llevando a mejores predicciones de rendimiento y una comprensión más profunda de las variables que impactan el éxito académico.

III. OBJETIVOS

III-A. Objetivo General

Optimizar los modelos de aprendizaje automático utilizados para predecir el rendimiento académico de estudiantes de secundaria, mejorando la precisión predictiva mediante técnicas avanzadas de optimización de la función de costos.

III-B. Objetivos Específicos

- Implementar optimizadores avanzados para la mejora de la función de costos en los modelos de machine learning.
- Evaluar el impacto de diferentes estrategias de optimización en la precisión y eficiencia de los modelos.
- Comparar el rendimiento de los modelos optimizados con los modelos iniciales para determinar las mejoras en la predicción del rendimiento académico.

IV. MARCO TEÓRICO

En el campo del aprendizaje automático, los *optimizadores* son algoritmos clave para entrenar modelos, ajustando los parámetros del modelo con el objetivo de minimizar la función de costo o pérdida. Estos algoritmos mejoran la capacidad de los modelos para hacer predicciones precisas y converger hacia una solución óptima. A continuación, se describen cuatro de los optimizadores a utilizar en este proyecto: *Adam*, *AdamW*, *SGD* y *BFGS*.

IV-A. Adam (*Adaptive Moment Estimation*)

Adam es uno de los optimizadores más utilizados en el aprendizaje profundo debido a su capacidad para manejar grandes volúmenes de datos y su eficiencia en tareas complejas. Este optimizador combina las ventajas de dos técnicas anteriores: *momentum* y *RMSprop*. *Adam* se basa en el cálculo de dos momentos estadísticos:

- El *primer momento* (m), que representa la media móvil de los gradientes.
- El *segundo momento* (v), que es la media móvil de los cuadrados de los gradientes.

El algoritmo ajusta la tasa de aprendizaje para cada parámetro de forma adaptativa, lo que permite que los parámetros con gradientes grandes tengan actualizaciones pequeñas, mientras que los parámetros con gradientes pequeños tienen actualizaciones mayores. Esto mejora la convergencia y evita problemas como el estancamiento en los mínimos locales.

Las actualizaciones de los parámetros en Adam son las siguientes:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L(\theta)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_\theta L(\theta))^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

IV-B. AdamW (Adam with Weight Decay)

AdamW es una variante de Adam que introduce una mejora importante en la regularización. En Adam, la regularización L2 (también conocida como *weight decay*) se implementa de manera implícita en la actualización de los parámetros, lo que puede ser menos efectivo. AdamW, por otro lado, separa explícitamente la regularización de la actualización de los gradientes, haciendo que el modelo se regularice de forma más efectiva durante el entrenamiento.

La fórmula de actualización de AdamW es similar a la de Adam, pero con un término de regularización explícito:

$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \lambda \theta_{t-1}$$

Donde λ es la tasa de decaimiento del peso.

IV-C. SGD (Stochastic Gradient Descent)

SGD es uno de los optimizadores más simples y ampliamente utilizados en el aprendizaje automático. A diferencia de Adam, que utiliza un enfoque adaptativo, SGD actualiza los parámetros del modelo usando el gradiente del error para cada muestra de entrenamiento. El algoritmo calcula el gradiente de la función de costo con respecto a los parámetros y luego ajusta estos parámetros en la dirección opuesta al gradiente, lo que reduce el error.

La fórmula de actualización en SGD es la siguiente:

$$\theta_t = \theta_{t-1} - \alpha \nabla_\theta L(\theta)$$

IV-D. BFGS (Broyden–Fletcher–Goldfarb–Shanno)

BFGS es un optimizador basado en la *optimización de cuasi-Newton*. A diferencia de los optimizadores basados en gradientes como Adam y SGD, BFGS utiliza una aproximación de la *matriz Hessiana* (segunda derivada de la función de costo) para mejorar la actualización de los parámetros. Esto hace que BFGS sea más preciso en la estimación de la dirección del descenso, especialmente en problemas donde el gradiente no es suficiente para encontrar la dirección correcta.

