Regresión Lineal Múltiple en Python

Por: Angel Francisco Hernández Gámez

Introducción

La regresión lineal múltiple es una técnica estadística ampliamente utilizada para analizar y modelar la relación entre una variable dependiente y múltiples variables independientes. Se trata de una extensión del modelo de regresión lineal simple y permite explicar un resultado cuantitativo en función de varios factores predictivos. En el contexto del aprendizaje automático, esta técnica pertenece al conjunto de algoritmos de aprendizaje supervisado y es fundamental para tareas de predicción continua.

La ecuación general del modelo de regresión lineal múltiple se expresa como:

$$Y = b + m_1X_1 + m_2X_2 + \ldots + m_nX_n$$

donde Y representa la variable a predecir, $X_1, X_2, \ldots, X_n$ son las variables predictoras, $m_1, m_2, \ldots, m_n$ son los coeficientes de regresión que indican la importancia de cada variable y b es el término constante o intercepto.

Metodología

Para completar esta actividad, se siguieron los pasos indicados en el libro "Aprende Machine Learning" de Juan Ignacio Bagnato. El objetivo fue implementar un modelo de regresión lineal múltiple en Python usando datos de artículos sobre Machine Learning. A continuación, se describen los pasos principales:

1. Carga de datos: Se utilizó un archivo CSV con información sobre artículos, incluyendo cantidad de palabras, número de enlaces, comentarios, imágenes y número de veces que el artículo fue compartido.
2. Filtrado: Se eliminaron artículos con más de 3000 palabras o más de 80,000 compartidos para evitar valores atípicos que afectaran el modelo.
3. Creación de variable adicional: Se generó una nueva variable llamada `suma`, que representa la suma del número de enlaces, comentarios e imágenes, como una medida agregada de interacción.

4. Preparación de datos: Se construyó una matriz de entrada (X) con dos variables: `Word count` y `suma`. La variable objetivo (Y) fue `# Shares`.
5. Entrenamiento del modelo: Se utilizó la clase `LinearRegression` de la biblioteca `scikit-learn` para ajustar el modelo con los datos de entrada.
6. Evaluación del modelo: Se calcularon el error cuadrático medio (MSE) y el coeficiente de determinación (R^2) para evaluar el rendimiento del modelo.
7. Predicción: Se realizó una predicción para un artículo con 2000 palabras, 10 enlaces, 4 comentarios y 6 imágenes, lo que da una `suma` de 20.

Fragmento del código:

```
suma = (
    filtered_data["# of Links"] +
    filtered_data['# of comments'].fillna(0) +
    filtered_data['# Images video']
)

dataX2 = pd.DataFrame()
dataX2["Word count"] = filtered_data["Word count"]
dataX2["suma"] = suma

XY_train = np.array(dataX2)
z_train = filtered_data['# Shares'].values

regr2 = linear_model.LinearRegression()
regr2.fit(XY_train, z_train)
z_pred = regr2.predict(XY_train)
```

Resultados

Tras aplicar el modelo, se obtuvieron los siguientes resultados:

- Coeficientes: $[6.63, -483.41]$
- Intercepto: aproximadamente 5570.93
- Error cuadrático medio (MSE): $352{,}122{,}816.48$
- Coeficiente de determinación (R^2): 0.11

Esto indica que, por cada palabra adicional, el número de veces que se comparte un artículo aumenta en promedio en 6.63. Sin embargo, un aumento en la variable `suma` se relaciona con una disminución de 483.41 en la cantidad de compartidos. El R^2 de 0.11 implica que el modelo explica un 11% de la variabilidad de los datos, lo cual es bajo, pero representa una mejora respecto al modelo de una sola variable.

Conclusión

La implementación de la regresión lineal múltiple en Python permitió comprender los fundamentos de este tipo de modelo y aplicarlo en un caso práctico con datos reales. Aunque el modelo generado no tiene un alto poder predictivo, se logró mostrar cómo añadir variables adicionales puede mejorar ligeramente la capacidad explicativa del modelo.

Este ejercicio refuerza la importancia de la ingeniería de características y del análisis exploratorio de datos para construir modelos efectivos. Además, resalta la necesidad de considerar alternativas como modelos no lineales, técnicas de regularización o la incorporación de más datos relevantes en futuras iteraciones. Finalmente, el modelo entrenado permitió realizar predicciones básicas, demostrando su utilidad como punto de partida en el aprendizaje automático.