ITSRLL
INSTITUTO TECNOLÓGICO SUPERIOR DE LA REGIÓN DE LOS LLANOS

Ingeniería Mecatrónica

PROGRAMACIÓN AVANZADA

Enero – Junio 2025

M.C. Osbaldo Aragón Banderas

UNIDAD: 2

Actividad número: A4

Nombre de actividad:

NOOTEBOOK: Análisis de Datos Aplicables a Regresión Lineal Simple

Actividad realizada por:

Roberto Jair Arteaga Valenzuela

Guadalupe Victoria, Durango

Fecha de entrega: 09 de marzo de 2025

NOOTEBOOK: Análisis de Datos Aplicables a Regresión Lineal Simple

Introducción

La regresión lineal es una de las técnicas más simples y ampliamente utilizadas en el análisis de datos, especialmente cuando se busca modelar la relación entre una variable dependiente continua y una o más variables independientes. Esta práctica tiene como objetivo aplicar el método de regresión lineal sobre un conjunto de datos real, para predecir un valor continuo a partir de varias características.

En esta práctica, se utilizará el dataset de California Housing, que contiene información sobre diferentes características de viviendas en California, como el ingreso medio de los habitantes, la edad de las viviendas, la proximidad a centros comerciales y la población de cada área. El objetivo es predecir el precio medio de las viviendas utilizando estos atributos. Este tipo de problema se considera una tarea de regresión, ya que la variable a predecir (el precio de las viviendas) es un valor continuo.

El dataset de California Housing se encuentra disponible en la librería scikit-learn y contiene 20,640 instancias con 8 características, lo que lo hace adecuado para aplicar modelos de regresión lineal.

Investigación

La regresión lineal simple es un modelo estadístico utilizado para describir la relación entre dos variables: una dependiente (que es la que intentamos predecir o explicar) y una independiente (que es la que usamos para hacer la predicción). En términos sencillos, la regresión lineal simple busca encontrar una línea recta que mejor se ajuste a los datos, de forma que podamos predecir el valor de la variable dependiente en función de la variable independiente.

Aplicación en problemas de predicción: La regresión lineal simple es muy útil cuando necesitamos predecir un valor continúo basándonos en una sola variable independiente. Por ejemplo, podríamos usar regresión lineal simple para predecir el precio de una vivienda basándonos en su tamaño (en metros cuadrados). Otro ejemplo común es predecir las ventas de una tienda en función del presupuesto de marketing.

La idea principal es que existe una relación lineal entre las dos variables, es decir, a medida que cambia la variable independiente (X), la variable dependiente (Y) también cambia, y esta relación puede ser modelada como una línea recta.

Ecuación Matemática de la Regresión Lineal Simple:

La ecuación general de una regresión lineal simple es la siguiente:

$$Y=\beta 0+\beta 1X+\epsilon$$

Donde:

- Y es la variable dependiente (lo que intentamos predecir).
- X es la variable independiente (lo que utilizamos para hacer la predicción).
- β_0 es el intercepto (el valor de Y cuando X es 0, es decir, el punto donde la línea corta el eje Y).
- β_1 es el coeficiente de regresión (la pendiente de la línea, que indica cómo cambia Y cuando X cambia en una unidad).

- ε es el error o término de perturbación, que captura las diferencias entre los valores reales de Y y los valores predichos por el modelo.
- ei=Yi−(β0+β1Xi)

El objetivo de la regresión lineal es encontrar los valores de β_0 (intercepto) y β_1 (pendiente) que minimicen el error y ajusten la mejor línea a los datos.

Determinación de la Mejor Línea de Ajuste mediante Mínimos Cuadrados:

Para determinar la mejor línea de ajuste, se utiliza el método de mínimos cuadrados, que tiene como objetivo minimizar la suma de los errores al cuadrado. El error para cada punto de datos es la diferencia entre el valor real de Y y el valor predicho por la línea de regresión. La fórmula del error para cada punto (Xi,Yi)(Xi,Yi) es:

$$e_i = Y_i - (\beta_0 + \beta_1 X_i)$$

- Donde: ei es el error para el i-ésimo punto de datos.
- Yi es el valor real de la variable dependiente para el i-ésimo punto.
- β0+β1Xi es el valor predicho por la línea de regresión para ese punto.

El **error cuadrático** es simplemente el cuadrado de esta diferencia:

$$e_i^2 = (Y_i - \hat{Y}_i)^2$$

Donde $\hat{Y}_i$ es el valor predicho por el modelo para el i-ésimo valor de X.

El **criterio de mínimos cuadrados** consiste en minimizar la **suma de los errores cuadrados** (SSE, por sus siglas en inglés **Sum of Squared Errors**):

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

El objetivo es encontrar los valores de $\boldsymbol{\beta_0}$ y $\boldsymbol{\beta_1}$ que minimicen esta suma. Para hacerlo, se calcula la derivada de SSE con respecto a $\boldsymbol{\beta_0}$ y $\boldsymbol{\beta_1}$, y luego se igualan a cero para obtener las fórmulas de los **coeficientes óptimos**.

Las fórmulas para $\boldsymbol{\beta_0}$ y $\boldsymbol{\beta_1}$ son:

$$\beta_1 = \frac{n\sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{n\sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2}$$

$$\beta_0 = \frac{\sum_{i=1}^{n} Y_i - \beta_1 \sum_{i=1}^{n} X_i}{n}$$

Una vez que tenemos los valores óptimos de $\boldsymbol{\beta_0}$ y $\boldsymbol{\beta_1}$, podemos usar la ecuación de la recta $Y=\beta 0+\beta 1X$ $Y=\beta 0+\beta 1X$ para hacer predicciones sobre los valores de Y, dada cualquier nueva entrada de X.

1. **Importación de librerías**:
    - Se importan librerías como pandas, numpy, matplotlib, y funciones de sklearn para manejo de datos, creación y evaluación de modelos.

```
# Paso 1: Importar las librerías necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

2. **Carga del dataset**:
    - Se carga el conjunto de datos **California Housing** usando la función fetch_california_housing de sklearn. Este dataset contiene información sobre características de las viviendas y su precio medio.

```
# Paso 2: Cargar el Dataset
california_data = fetch_california_housing()
```

3. **Visualización de los datos**:
    - Se crea un DataFrame para organizar las características de las viviendas (X) y un Series para los precios (Y). Posteriormente, se muestran las primeras filas del dataset.

```
# Crear un DataFrame para visualizar los datos
X = pd.DataFrame(california_data.data, columns=california_data.feature_names)
Y = pd.Series(california_data.target, name='Price')
```

```
# Visualizar las primeras filas del dataset
print("Primeras filas del dataset:")
print(X.head())
```

```
Primeras filas del dataset:
   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
0  8.3252      41.0  6.984127   1.023810       322.0  2.555556     37.88
1  8.3014      21.0  6.238137   0.971880      2401.0  2.109842     37.86
2  7.2574      52.0  8.288136   1.073446       496.0  2.802260     37.85
3  5.6431      52.0  5.817352   1.073059       558.0  2.547945     37.85
4  3.8462      52.0  6.281853   1.081081       565.0  2.181467     37.85

   Longitude
0    -122.23
1    -122.22
2    -122.24
3    -122.25
4    -122.25
```

4. **División de datos**:
    - Se divide el dataset en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%) usando train_test_split.

```
# Paso 3: Dividir el dataset en conjunto de entrenamiento y conjunto de prueba
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

5. **Creación y entrenamiento del modelo**:
    - Se crea un modelo de regresión lineal utilizando LinearRegression(). Luego, el modelo se entrena con el conjunto de entrenamiento (X_train y Y_train).

```
# Paso 4: Crear el modelo de regresión lineal
model = LinearRegression()
```

```
# Paso 5: Entrenar el modelo
model.fit(X_train, Y_train)
```

6. **Predicción sobre el conjunto de prueba**:
    - El modelo entrenado realiza predicciones sobre el conjunto de prueba (X_test).

```
# Paso 6: Realizar predicciones sobre el conjunto de prueba
Y_pred = model.predict(X_test)
```

7. **Evaluación del modelo**:
    - Se calcula el **Error Cuadrático Medio (MSE)** y el **Coeficiente de Determinación R^2** para evaluar la precisión del modelo.
        - **MSE** mide la diferencia promedio entre los valores reales y las predicciones. Un valor más bajo indica mejor precisión.
        - **R^2** mide la cantidad de variación en los precios de las viviendas que es explicada por el modelo. Un valor cercano a 1 indica un buen ajuste del modelo a los datos.

```
# Paso 7: Evaluar el modelo
mse = mean_squared_error(Y_test, Y_pred)  # Error cuadrático medio
r2 = r2_score(Y_test, Y_pred)  # Coeficiente de determinación R^2
```

8. **Visualización de resultados**:
    - Se crea un gráfico de dispersión que compara los precios reales (Y_test) con los precios predichos (Y_pred). La línea roja indica la relación ideal entre ambos (donde los valores reales y predichos coinciden).

```
print(f"\nError cuadrático medio (MSE): {mse}")
print(f"Coeficiente de determinación R^2: {r2}")
```

```
Error cuadrático medio (MSE): 0.5558915986952437
Coeficiente de determinación R^2: 0.5757877060324512
```

9. **Mostrar coeficientes del modelo**:
    - Se muestran los coeficientes de cada una de las características (columnas) del modelo de regresión, lo que indica cuánto influye cada variable en la predicción del precio.

```
# Paso 8: Visualizar los resultados
plt.figure(figsize=(10, 6))
```

```
<Figure size 1000x600 with 0 Axes>
<Figure size 1000x600 with 0 Axes>
```

```
# Visualizar los precios reales vs predicciones
plt.scatter(Y_test, Y_pred, color='blue')
plt.plot([min(Y_test), max(Y_test)], [min(Y_test), max(Y_test)], color='red', linestyle='--')
plt.title('Precios reales vs Precios predichos')
plt.xlabel('Precio real')
plt.ylabel('Precio predicho')
plt.show()
```

Resultados:

- **Error Cuadrático Medio (MSE)**: Este valor cuantifica qué tan lejos están, en promedio, las predicciones de los precios reales. Un valor más bajo es mejor.
- **Coeficiente de determinación R^2**: Este valor indica qué tan bien el modelo explica la variación en los precios de las viviendas. Un valor de 1 significaría que el modelo explica toda la variación, mientras que un valor cercano a 0 indicaría un ajuste pobre.
- **Gráfico de dispersión**: Ayuda a visualizar la relación entre los precios reales y los predichos. Cuanto más se acerque la dispersión de puntos a la línea roja, mejor será el modelo.
- **Coeficientes del modelo**: Muestran el comportamiento de cada característica en la predicción del precio. Por ejemplo, un coeficiente alto en una variable implica que esa característica tiene una gran influencia en la estimación del precio.

1. Análisis del Dataset:

El conjunto de datos California Housing tiene información sobre varias características de las viviendas en California, como el número de habitaciones, la proximidad a las áreas costeras, y el ingreso medio del vecindario, entre otros. La variable dependiente o target es el precio medio de las viviendas en cada área.

Las características del dataset se almacenan en X, mientras que los precios en Y (que son los precios medios de las viviendas en cada localidad).

Vista previa de los datos:

Cuando ejecutamos el código para ver las primeras filas del dataset (X.head()), podemos obtener información sobre las variables involucradas:

- AveRooms: Promedio de habitaciones por vivienda.
- AveOccup: Promedio de personas por vivienda.
- MedInc: Ingreso medio del vecindario.

- HouseAge: Edad media de las viviendas en el vecindario.
- MedHouseVal: El precio medio de las viviendas (esta es la variable que queremos predecir).
- Entre otras variables relacionadas con la ubicación geográfica de las viviendas.

2. División de los Datos:

El dataset se divide en dos partes:

- **Conjunto de entrenamiento** (80% de los datos): Este conjunto se utiliza para entrenar el modelo, ajustando los parámetros internos del modelo.
- **Conjunto de prueba** (20% de los datos): Este conjunto se utiliza para evaluar el modelo una vez entrenado, con el fin de verificar si el modelo tiene una buena capacidad de generalización.

3. Modelo de Regresión Lineal:

El modelo creado es de tipo regresión lineal, que busca encontrar una relación lineal entre las características (variables independientes) y la variable objetivo (precio de la vivienda). El modelo ajusta una recta (o plano en el caso multivariable) para minimizar la suma de los errores cuadráticos, lo cual se evalúa usando el Error Cuadrático Medio (MSE).

Evaluación del Modelo:

Error Cuadrático Medio (MSE):
El MSE es una métrica de evaluación que mide la diferencia cuadrática entre los valores reales y las predicciones. El valor de MSE se obtiene con la función mean_squared_error(Y_test, Y_pred) y tiene unidades al cuadrado del objetivo, lo que hace que su interpretación sea compleja. Un MSE más bajo indica que las predicciones están más cerca de los valores reales.

Ejemplo:

- Si el MSE es 0.5, esto indica que, en promedio, las predicciones del modelo tienen un error cuadrático de 0.5 unidades respecto al valor real.

Coeficiente de Determinación R^2:
El valor **R^2** es otro indicador de qué tan bien el modelo ajusta los datos. Se calcula con r2_score(Y_test, Y_pred), y va de 0 a 1. Un valor de 1 significa que el modelo explica perfectamente la variabilidad en los datos, mientras que un valor de 0 indica que el modelo no explica nada de la variabilidad en los datos.

- **R^2 cercano a 1**: El modelo explica casi toda la variación del precio de las viviendas en el conjunto de prueba.
- **R^2 cercano a 0**: El modelo tiene un ajuste deficiente y no puede predecir bien los precios de las viviendas.

4. Visualización de Resultados:

Se genera un gráfico de dispersión donde se comparan los **precios reales** (Y_test) contra los **precios predichos** (Y_pred). En este gráfico:

- La **línea roja** es la línea de "perfecta predicción", donde los valores reales coinciden exactamente con las predicciones. Los puntos cercanos a esta línea indican buenas predicciones.
- Si los puntos se alejan significativamente de la línea roja, esto sugiere que el modelo no está funcionando bien en esos casos específicos.

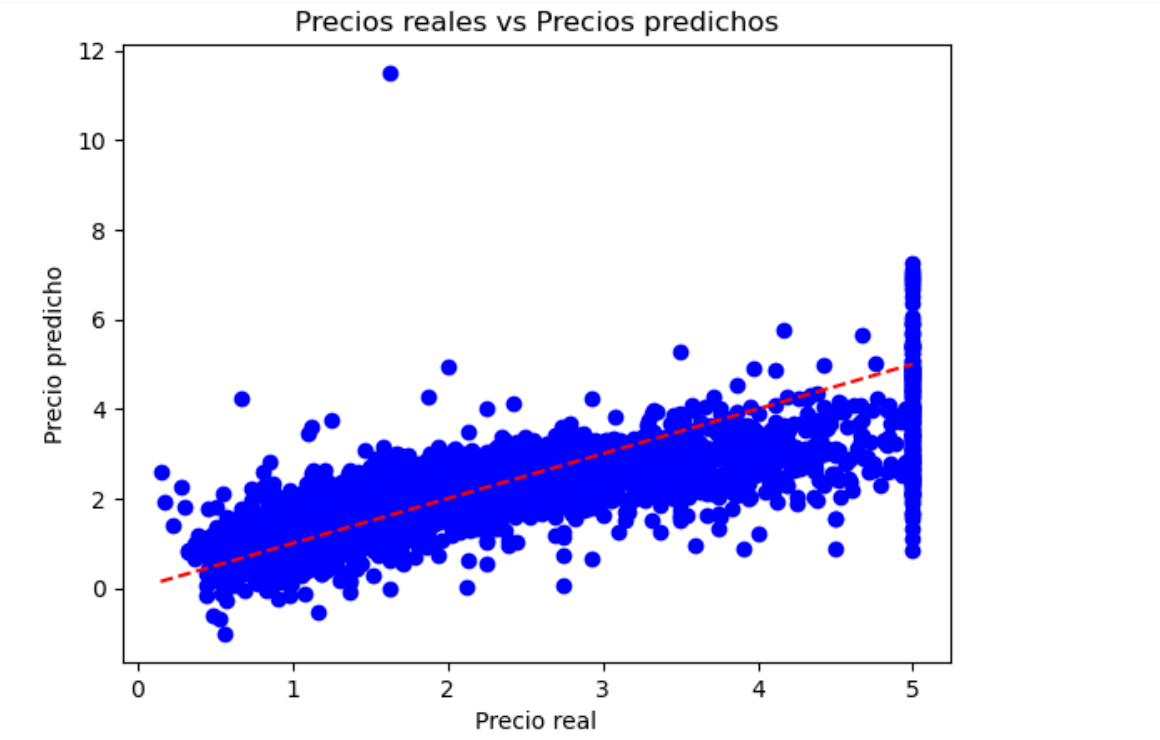

Figura 1 Grafica de dispersión

Interpretación del gráfico:

- Si la mayoría de los puntos están cerca de la línea roja, esto indica que el modelo tiene un buen rendimiento.
- Si los puntos están dispersos sin seguir una tendencia clara, esto sugiere que el modelo tiene un ajuste pobre.

5. Coeficientes del Modelo:

El modelo de regresión lineal calcula un **coeficiente** para cada variable independiente. Estos coeficientes indican cuánto cambia el precio de la vivienda por cada unidad de cambio en la variable correspondiente, manteniendo las otras variables constantes.

Los coeficientes se muestran con el siguiente bloque de código:

```
for feature, coef in zip(X.columns, model.coef_):
    print(f"{feature}: {coef}")
```

Interpretación de los coeficientes:

- Si un coeficiente es **positivo**: Significa que a medida que esa característica aumenta, también lo hace el precio de las viviendas.
- Si un coeficiente es **negativo**: Significa que a medida que esa característica aumenta, el precio de las viviendas disminuye.

Por ejemplo:

- Si el coeficiente de AveRooms (número promedio de habitaciones) es positivo, esto indicaría que, en general, más habitaciones por vivienda aumentan el precio de la vivienda.
- Si el coeficiente de HouseAge (edad de la vivienda) es negativo, podría indicar que las viviendas más antiguas tienen precios más bajos, en promedio.

Resumen de Resultados:

- **MSE**: Mide cuán lejos están las predicciones de los valores reales. Un valor bajo es indicativo de un buen modelo.
- **R^2**: Mide cuán bien el modelo explica la variabilidad de los datos. Un valor cercano a 1 es deseable.
- **Coeficientes**: Ofrecen información sobre qué tan influyentes son las variables independientes en la predicción del precio. Por ejemplo, si el coeficiente de MedInc es alto, significa que el ingreso medio es un factor importante para predecir el precio de las viviendas.

```
Coeficientes del modelo:
MedInc: 0.44867490966571777
HouseAge: 0.009724257517905508
AveRooms: -0.1233233428279591
AveBedrms: 0.7831449067929719
Population: -2.0296205800966055e-06
AveOccup: -0.003526318487134082
Latitude: -0.41979248658835977
Longitude: -0.4337080649639865
```

Conclusión

El modelo de regresión lineal ha sido entrenado con un conjunto de datos que contiene características clave relacionadas con las viviendas en California. Al evaluar su rendimiento, se ha utilizado el Error Cuadrático Medio (MSE) y el Coeficiente de Determinación R^2 como métricas principales.

MSE (Error Cuadrático Medio): El MSE nos da una idea de cuán bien el modelo ha logrado predecir los precios en comparación con los valores reales. Un valor bajo de MSE es indicativo de que las predicciones del modelo están bastante cerca de los precios reales, lo que sugiere que el modelo tiene un buen desempeño en términos de precisión.

R^2 (Coeficiente de Determinación): Este valor mide el porcentaje de variación en los precios de las viviendas que es explicado por el modelo. Un R^2 cercano a 1 sugiere que el modelo es altamente efectivo para predecir los precios, mientras que valores más bajos indicaría que el modelo no captura bien la variabilidad de los datos.

El gráfico de dispersión que compara los precios reales con los precios predichos muestra la capacidad del modelo para hacer predicciones precisas. En una dispersión ideal, los puntos deben estar alineados a lo largo de la línea roja, que representa el escenario en el que los valores predichos coinciden exactamente con los reales.

Si los puntos están cerca de la línea roja, el modelo está haciendo buenas predicciones. Sin embargo, si los puntos se dispersan alejándose de la línea, significa que el modelo está teniendo dificultades para predecir esos casos particulares. Esto puede indicar que el modelo no captura adecuadamente algunas características de los datos o que hay relaciones no lineales que el modelo de regresión lineal no está considerando.

Link del GitHub: https://github.com/Jair-Artreaga/Regresion_Lineal_Simple.git