BFGS es un algoritmo *de segundo orden*, lo que significa que tiene en cuenta tanto la primera derivada (gradiente) como la segunda derivada (curvatura). La fórmula general de actualización para BFGS es más compleja, pero la idea central es utilizar una aproximación a la matriz Hessiana para calcular un mejor paso de actualización.

La fórmula de actualización para la aproximación de la inversa de la matriz Hessiana H_k en BFGS es la siguiente:

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}$$

Donde:

- H_k es la aproximación de la matriz Hessiana inversa en la iteración k .
- $s_k = \theta_{k+1} - \theta_k$ es el cambio en los parámetros del modelo entre la iteración k y $k + 1$.
- $y_k = \nabla f(\theta_{k+1}) - \nabla f(\theta_k)$ es el cambio en el gradiente de la función de costo entre las iteraciones k y $k + 1$.

Este proceso permite que el algoritmo BFGS actualice la aproximación de la Hessiana sin tener que calcularla directamente, lo que hace que sea computacionalmente más eficiente que los métodos de Newton, que requieren el cálculo explícito de la Hessiana.

Criterio	SGD	ADAM	ADAMW	BFGS
Tipo	Descenso estocástico	Adaptativo por momentos	ADAM con weight decay separado	Cuasi-Newton
Learning Rate	Constante	Adaptativa	Adaptativa	Ajustada con Hessiana aprox.
Momentos	Opcional	Sí	Sí	Hessiana aprox.
Regularización	Externa	L2 mezclada	Weight decay des-acoplado	No explícita
Convergencia	Lenta	Rápida	Rápida y estable	Muy rápida en problemas pequeños
Costo	Bajo	Medio	Medio	Alto
Escalabilidad	Muy alta	Alta	Muy alta	Baja

Cuadro I

RESUMEN COMPARATIVO DE OPTIMIZADORES.

V. METODOLOGÍA

En este proyecto se siguió un enfoque de Machine Learning para predecir el rendimiento académico de estudiantes de secundaria, utilizando un conjunto de datos obtenido de Kaggle.

Este conjunto de datos contiene registros de estudiantes de dos cursos: Matemáticas y Portugués, con variables socioeconómicas, demográficas y relacionadas con la escuela.

El dataset contiene un total de 650 registros para el curso de Portugués y 396 registros para el curso de Matemáticas. A continuación, se describen las variables incluidas en el conjunto de datos:

- **school** - Escuela del estudiante (binario: "GP Gabriel Pereira o "MS Mousinho da Silveira)
- **sex** - Sexo del estudiante (binario: "F femenino o "M- masculino)
- **age** - Edad del estudiante (numérico: de 15 a 22 años)
- **address** - Tipo de dirección del hogar del estudiante (binario: U urbano o R rural)
- **famsize** - Tamaño de la familia (binario: "LE3 menor o igual a 3 o "GT3 mayor que 3)
- **Pstatus** - Estado de convivencia de los padres (binario: "T viviendo juntos o .^ separados)
- **Medu** - Nivel educativo de la madre (numérico: 0 - ninguna, 1 - educación primaria, 2 - de 5º a 9º grado, 3 - educación secundaria o 4 - educación superior)
- **Fedu** - Nivel educativo del padre (numérico: 0 - ninguno, 1 - educación primaria, 2 - de 5º a 9º grado, 3 - educación secundaria o 4 - educación superior)
- **Mjob** - Trabajo de la madre (nominal: "teacher", "healthrelacionado con salud, "servicesciviles, .@home" o "other")
- **Fjob** - Trabajo del padre (nominal: "teacher", "healthrelacionado con salud, "servicesciviles, .@home" o "other")
- **reason** - Motivo para elegir esta escuela (nominal: "home cerca de casa, reputation reputación de la escuela, course preferencia de curso o .^other")
- **guardian** - Tutor del estudiante (nominal: "mother", "fatherü .^other")
- **traveltime** - Tiempo de viaje de casa a la escuela (numérico: 1 - <15 min, 2 - 15 a 30 min, 3 - 30 min a 1 hora, 4 - >1 hora)
- **studytme** - Tiempo semanal de estudio (numérico: 1 - <2 horas, 2 - 2 a 5 horas, 3 - 5 a 10 horas, 4 - >10 horas)
- **failures** - Número de fracasos académicos previos (numérico: n si 1<=n<3, de lo contrario 4)
- **schoolsup** - Apoyo educativo adicional (binario: sí o no)
- **famsup** - Apoyo educativo familiar (binario: sí o no)
- **paid** - Clases adicionales pagadas dentro de la materia (binario: sí o no)
- **activities** - Actividades extracurriculares (binario: sí o no)
- **nursery** - Asistencia a la escuela infantil (binario: sí o no)
- **higher** - Desea cursar estudios superiores (binario: sí o no)
- **internet** - Acceso a Internet en casa (binario: sí o no)
- **romantic** - Relación romántica (binario: sí o no)
- **famrel** - Calidad de las relaciones familiares (numérico:

