Instituto Tecnológico y de Estudios Superiores de Monterrey
Inteligencia artificial avanzada para la ciencia de datos I

Predicción de demanda de bicicletas mediante regresión lineal implementada sin framework

Profesor:
Benjamín Valdés Aguirre

Paul Park - A01709885

31 de agosto de 2025

Introducción

Los sistemas de bicicletas compartidas facilitan la renta y devolución de bicicletas alrededor del mundo. Esta tecnología ha generado un gran interés por su importancia en áreas de tráfico, ambientalismo y salud. Además, dado que los datos de uso (como el tiempo de renta y las estaciones de origen y destino) están explícitamente registrados, estos sistemas ofrecen mayores oportunidades para investigaciones comparado con otros transportes, como autobuses o metro.

El objetivo de este proyecto es implementar regresión lineal desde cero, sin usar frameworks de Machine Learning, para predecir la cantidad total de bicicletas rentadas. Esta implementación permite comprender los fundamentos del aprendizaje automático, el ajuste de parámetros y la optimización mediante gradiente descendente.

Descripción del dataset

Se utilizó el dataset Bike Sharing del UC Irvine Machine Learning Repository, que contiene registros diarios y horarios de 2011 a 2012 del sistema Capital Bikeshare en Washington D.C. Por cuestiones de simplicidad (el fin es evaluar la implementación manual del algoritmo), se eligió el conjunto de registros diarios, dado que contiene 731 filas, a diferencia del de registros horarios con 17379 filas. Las variables seleccionadas y su tratamiento son:

Variable	Tipo	Descripción	Uso
season	categórica	Estación del año (1: primavera, 2: verano, 3: otoño, 4: invierno)	Usada (one-hot encoding)
yr	categórica	Año del registro (0: 2011, 1: 2012)	Usada (one-hot)
mnth	categórica	Mes del registro (1-12)	Usada (one-hot)
weekday	categórica	Día de la semana (0-6)	Usada (one-hot)
weathersit	categórica	Tipo de clima (1: soleado, 2: nublado, 3: lluvia ligera, 4: tormenta/nieve)	Usada (one-hot)
temp	continua	Temperatura normalizada (°C)	Usada (normalizada)
atemp	continua	Sensación térmica normalizada (°C)	Usada (normalizada)
hum	continua	Humedad normalizada (%)	Usada (normalizada)
windspeed	continua	Velocidad del viento normalizada	Usada (normalizada)

casual	integer	Número de usuarios casuales	Descartada
registered	integer	Número de usuarios registrados	Descartada
instant	integer	Índice del registro	Descartada
dteday	date	Fecha del registro	Descartada
cnt	integer	Objetivo: Total de bicicletas rentadas	Usada

El preprocesamiento realizado fue el siguiente: Primero, se utilizó una codificación one-hot manual para todas las variables categóricas (season, yr, mnth, weekday, weathersit, asignando los números de columna especificados en la tabla), descartando la primera categoría para evitar multicolinealidad (es decir, queda implícita en el algoritmo cuando todas las demás tienen un valor de 0). Después, se normalizaron las variables continuas (temp, atemp, hum, windspeed) manualmente para que estén en un rango similar y evitar que el algoritmo se confunda, usando la fórmula:

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)}$$

Luego, se descartaron las variables irrelevantes para el análisis (casual, registered, instant, dteday) y se dividió el dataset en conjuntos de entrenamiento (train) y de prueba (test) con una proporción de 80/20.

Metodología

Se implementó regresión lineal con las siguientes características:

- **Hipótesis:**

$$\widehat{y} = X \cdot \omega$$

donde $\widehat{y}$ son las predicciones del modelo, X son los datos de entrada (con un término de bias) y ω son los pesos a aprender.

