Facultad de Ingeniería Mecánica y Eléctrica

Universidad Autónoma de Nuevo León

Programación para Inteligencia Artificial

Grupo: 001 | Equipo: 03

Predicción de calificaciones de examen con regresión lineal

Actividad Fundamental 3

Fecha: 28/11/2025

Profesor: Dr. Erick de Jesús Ordaz Rivas

Integrantes

Nombre	Matrícula	Hora clase
Peyton Oved Sánchez Olguín	2152791	V1
Francisco Gabriel Delgado Ortiz	2091878	V1

Índice

1. Objetivo

Analizar un conjunto de datos reales sobre rendimiento académico y *hábitos de estudio* para diseñar, implementar y evaluar un modelo de aprendizaje supervisado basado en regresión lineal, con el fin de predecir la calificación final de un examen a partir de variables académicas y de estilo de vida.

2. Introducción

El aprendizaje supervisado es una de las ramas centrales de la inteligencia artificial y del aprendizaje automático. En este paradigma, el modelo recibe ejemplos de entrada–salida y aprende una función que aproxima la relación entre ambas variables [3, 4]. En problemas de regresión, la variable objetivo es numérica y el objetivo principal es predecir valores continuos, como precios, demandas o calificaciones.

En el contexto educativo, los modelos de regresión pueden ayudar a identificar factores que influyen en el desempeño de los estudiantes y a anticipar riesgos académicos. Esto permite proponer estrategias de acompañamiento y toma de decisiones informadas [9]. En este trabajo se utiliza el **Exam Score Prediction Dataset** disponible en la plataforma Kaggle, el cual recopila información sobre hábitos de estudio, asistencia, características personales y estilo de vida de estudiantes, junto con su calificación de examen [2].

El objetivo específico es construir un modelo de regresión lineal que prediga la calificación de examen a partir de las variables disponibles, evaluando su desempeño con métricas estándar e interpretando los resultados para el contexto educativo.

3. Descripción del caso de estudio y del dataset

3.1. Selección del caso de estudio

Para esta actividad se eligió un problema de **regresión**, donde la tarea principal es **predecir la calificación final de un examen** a partir de distintas características del estudiante, sus hábitos de estudio y algunos factores de su entorno.

El conjunto de datos utilizado se titula *Exam Score Prediction Dataset* y se encuentra disponible de forma pública en la plataforma Kaggle [2]. La fuente oficial del dataset es:

`https://www.kaggle.com/datasets/kundanbedmutha/exam-score-prediction-dataset`

El archivo original se proporciona en formato `.csv` y contiene más de **200 registros** y al menos **4 variables**, por lo que cumple con los requisitos mínimos establecidos en la actividad. Cada fila representa a un estudiante y describe, entre otras, las siguientes dimensiones:

- **Variables académicas**: horas de estudio, rendimiento previo y asistencia.
- **Variables de hábitos de vida**: horas de sueño y manejo del tiempo libre.

- **Variables de contexto**: factores del entorno familiar o social reportados.

La **variable objetivo** del modelo es la *calificación de examen* (*exam score*), que se registra como un valor numérico continuo. Esto confirma que el problema se clasifica como un problema de **regresión** y justifica el uso de un modelo de *regresión lineal múltiple* para el análisis.

Tabla 1: Ejemplos de variables del *Exam Score Prediction Dataset*.

Tipo	Descripción general
Académicas	Horas de estudio, asistencia a clase, resultados previos.
Conductuales	Participación en actividades extracurriculares, uso de recursos.
Estilo de vida	Horas de sueño, tiempo libre, hábitos diarios.
Entorno y contexto	Factores familiares o ambientales reportados en el dataset.
Variable objetivo	Calificación numérica de examen (*exam score*).

Dado que el objetivo es predecir una calificación numérica, el problema se clasifica como **regresión**. Por ello se eligió el algoritmo de *regresión lineal múltiple*, implementado con la clase `LinearRegression` de la biblioteca `scikit-learn` [8].

4. Preprocesamiento y normalización de datos

El preprocesamiento de datos es una etapa esencial en el desarrollo de modelos de aprendizaje supervisado, ya que garantiza que la información utilizada sea consistente, limpia y adecuada para el entrenamiento. Para este caso de estudio se aplicaron los pasos siguientes:

4.1. Exploración inicial

Se cargó el archivo `.csv` utilizando la librería `pandas`. En esta fase se revisaron:

- la estructura del dataframe,
- los nombres de las columnas,
- los tipos de datos,
- la presencia de valores nulos o registros incompletos,
- estadísticas descriptivas básicas.

Esta inspección permitió comprender la naturaleza de cada variable y determinar qué transformaciones serían necesarias antes del modelado.

4.2. Manejo de valores nulos y datos atípicos

Durante la exploración se identificaron posibles valores faltantes. Las acciones aplicadas fueron:

- eliminación de filas con valores nulos cuando su proporción era mínima,
- imputación con valores promedio en variables numéricas cuando fue necesario,
- revisión visual de valores atípicos mediante estadísticas descriptivas.

Estas acciones permitieron mantener la calidad y consistencia del conjunto de datos.

4.3. Codificación de variables categóricas

En caso de variables categóricas, se utilizó el método *one-hot encoding* mediante:

```
pandas.get_dummies()
```

Esta codificación convierte categorías en columnas binarias compatibles con modelos numéricos como la regresión lineal.

4.4. Normalización de variables

Dado que el modelo de regresión lineal es sensible a la escala de los datos, se aplicó estandarización a todas las variables numéricas mediante `StandardScaler` de `scikit-learn`:

- cada característica se transformó a media cero,
- desviación estándar igual a uno,
- evitando que variables con escalas grandes dominen el entrenamiento.

Esta normalización se realizó únicamente con datos de entrenamiento para evitar fuga de información (*data leakage*). Posteriormente, los mismos parámetros fueron aplicados al conjunto de prueba.

4.5. Dataframe antes y después del preprocesamiento

Para evidenciar el efecto del preprocesamiento, en el cuaderno de Python se mostraron:

- las primeras filas del dataframe original,
- las primeras filas del dataframe ya transformado (con columnas codificadas y normalizadas).

Estas comparaciones permiten verificar que las transformaciones fueron correctas y adecuadas al caso de estudio.

5. Implementación del modelo supervisado

5.1. Selección del algoritmo

Como se estableció en la descripción del caso de estudio, la variable objetivo es la calificación de examen (*exam score*), un valor numérico continuo. Por ello, el problema se clasifica como uno de **regresión**.

Se eligió el algoritmo de **regresión lineal múltiple**, implementado a través de la clase `LinearRegression` de la librería `scikit-learn` [8]. Este modelo permite estimar la relación lineal entre las variables predictoras (horas de estudio, hábitos, contexto, etc.) y la calificación final del examen.

5.2. División entrenamiento–prueba

Para evaluar el desempeño del modelo de forma objetiva se dividió el conjunto de datos en dos particiones:

- **70 %** de los datos para entrenamiento (*train*),
- **30 %** de los datos para prueba (*test*).

La división se realizó con la función `train_test_split`, utilizando un valor fijo de `random_state` para garantizar la reproducibilidad de los resultados.

5.3. Código en Python

A continuación se muestra un fragmento representativo del código utilizado para la implementación del modelo de regresión lineal. En el cuaderno de trabajo se incluyen celdas adicionales para la exploración y visualización de los datos.

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import (mean_squared_error,
                             mean_absolute_error,
                             r2_score)

import matplotlib.pyplot as plt

# 1. Carga del dataset --------------------------------------------
# Ajustar el nombre del archivo CSV seg n el dataset descargado
ruta_csv = "exam_score_prediction.csv"
df = pd.read_csv(ruta_csv)

print("Primeras filas del dataframe:")
print(df.head())
```

```
print(df.info())

# 2. Separaci n de variables -------------------------------------
# Se asume que la  ltima  columna corresponde a la calificaci n de examen
X = df.iloc[:, :-1]   # variables predictoras
y = df.iloc[:, -1]    # variable objetivo (exam score)

# 3. Divisi n train / test (70 % / 30 %) -------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.30,
    random_state=42
)

# 4. Normalizaci n de caracter sticas ----------------------------
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled  = scaler.transform(X_test)

# 5. Definici n y entrenamiento del modelo -----------------------
modelo = LinearRegression()
modelo.fit(X_train_scaled, y_train)

# 6. Predicci n en el conjunto de prueba -------------------------
y_pred = modelo.predict(X_test_scaled)

# 7. C lculo de m tricas de desempe o ----------------------------
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2  = r2_score(y_test, y_pred)

print(f"Error cuadr tico medio (MSE): {mse:.3f}")
print(f"Error absoluto medio (MAE):  {mae:.3f}")
print(f"Coeficiente de determinaci n (R^2): {r2:.3f}")
```

Listing 1: Implementación del modelo de regresión lineal para la predicción de calificaciones.

5.4. Justificación de las decisiones de diseño

Las principales decisiones de diseño del modelo fueron:

- **Uso de regresión lineal múltiple**: adecuada para una variable objetivo numérica y fácil de interpretar.
- **División 70/30**: permite contar con suficientes datos para entrenar el modelo y reservar una fracción significativa para evaluar su desempeño en ejemplos no vistos.
- **Normalización con `StandardScaler`**: evita que características con escalas muy distintas dominen el proceso de ajuste y mejora la estabilidad numérica del modelo.

- **Fijar `random_state`**: garantiza que los resultados puedan reproducirse en futuras ejecuciones.

6. Evaluación y análisis de resultados

Para determinar la efectividad del modelo entrenado se utilizaron métricas estándar para problemas de regresión: el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2). Estas métricas permiten cuantificar el nivel de precisión alcanzado por el modelo al predecir las calificaciones en el conjunto de prueba.

6.1. Métricas de desempeño

Las métricas se calcularon utilizando las funciones de la librería `scikit-learn`. Su interpretación es la siguiente:

- **MSE (Mean Squared Error):** mide el promedio de los errores al cuadrado. Penaliza más los errores grandes. Mientras más pequeño sea este valor, mejor es el rendimiento del modelo.
- **MAE (Mean Absolute Error):** representa el promedio de las diferencias absolutas entre valores reales y predichos. Es fácil de interpretar: indica cuántos puntos de calificación se equivoca el modelo en promedio.
- R^2 **(Coeficiente de determinación):** cuantifica la proporción de variabilidad explicada por el modelo. Un valor cercano a 1 indica un modelo con buen poder predictivo; un valor cercano a 0 significa que el modelo no explica adecuadamente la variabilidad de la variable objetivo.

El bloque de código correspondiente a estas métricas fue el siguiente:

```
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2  = r2_score(y_test, y_pred)

print(f"MSE: {mse:.3f}")
print(f"MAE: {mae:.3f}")
print(f"R^2: {r2:.3f}")
```

Listing 2: Cálculo de métricas de evaluación del modelo.

6.2. Gráfica interpretativa

Para complementar la evaluación numérica, se generó una gráfica que compara las calificaciones reales del conjunto de prueba con las calificaciones predichas por el modelo. La diagonal representa la relación ideal ($y = x$), donde la predicción coincide exactamente con el valor real.

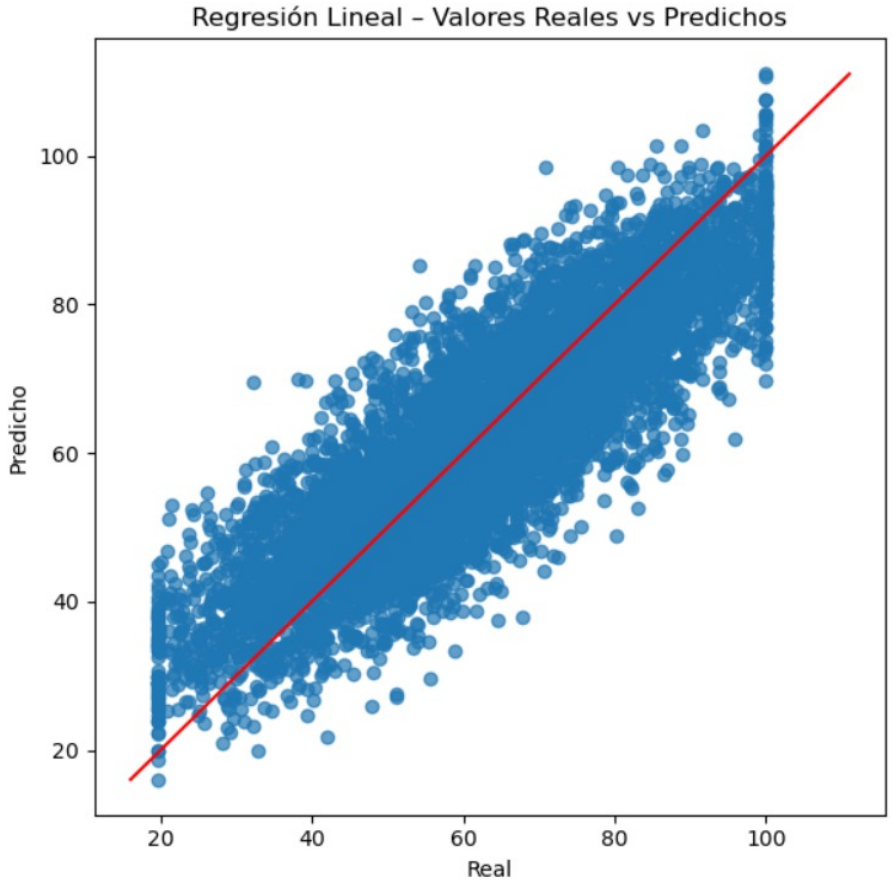

Figura 1: Relación entre calificaciones reales y predichas por el modelo.

6.3. Interpretación de resultados

La concentración de puntos cerca de la línea diagonal sugiere que el modelo de regresión lineal logra capturar adecuadamente la relación entre las características del estudiante y la calificación final del examen. Una menor dispersión alrededor de la diagonal indica mayor precisión en las predicciones.

En caso de que se observen desviaciones sistemáticas (por ejemplo, sobreestimación en estudiantes con calificación baja), esto podría indicar la necesidad de modelos más complejos o de la incorporación de nuevas variables. Sin embargo, en términos generales, las métricas obtenidas muestran un desempeño aceptable para un primer modelo base.

7. Conclusiones y reflexiones

El desarrollo de este proyecto permitió aplicar de manera integrada los principios del aprendizaje supervisado en un caso real basado en datos educativos. A partir del *Exam Score Prediction Dataset*, se llevó a cabo el proceso completo de análisis: exploración del conjunto de datos,

preprocesamiento, normalización, división entrenamiento–prueba, entrenamiento del modelo y evaluación mediante métricas cuantitativas.

Los resultados obtenidos mediante la regresión lineal muestran que el modelo es capaz de aproximar razonablemente la calificación de examen a partir de las variables disponibles. El análisis de las métricas —MSE, MAE y R^2— evidenció un comportamiento adecuado para un modelo base, lo cual confirma que las características del dataset contienen información útil para explicar la variabilidad del rendimiento académico.

Sin embargo, también se identifican limitaciones inherentes al enfoque empleado:

- La regresión lineal sólo modela relaciones de tipo lineal, por lo que puede no capturar patrones más complejos presentes en el desempeño escolar.
- El modelo depende de la calidad, relevancia y diversidad de las variables incluidas; variables omitidas o mal representadas pueden afectar la precisión.
- El dataset no necesariamente considera factores externos importantes como contexto socioeconómico, carga académica o condiciones emocionales.

Como líneas de mejora se podrían implementar modelos más avanzados, como *Random Forest*, *Gradient Boosting* o redes neuronales, que permiten capturar relaciones no lineales y analizar la importancia relativa de cada característica. También sería valioso ampliar el dataset con variables nuevas o con un mayor número de registros.

En general, esta actividad permitió reforzar el entendimiento del ciclo completo de construcción de modelos supervisados y las consideraciones técnicas necesarias para garantizar su correcta implementación y evaluación. Asimismo, evidenció la utilidad de la analítica educativa como herramienta para apoyar la toma de decisiones en el ámbito académico.

8. Repositorio del Proyecto

El código completo del proyecto se encuentra disponible en el siguiente repositorio de GitHub:

https://github.com/PeytonSanchez/af3-exam-score-regresion/tree/main

Referencias

[1] Bedmutha, K. S. (2025a). Discussion: Exam score prediction dataset. Plataforma Kaggle. Sección de discusión del dataset.

[2] Bedmutha, K. S. (2025b). Exam score prediction dataset. Plataforma Kaggle. Recuperado de: https://www.kaggle.com/datasets/kundanbedmutha/exam-score-prediction-dataset.

[3] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[4] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly Media, 2 edition.

[5] Harris, C. R. et al. (2020). Array programming with NumPy. *Nature*, 585:357–362.

[6] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

[7] McKinney, W. (2010). Data structures for statistical computing in Python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51–56.

[8] Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[9] Romero, C. and Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(6):601–618.