de 1 - muy mala a 5 - excelente)

- **freetime** - Tiempo libre después de la escuela (numérico: de 1 - muy bajo a 5 - muy alto)
- **goout** - Salir con amigos (numérico: de 1 - muy bajo a 5 - muy alto)
- **Dalc** - Consumo de alcohol entre semana (numérico: de 1 - muy bajo a 5 - muy alto)
- **Walc** - Consumo de alcohol en fin de semana (numérico: de 1 - muy bajo a 5 - muy alto)
- **health** - Estado de salud actual (numérico: de 1 - muy malo a 5 - muy bueno)
- **absences** - Número de ausencias escolares (numérico: de 0 a 93)
- **G1** - Nota del primer periodo (numérico: de 0 a 20)
- **G2** - Nota del segundo periodo (numérico: de 0 a 20)
- **G3** - Nota final (numérico: de 0 a 20, objetivo de predicción)

V-A. Análisis Exploratorio de Datos

Antes de comenzar con el preprocesamiento y la construcción del modelo, se realizó un análisis exploratorio de los datos (EDA, por sus siglas en inglés). El objetivo de este análisis fue obtener una visión general de las características del conjunto de datos, detectar posibles inconsistencias y entender las relaciones entre las variables.

Se comenzó con un análisis descriptivo de las variables numéricas y categóricas del conjunto de datos. Las principales variables de interés incluyen el sexo del estudiante, su edad, el tamaño de la familia, el estado de salud, el tiempo de estudio, las calificaciones y otras variables socioeconómicas y familiares.

Además, se generaron gráficos de distribución, como histogramas y diagramas de caja (boxplots), para visualizar la dispersión de las variables numéricas y detectar posibles valores atípicos (outliers). Para las variables categóricas, se crearon diagramas de barras para observar la distribución de las diferentes categorías.

El análisis preliminar mostró que las variables estaban bastante equilibradas, con una ligera inclinación hacia un mayor número de estudiantes de sexo femenino, y un número equilibrado de estudiantes de las dos escuelas disponibles (GP y MS). También se observó que un pequeño porcentaje de estudiantes tenía un número elevado de ausencias, lo que podría influir en el rendimiento académico.

V-B. Preprocesamiento de Datos

Primero, se cargó y se inspeccionó el conjunto de datos, que incluía características como el sexo, la edad, el tamaño de la familia, el estado de salud, las calificaciones y otros factores. A continuación, se realizó un proceso de codificación de variables categóricas utilizando el *LabelEncoder* de la librería sklearn, lo que convirtió las variables cualitativas en valores numéricos.

Se llevó a cabo una división de los datos en tres conjuntos: entrenamiento (70 %), validación (15 %) y prueba (15 %) para evaluar el rendimiento del modelo en datos no vistos.

V-C. Construcción y Entrenamiento del Modelo

Se construyó un modelo de red neuronal utilizando la librería Keras. El modelo contó con dos capas ocultas y una capa de salida con activación *sigmoid* para la predicción de la variable de rendimiento académico. Debido a que el rango de notas iba de 0 a 20, se dividió la variable objetivo entre 20 para que la sigmoide la prediga correctamente.

El modelo se compiló utilizando diferentes optimizadores y funciones de pérdida. Para los optimizadores *Adam*, *AdamW* y *SGD*, se utilizó la función de pérdida *mean_squared_error* propia de Keras. Para el optimizador *BFGS*, se implementó una clase personalizada desde cero, que utilizó una función de pérdida basada en *scipy.optimize* para realizar la optimización.

Para el entrenamiento, se utilizó el conjunto de entrenamiento para ajustar los pesos del modelo, y se validó el modelo en el conjunto de validación para prevenir el sobreajuste.

V-D. Evaluación del Modelo

Una vez entrenado el modelo, se evaluó su rendimiento utilizando el conjunto de prueba. Se compararon las pérdidas (loss) del entrenamiento y la validación, así como las métricas de rendimiento en el conjunto de prueba. Las métricas clave incluyeron la pérdida de entrenamiento, la pérdida de validación y la pérdida en el conjunto de prueba, que se utilizaron para comparar la efectividad de los diferentes optimizadores y funciones de pérdida.

V-E. Visualización de Resultados

Para analizar el rendimiento de los modelos, se graficaron las curvas de pérdida durante el entrenamiento y la validación. Además, se compararon las pérdidas de los distintos modelos para determinar cuál de ellos ofreció el mejor rendimiento en términos de generalización y precisión.

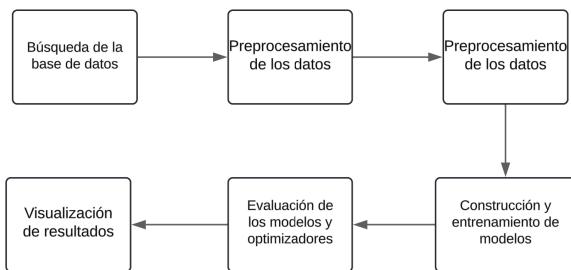


Figura 1. Flujograma del proceso seguido en la metodología del proyecto.

VI. RESULTADOS

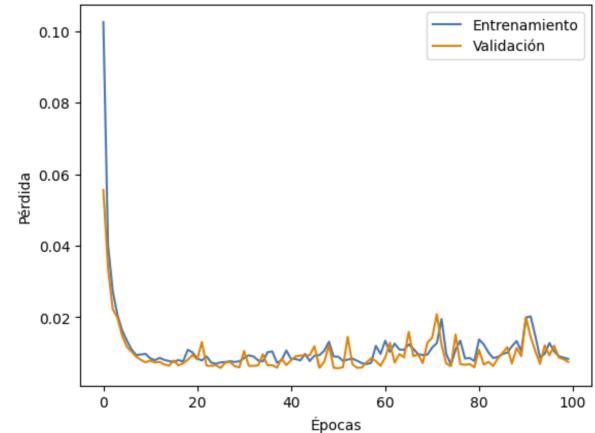


Figura 2. Curva de pérdida durante el entrenamiento y la validación con el optimizador Adam.

La pérdida de entrenamiento disminuye rápidamente al principio, mientras que la de validación muestra pequeñas fluctuaciones, lo que sugiere un buen ajuste del modelo sin sobreajuste significativo.

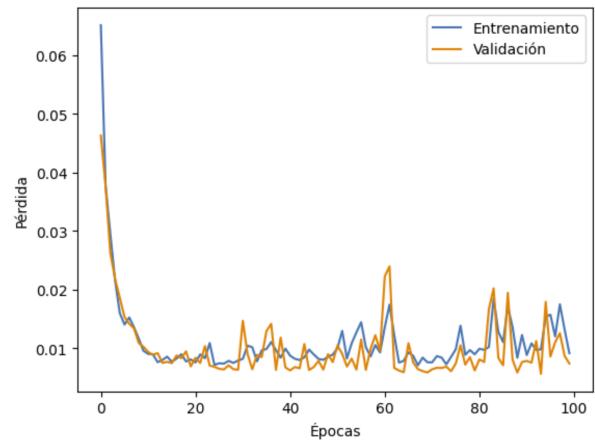


Figura 3. Curva de pérdida durante el entrenamiento y la validación con el optimizador AdamW.

Aunque la pérdida de entrenamiento disminuye considerablemente al principio, la pérdida de validación muestra fluctuaciones, lo que puede indicar un leve sobreajuste o una adaptación mejor a los datos de validación.