- **Función de costo:** Error cuadrático medio (MSE)

$$J(\omega) = \frac{1}{m}\sum_{i=1}^{m} (\widehat{y}_i - y_i)^2$$

donde m es el número de datos (ejemplos) y_i es el valor real, y $\widehat{y}_i$ es la predicción del modelo.

- **Optimización:** Gradiente descendente

$$\omega := \omega - \alpha \nabla J(\omega)$$

donde $\alpha \nabla J(\omega)$ es el gradiente (pendiente de la función de costo respecto a los pesos) $\alpha = 0.01$ es la tasa de aprendizaje y epochs = 1000 es el número de veces que el algoritmo recorre todos los datos ajustando los pesos.

- **Métricas:** Se calculó el R^2 para evaluar el ajuste del modelo, dado que mide qué porcentaje de la variabilidad de los datos reales explica el modelo (entre 0 y 1).

Implementación

Dado que el requisito principal de este trabajo es implementar una técnica de Machine Learning sin frameworks, las únicas librerías que se utilizaron para el código fueron numpy y pandas (para la manipulación de datos), así como matplotlib para graficar los datos. La implementación se puede dividir en las siguientes funciones principales:

- **hipotesis(X, w):** calcula $\hat{y} = X \cdot \omega$
- **costo(X, y, w):** calcula MSE
- **gradiente(X, y, w):** calcula gradiente para actualización de pesos
- **r2_score(y_true, y_pred):** calcula coeficiente de determinación R^2

Todas estas funciones se encuentran en un archivo main.py.

Resultados

El output del código fue el siguiente:

```
Epoch 0, Cost: 21084212.16
Epoch 100, Cost: 1617811.20
Epoch 200, Cost: 1199135.68
Epoch 300, Cost: 1006510.60
Epoch 400, Cost: 904035.51
Epoch 500, Cost: 841674.29
Epoch 600, Cost: 799330.86
Epoch 700, Cost: 768156.88
Epoch 800, Cost: 743872.45
Epoch 900, Cost: 724209.83
```

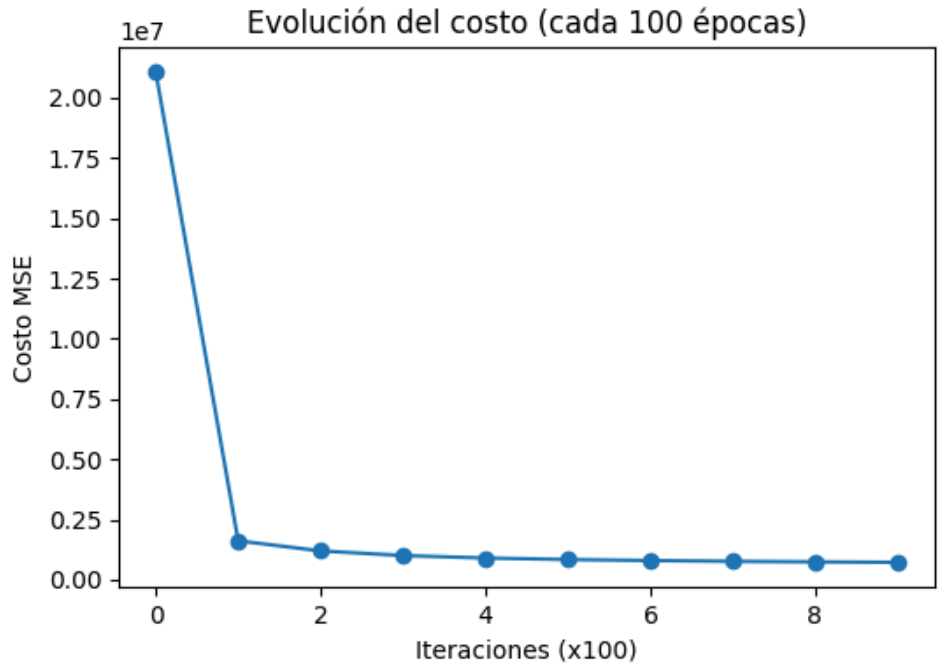
```
Primeras 10 predicciones vs reales:
Real: 5424, Predicción: 6508.17
Real: 8714, Predicción: 6932.98
Real: 5976, Predicción: 6482.76
Real: 5976, Predicción: 6482.76
Real: 6290, Predicción: 6132.06
Real: 7216, Predicción: 6691.94
Real: 1693, Predicción: 1928.89
Real: 5058, Predicción: 4708.56
Real: 3915, Predicción: 4932.80
Real: 7736, Predicción: 7072.22
Real: 1005, Predicción: 1148.29

R² en test: 0.8470
```

