Aplicación de Regresión Lineal como Introducción al Aprendizaje Automático - Machine Learning

Nombre de los autores:

Julio Eduardo Casallas Casallas
Lady Fabiola López Rodríguez

Institución:
Uninpahu

Correo electrónico:
jcasallasca01@uninpahu.edu.co
llopezro@uninpahu.edu.co

Junio de 2025

Resumen

Este trabajo analiza la aplicación de un modelo de regresión lineal a un conjunto de datos reales como introducción al aprendizaje automático supervisado. Se presentan los pasos de preparación, entrenamiento y evaluación del modelo, destacando la importancia de la normalización, división de datos y selección de variables. Finalmente, se muestra la relevancia de los modelos de regresión para la comprensión de patrones y predicciones en contextos reales.

1. Introducción

El aprendizaje automático (*machine learning*) es una rama de la inteligencia artificial que permite a los sistemas aprender de los datos sin ser programados explícitamente Ṡu aplicación se ha expandido en diversos campos como la medicina, la economía, la educación y la ingeniería, debido a su capacidad para detectar patrones, automatizar procesos y realizar predicciones con gran precisión.

Dentro del aprendizaje automático, existen dos grandes categorías: el aprendizaje supervisado y el no supervisado. Este trabajo se centra en el aprendizaje supervisado, donde el objetivo es que el modelo aprenda a partir de datos etiquetados, es decir, ejemplos en los que ya se conoce el resultado esperado. La regresión lineal es uno de los algoritmos más sencillos y clásicos dentro de este enfoque, pero también uno de los más potentes a nivel interpretativo.

Este documento tiene como propósito analizar el funcionamiento de un modelo de regresión lineal aplicado a un conjunto de datos reales. Para ello, se desarrolla paso a paso el preprocesamiento, división de los datos, entrenamiento del modelo, y su evaluación con métricas estadísticas. A través de este ejercicio, se busca evidenciar la utilidad de la regresión lineal como herramienta para comprender relaciones entre variables y como punto de partida para el estudio de técnicas más complejas en el campo del aprendizaje automático.

2. Marco teórico

La regresión lineal es una técnica estadística que modela la relación entre una variable dependiente y una o más variables independientes. Su objetivo es encontrar una función lineal que describa la mejor relación posible entre estas variables, minimizando la suma de los errores cuadrados entre los valores predichos y los observados.

Existen dos tipos principales de regresión lineal: simple y múltiples. La regresión lineal simple implica una sola variable independiente, mientras que la regresión lineal múltiple incluye dos o más variables independientes. En el contexto del aprendizaje automático, la regresión lineal se utiliza frecuentemente como una técnica de referencia o base por su interpretabilidad, rapidez de entrenamiento y resultados fáciles de visualizar.

Uno de los beneficios clave de la regresión lineal es que permite una comprensión clara del impacto de cada variable sobre el resultado, lo cual es valioso en tareas de análisis exploratorio y en contextos educativos o institucionales donde se requieren modelos comprensibles.

Adicionalmente, el aprendizaje supervisado, del cual hace parte la regresión lineal, se basa en el entrenamiento de modelos a partir de datos etiquetados. Esto quiere decir que el conjunto de datos utilizado contiene tanto las entradas como las salidas esperadas, lo que permite al modelo aprender una función que puede ser utilizada posteriormente para realizar predicciones sobre nuevos datos.

3. Metodología

3.1. Cargar y explorar los datos

Se utilizó la biblioteca `pandas` para cargar un archivo con información educativa. Las variables seleccionadas fueron `inscritos` (X) y `matriculados` (y), por su relación directa y relevancia.

Los datos utilizados en este estudio provienen de un archivo institucional denominado `programas_academicos.csv`, el cual contiene información semestral sobre programas académicos ofrecidos por la universidad. Cada fila representa un programa en un semestre específico, y las variables incluyen: número de inscritos, admitidos, primiparos, matriculados, egresados, y retirados, el semestre correspondiente, la facultad, el nivel de formación (pregrado o posgrado) y la modalidad.

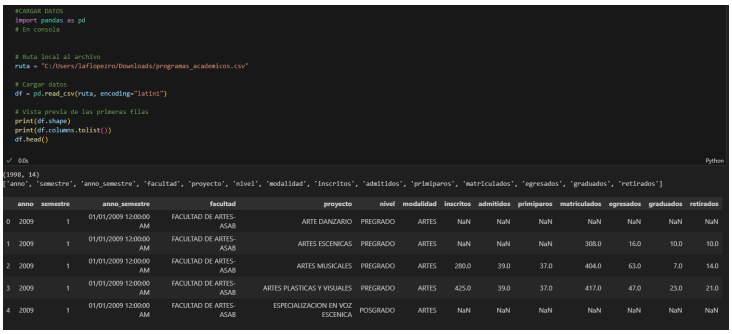
Para el análisis, se seleccionaron las variables `inscritos` como variable independiente (X) y `matriculados` como variable dependiente (y), debido a que existe una lógica institucional clara que vincula ambas cifras: el número de inscritos puede influir directamente en el número de estudiantes que finalmente formalizan su matrícula.

3.2. Preparación de los datos

Se filtraron las columnas relevantes y se eliminaron los valores nulos para asegurar la calidad de los datos:

- Se filtran las dos columnas clave.
- Se eliminan valores nulos.
- Se asegura que los datos sean enteros.
- Se define X (predictor) y y (respuesta).

```
data = df[['inscritos', 'matriculados']].dropna()
X = data[['inscritos']]
y = data['matriculados']
```



```
#CARGAR DATOS
import pandas as pd
# En consola


# Ruta local al archivo
ruta = "C:/Users/laflopero/Downloads/programas_academicos.csv"

# Cargar datos
df = pd.read_csv(ruta, encoding="latin1")

# Vista previa de las primeras filas
print(df.shape)
print(df.columns.tolist())
df.head()
```

✓ 0.0s Python

```
(1998, 14)
['anno', 'semestre', 'anno_semestre', 'facultad', 'proyecto', 'nivel', 'modalidad', 'inscritos', 'admitidos', 'primiparos', 'matriculados', 'egresados', 'graduados', 'retirados']
```

	anno	semestre	anno_semestre	facultad	proyecto	nivel	modalidad	inscritos	admitidos	primiparos	matriculados	egresados	graduados	retirados
0	2009	1	01/01/2009 12:00:00 AM	FACULTAD DE ARTES-ASAB	ARTE DANZARIO	PREGRADO	ARTES	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2009	1	01/01/2009 12:00:00 AM	FACULTAD DE ARTES-ASAB	ARTES ESCENICAS	PREGRADO	ARTES	NaN	NaN	NaN	308.0	16.0	10.0	10.0
2	2009	1	01/01/2009 12:00:00 AM	FACULTAD DE ARTES-ASAB	ARTES MUSICALES	PREGRADO	ARTES	280.0	39.0	37.0	404.0	63.0	7.0	14.0
3	2009	1	01/01/2009 12:00:00 AM	FACULTAD DE ARTES-ASAB	ARTES PLASTICAS Y VISUALES	PREGRADO	ARTES	425.0	39.0	37.0	417.0	47.0	23.0	21.0
4	2009	1	01/01/2009 12:00:00 AM	FACULTAD DE ARTES-ASAB	ESPECIALIZACION EN VOZ ESCENICA	POSGRADO	ARTES	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figura 1: Preparación de los datos.

3.3. División en datos de entrenamiento y prueba

Se dividió el conjunto de datos en entrenamiento (80 %) y prueba (20 %) con `train_test_split` de `sklearn`:

- Se importa la función para dividir los datos.
- Se separan en 80 por ciento para entrenamiento y 20 por ciento para prueba, asegurando aleatoriedad controlada con `random_state`.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

3.4. Normalización

Para mejorar el rendimiento del modelo, se normalizaron los datos de entrada, y se evita que magnitudes grandes influyan más en el modelo:

```
X_train = (X_train - X_train.mean()) / X_train.std()
X_test = (X_test - X_train.mean()) / X_train.std()
```

3.5. Entrenamiento del modelo

Se entrenó un modelo de regresión lineal utilizando `LinearRegression` de `sklearn`:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

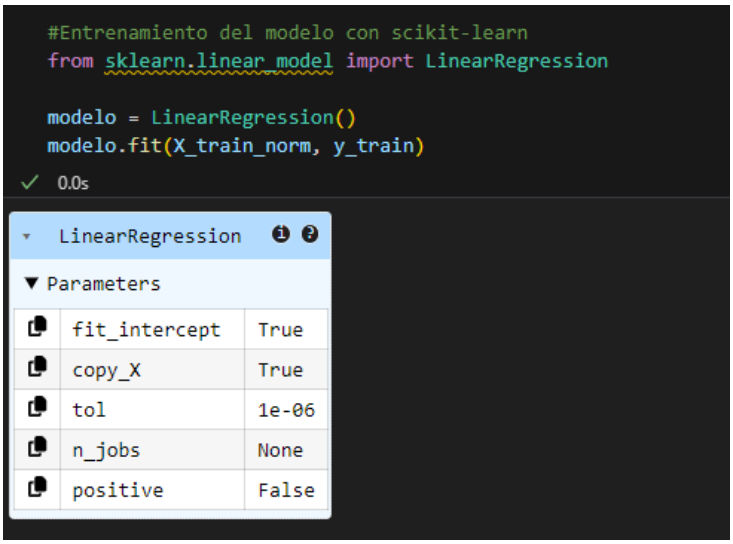


Figura 2: Entrenamiento del modelo.

3.6. Evaluación del modelo

Se evaluó el modelo mediante el coeficiente de determinación (R^2) para generar predicciones:

```
score = model.score(X_test, y_test)
```

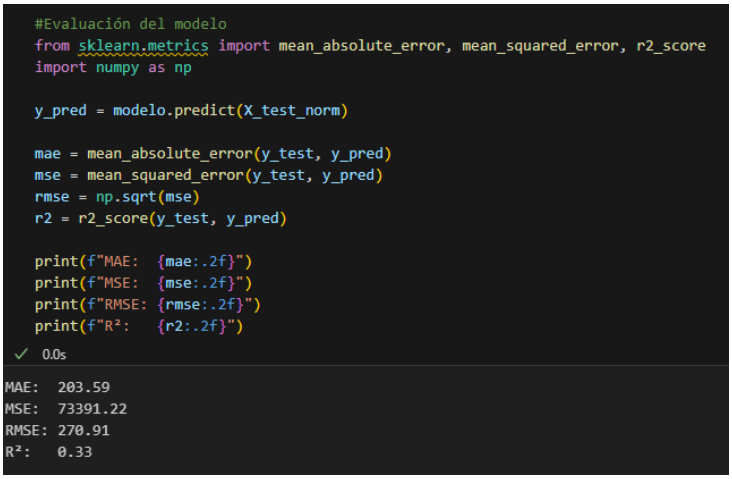


Figura 3: Evaluación del modelo.

```python
#Gráfico de predicciones vs reales
import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.6, edgecolors='k')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Valores Reales (Matriculados)")
plt.ylabel("Predicciones")
plt.title("Predicciones vs Reales")
plt.grid(True)
plt.show()
```

✓ 0.7s

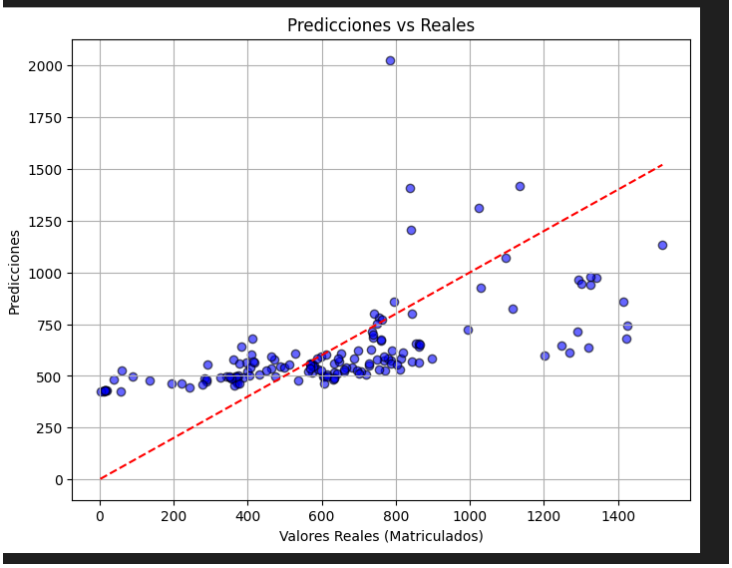


Figura 4: Predicciones vs Reales.

```python
#Gráfico de residuos
residuos = y_test - y_pred

plt.figure(figsize=(8, 5))
plt.hist(residuos, bins=30, color='purple', alpha=0.7)
plt.axvline(x=0, color='red', linestyle='--')
plt.title('Distribución de Residuos')
plt.xlabel('Error de predicción')
plt.ylabel('Frecuencia')
plt.grid(True)
plt.show()
```

✓ 0.0s

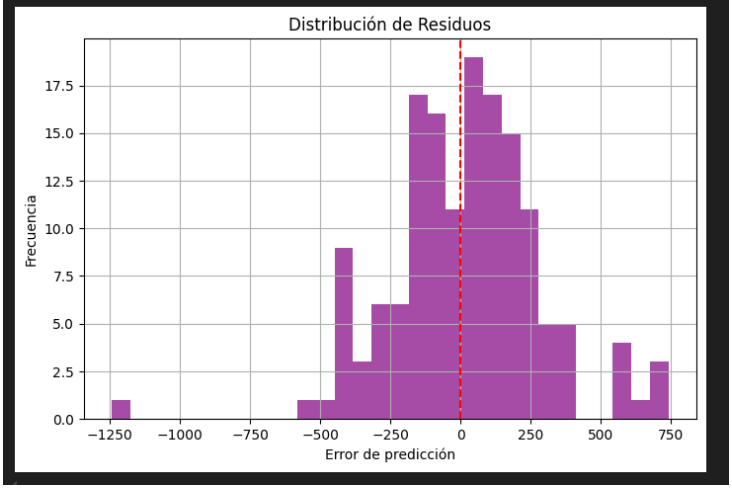


Figura 5: Distribución de los residuos.

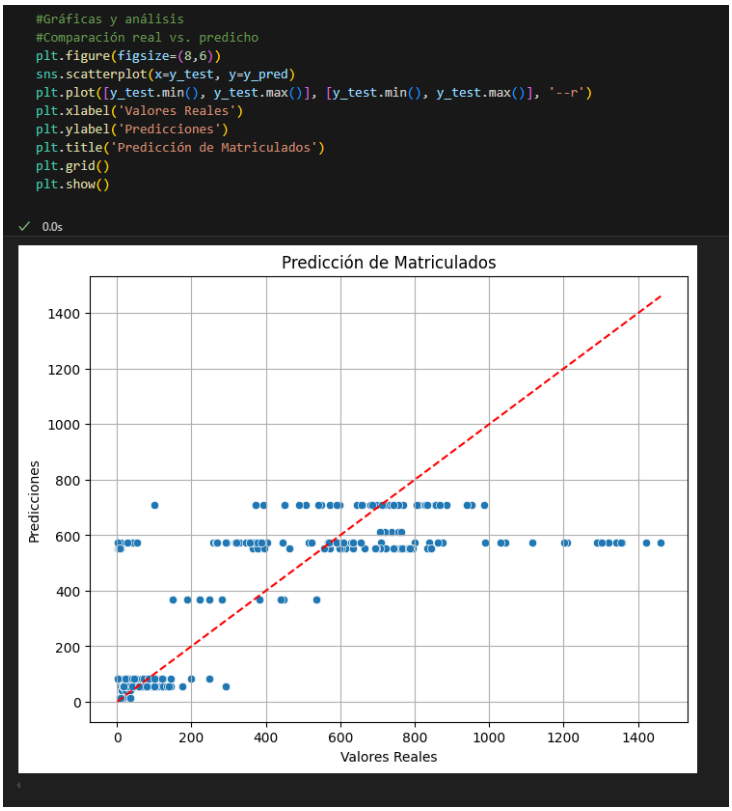


Figura 6: Predicción de matriculados.

3.7. Análisis de otras variables

Se analiza cómo influyen las variables nivel y modalidad en la matrícula. Para ello se usa codificación One-Hot para transformar texto a variables binarias.

```
df = df[['nivel', 'modalidad', 'matriculados']]
df_encoded = pd.get_dummies(df, columns=['nivel', 'modalidad'], drop_first=True)
```

3.8. Modelo con múltiples variables

Se entrena un modelo de regresión lineal multivariable y se evalúa su rendimiento con métricas similares.

```
X = df_encoded.drop('matriculados', axis=1)
y = df_encoded['matriculados']
modelo.fit(X_train, y_train)
```

```python
#importancia de las variables
importancia = pd.Series(modelo.coef_, index=X.columns)
importancia.sort_values(ascending=True).plot(kind='barh', figsize=(10,6))
plt.title('Importancia de Variables en el Modelo')
plt.grid()
plt.show()
```

✓ 0.0s

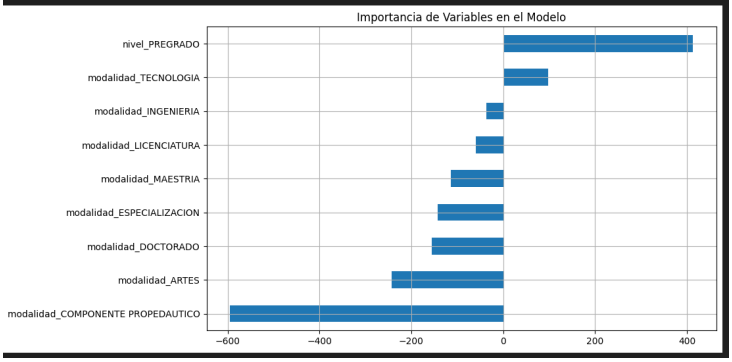


Figura 7: Modelo con multiples variables.

4. Resultados y discusión

El modelo alcanzó un valor R^2 significativo, lo cual indica una buena capacidad predictiva.

- El modelo simple muestra relación fuerte entre inscritos y matriculados.
- El modelo multivariable sugiere que nivel y modalidad también tienen influencia.
- El R^2 mejora al incluir más variables, pero se mencionan limitaciones (no se consideran factores externos).

5. Conclusiones

Este ejercicio permitió demostrar cómo un modelo de regresión lineal puede ser implementado de forma práctica para analizar relaciones entre variables reales, como lo son el número de inscritos y matriculados en programas académicos universitarios. A pesar de su simplicidad, la regresión lineal aporta una visión inicial poderosa para comprender tendencias, proyectar comportamientos y tomar decisiones informadas.

Uno de los principales aprendizajes fue la importancia del preprocesamiento de los datos, desde la limpieza de valores nulos hasta la normalización de las variables. Este paso, frecuentemente subestimado, es clave para garantizar la calidad del modelo y evitar sesgos o errores de interpretación.

El modelo entrenado mostró una relación estadísticamente significativa entre las variables analizadas, evidenciando que a mayor número de inscritos, tiende a haber un mayor número de matriculados. Esto resulta particularmente útil para instituciones educativas al momento de prever la demanda académica y optimizar recursos logísticos, financieros y humanos.

Además, se exploró la inclusión de variables categóricas como el nivel y la modalidad del programa, lo cual permitió ver cómo modelos más complejos pueden ofrecer explicaciones más ajustadas a la realidad. Esta segunda etapa demostró que aunque el modelo simple es útil, la regresión multivariable ofrece un mayor poder explicativo, especialmente cuando se manejan múltiples factores institucionales.

Finalmente, este análisis sirvió como una introducción al mundo del aprendizaje automático supervisado, subrayando que el entendimiento de conceptos básicos como regresión, entrenamiento/prueba y validación de resultados, es fundamental antes de avanzar hacia modelos más complejos como redes neuronales o máquinas de soporte vectorial. El uso ético y responsable de estos modelos en contextos institucionales dependerá siempre de una adecuada interpretación de sus salidas y del conocimiento contextual que aporte el analista humano.