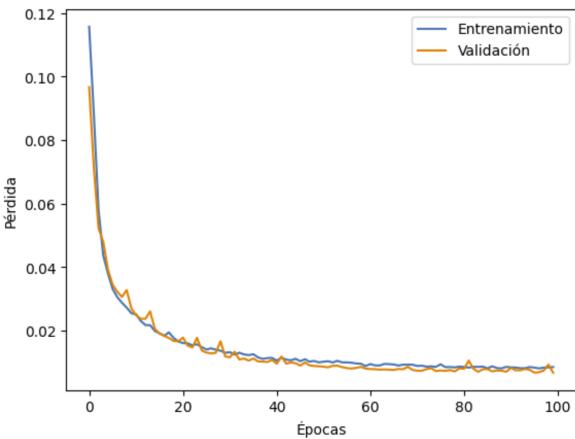


Figura 4. Curva de pérdida durante el entrenamiento y la validación con el optimizador SGD.

La pérdida de entrenamiento disminuye rápidamente al principio y luego se estabiliza, mientras que la pérdida de validación permanece estable, indicando una buena generalización y ausencia de sobreajuste.

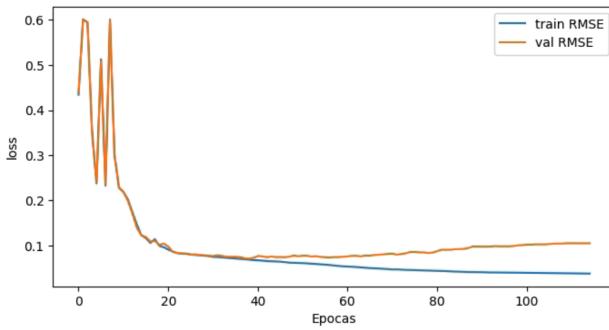


Figura 5. Curva de pérdida durante el entrenamiento y la validación con el optimizador BFGS.

La pérdida de entrenamiento disminuye rápidamente, mientras que la de validación muestra pequeñas fluctuaciones, sugiriendo un buen ajuste sin sobreajuste.

Después del entrenamiento de los modelos, se evaluaron utilizando las métricas de error cuadrático medio (RMSE) para los conjuntos de entrenamiento, validación y prueba. A continuación, se presentan los resultados obtenidos para los diferentes optimizadores probados: *Adam*, *AdamW*, *SGD* y *BFGS*. La comparativa de RMSE muestra cómo cada optimizador se comportó en términos de ajuste y generalización a los datos.

VI-A. Comparación de RMSE

Modelo	train_RMSE	val_RMSE	test_RMSE
Adam	1.732	1.733	2.062
AdamW	1.974	1.720	2.412
SGD	1.763	1.646	2.283
BFGS	8.410	8.691	9.028

Cuadro II
RESULTADOS DE RMSE PARA LOS MODELOS DE OPTIMIZACIÓN PROBADOS

Los resultados muestran que el optimizador *Adam* presentó el mejor rendimiento en términos de precisión, con los valores de RMSE más bajos en los tres conjuntos de datos: entrenamiento (1.732), validación (1.733) y prueba (2.062). Esto indica que el modelo entrenado con *Adam* generaliza bien en los datos no vistos.

El optimizador *SGD* mostró un rendimiento competitivo, especialmente en el conjunto de validación (1.646), con un valor de RMSE en el conjunto de prueba de 2.283. Sin embargo, su desempeño en el conjunto de entrenamiento (1.763) fue ligeramente peor que el de *Adam*.

El optimizador *AdamW* presentó un rendimiento similar al de *Adam* en el conjunto de validación (1.720), pero su desempeño en los conjuntos de entrenamiento (1.974) y prueba (2.412) fue algo peor.

Por otro lado, el optimizador *BFGS* mostró un desempeño significativamente inferior en comparación con los otros optimizadores, con valores de RMSE mucho más altos en los tres conjuntos de datos (entrenamiento: 8.410, validación: 8.691, prueba: 9.028). Estos resultados sugieren que el método *BFGS* no es adecuado para este conjunto de datos y no logró un buen ajuste del modelo.

En resumen, el optimizador *Adam* fue el más efectivo en la predicción del rendimiento académico, mientras que *BFGS* presentó los peores resultados, tanto en términos de ajuste como de generalización.

VII. CONCLUSIONES

El proyecto logró cumplir el objetivo general de optimizar los modelos de aprendizaje automático para predecir el rendimiento académico, demostrando que la elección del optimizador tiene un impacto directo en la precisión final del modelo. Entre las alternativas evaluadas, *Adam* obtuvo el mejor desempeño, alcanzando los valores más bajos de RMSE en los conjuntos de entrenamiento, validación y prueba, lo que confirma su eficacia para problemas con múltiples variables y datos limitados.

1. En relación con el primer objetivo específico, la implementación de los optimizadores *Adam*, *AdamW*, *SGD* y *BFGS* permitió comparar distintas estrategias de actualización. Los optimizadores adaptativos (*Adam* y *AdamW*) mostraron una convergencia rápida y una reducción sostenida de la pérdida, mientras que *SGD* presentó un comportamiento más estable pero con convergencia más lenta. *BFGS*, pese a su rapidez inicial, no logró un ajuste adecuado al conjunto de datos, reflejando

- que los métodos de segundo orden no son adecuados para este tipo de modelo ni para la escala del problema.
2. Respecto al segundo objetivo específico, los resultados de RMSE evidencian que las estrategias de optimización influyen significativamente en la precisión. Adam fue el más consistente en todos los conjuntos, seguido por SGD, que mostró un desempeño competitivo en validación. AdamW presentó cierta estabilidad pero sin superar a Adam, mientras que BFGS registró errores considerablemente mayores, lo que confirma su limitada capacidad de generalización en este contexto.
 3. Finalmente, en cuanto al tercer objetivo específico, la comparación entre los modelos optimizados y los modelos base demuestra que la optimización mejora la generalización y reduce el error de predicción. El comportamiento de Adam y SGD evidencia una mejora real respecto a configuraciones sin optimización avanzada. En contraste, el bajo desempeño de BFGS confirma que no todas las técnicas avanzadas producen beneficios en modelos neuronales de este tipo.

En el caso de BFGS, el modelo presentó un RMSE significativamente superior al de los demás optimizadores. Una posible explicación es la combinación de dos factores: la naturaleza no convexa y de alta dimensionalidad del modelo de red neuronal, para la cual los métodos de segundo orden como BFGS pueden resultar menos adecuados, y las limitaciones prácticas de la implementación personalizada del algoritmo, que depende de una integración externa con `scipy.optimize`. En conjunto, estos elementos podrían haber comprometido la calidad de las actualizaciones y explicado el bajo rendimiento observado.

En conjunto, los resultados indican que Adam es el optimizador más adecuado para este problema, mientras que SGD constituye una alternativa viable con buen equilibrio entre estabilidad y desempeño. AdamW ofrece beneficios en regularización, pero no supera a Adam en este caso. BFGS no resulta recomendable debido a su bajo rendimiento y su incompatibilidad con modelos de mayor dimensionalidad. Estas conclusiones permiten orientar futuras implementaciones hacia métodos adaptativos y datasets más amplios para mejorar la exactitud predictiva del sistema.

REFERENCIAS

- [1] Agyemang, E. F., et al. (2024). Predicting students' academic performance via machine learning algorithms: An empirical review and practical application. *Computer Engineering and Intelligent Systems*.
- [2] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 281–305.
- [3] Chen, S., & Ding, Y. (2023). A machine learning approach to predicting academic performance in Pennsylvania's schools. *Social Sciences*, 12(3), 118.
- [4] Díaz, A. L. (2003). Factores personales, familiares y académicos que afectan al fracaso escolar en la Educación Secundaria. *Electronic Journal of Research in Educational Psychology*, 1(1), 43-66.
- [5] Li, I. W., & Dockery, A. (2015). Does school socio-economic status influence university outcomes?. *Australian Journal of Labour Economics*, 18(1), 75-94.
- [6] NCES. (2023). Report on the Condition of Education 2023. National Center for Education Statistics.
- [7] OCDE. (2019). PISA 2018 Results: What students know and can do (Volume I). OCDE Publishing.
- [8] Tomasevic, N., et al. (2020). Academic achievement prediction in higher education through interpretable modeling. *PLOS ONE*.