El R^2 fue de 0.8470 en test, indicando que ~84% de la varianza de los datos es explicada por el modelo.

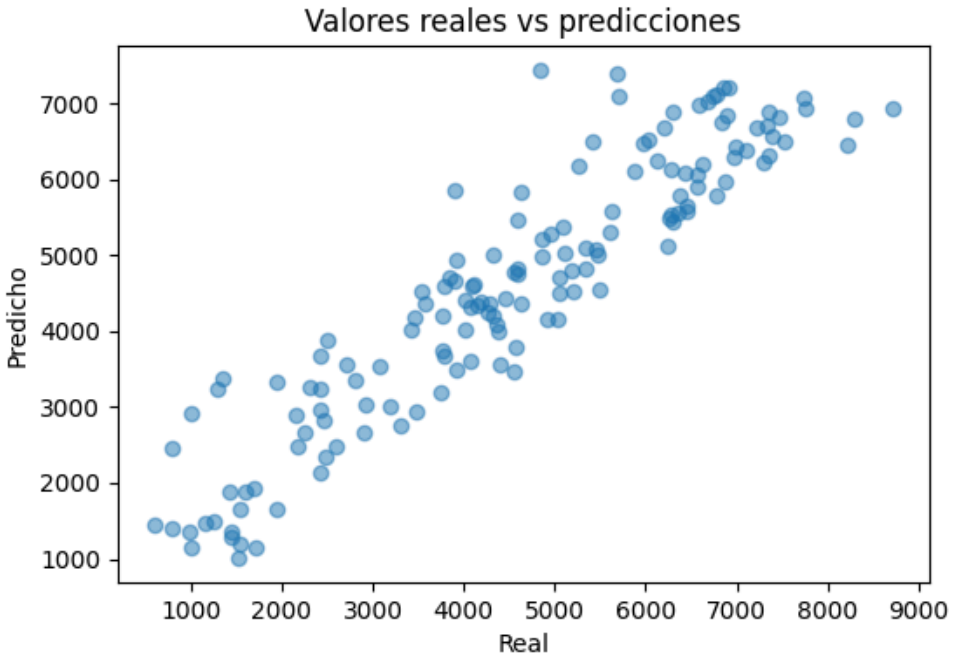
También se creó un folder “Results”, en donde se almacenaron gráficas correspondientes a la curva de costo (training_curve.png) y las predicciones vs realidad (predictions.png):

training_curve.png



Aquí se puede observar una disminución consistente del MSE cada 100 épocas, indicando convergencia del gradiente descendente.

predictions.png



La dispersión es cercana a la diagonal, demostrando que el modelo predice de manera razonable.

Conclusiones

La regresión lineal implementada aprendió del dataset correctamente sin frameworks y las predicciones son razonablemente precisas considerando la simplicidad del modelo. Sin embargo, cabe destacar que no se aplicó regularización, lo que podría mejorar la generalización. Además, algunas variables pueden tener relaciones no lineales que este modelo lineal no captura. Si se consideraran mejoras para este proyecto, se podría probar regresión polinómica, regularización L1/L2 u otros algoritmos manuales.

Referencias

UC Irvine Machine Learning Repository – Bike Sharing Dataset.
https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset