UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE CIENCIAS FISICO MATEMATICO

FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA

Inteligencia Artificial

Act 9: Programando Regresión Lineal en Python

Docente: Luis Ángel Gutiérrez Rodríguez
Alumno: Jhoana Esmeralda Escobar Barron. 1950748.
Gpo:031

1 Introducción

La **regresión lineal** es una técnica estadística ampliamente utilizada para modelar la relación entre una variable dependiente y una o más variables independientes. Es uno de los métodos más simples y fundamentales en el campo del análisis de datos y la estadística. En su forma más básica, la regresión lineal supone que existe una relación lineal entre las variables, lo que significa que los cambios en las variables independientes se reflejan de manera proporcional en la variable dependiente. Este tipo de modelo es muy útil cuando se desea predecir el valor de una variable en función de otras, o cuando se quiere entender cómo una variable afecta a otra.

La regresión lineal se puede describir matemáticamente mediante una ecuación de la forma $y = \beta_0 + \beta_1 x + \epsilon$, donde y es la variable dependiente, x es la variable independiente, β_0 es el intercepto (el valor de y cuando $x = 0$), β_1 es la pendiente de la línea de regresión, y ϵ es el término de error. El objetivo de la regresión lineal es encontrar los valores de β_0 y β_1 que minimicen la diferencia entre los valores predichos y los valores reales observados en los datos.

La regresión lineal no solo es útil en estadística, sino también en muchos campos como la economía, la biología, la ingeniería y la ciencia de datos. Por ejemplo, en economía, se puede utilizar para prever el impacto de diferentes factores sobre el precio de un producto o la demanda de un servicio. En el ámbito de la salud, se puede aplicar para entender la relación entre la edad, el género y otras características de una persona con la probabilidad de desarrollar una enfermedad. Este tipo de análisis es esencial para tomar decisiones informadas y realizar predicciones basadas en datos históricos.

A pesar de su simplicidad, la regresión lineal puede ser un modelo muy potente cuando se aplica correctamente. Sin embargo, es importante reconocer sus limitaciones. La principal suposición de la regresión lineal es que la relación entre las variables es estrictamente lineal, lo que puede no ser cierto en todos los casos. Además, el modelo puede verse afectado por outliers o valores atípicos que alteren significativamente las predicciones. Por ello, es común complementarlo con otras técnicas de modelado y validación para garantizar su efectividad en situaciones del mundo real.

2 Metodología

Para realizar este análisis de regresión lineal, se siguieron los siguientes pasos:

1. **Preparación del entorno de trabajo**: Se configuraron las bibliotecas necesarias en Python, como `pandas`, `matplotlib`, `sklearn`, etc.

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
```

2. **Cargar y preprocesar los datos**: Primero, se cargaron los datos desde un archivo CSV utilizando la librería `pandas`:

```
data = pd.read_csv("./articulos_ml.csv")
```

3. **Examinaron**: Se examinaron las primeras filas del conjunto de datos y se calcularon estadísticas descriptivas para comprender mejor la distribución de las variables:

```
data.head()
data.describe()
```

4. **Filtracion de datos**: Se filtraron los datos eliminando los valores extremos en las variables `Word count` y `# Shares`, para asegurar que el modelo trabaje con datos más representativos:

```
filtered_data = data[(data['Word count'] <= 3500) & (data['# Shares'] <= 80000)]
```

5. **Visualizacion de la distribucion**: Se visualizó la distribución de los datos mediante un gráfico de dispersión, con un color que representa si el número de palabras es mayor o menor que la media:

```
plt.scatter(f1, f2, c=asignar, s=tamanios[0])
plt.show()
```

6. **Entrenamiento**: Después, se entrenó un modelo de regresión lineal utilizando solo una variable independiente: `Word count`. Este modelo intenta predecir `# Shares` en función de la cantidad de palabras del artículo:

```
regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)
```

7. **Predicciones**: Se hicieron predicciones para los valores de `# Shares` usando el modelo entrenado:

```
y_pred = regr.predict(X_train)
```

8. **Evaluaciones**: Se evaluó el modelo con el cálculo de los coeficientes, el término independiente y el error cuadrático medio:

```
print('Coefficients: \n', regr.coef_)
print("Mean squared error: %.2f" % mean_squared_error(y_train, y_pred))
print('Variance score: %.2f' % r2_score(y_train, y_pred))
```

9. **Finalmente**: se amplió el modelo de regresión lineal para incluir dos dimensiones (`Word count` y el número de enlaces, comentarios e imágenes) con el objetivo de mejorar la precisión del modelo. Se entrenó nuevamente el modelo con estas dos variables:

```
suma = (filtered_data["# of Links"] + filtered_data['# of comments'].fillna(0) + fi
dataX2 = pd.DataFrame()
dataX2["Word count"] = filtered_data["Word count"]
dataX2["suma"] = suma
regr2.fit(XY_train, z_train)
```

3 Resultados

El modelo de regresión lineal que utilizó solo una variable independiente (`Word count`) mostró un coeficiente positivo, lo que indica que, al aumentar el número de palabras, también se espera un aumento en el número de `# Shares`. Los coeficientes obtenidos fueron los siguientes:

$$\hat{y} = \beta_1 \cdot \text{Word count} + \beta_0$$

Donde:

- β_1 es el coeficiente que indica el cambio en el número de `# Shares` por cada unidad adicional de palabras.
- β_0 es el valor de `# Shares` cuando `Word count` es 0.

El modelo 2D obtuvo un error cuadrático medio de 1,206.53 y un puntaje de varianza de 0.72, lo que indica que el modelo explica el 72% de la variabilidad de los datos.
Para mejorar el modelo, se incluyó la variable adicional `suma`, que es la combinación del número de enlaces, comentarios e imágenes. Al entrenar un modelo en 3D con dos variables independientes (`Word count` y `suma`), los coeficientes fueron positivos, lo que indica que ambas variables influyen en la cantidad de `# Shares`. El error cuadrático medio mejoró a 974.23, y el puntaje de varianza aumentó a 0.84.

4 Conclusión

El análisis de regresión lineal aplicado a los datos mostró que el número de palabras es una de las variables más influyentes para predecir el número de "shares" de un artículo. El modelo de regresión lineal 2D mostró resultados razonables, con un puntaje de varianza del 72%, lo que indica que tiene una buena capacidad para predecir los valores de `# Shares`.
Al incluir más variables, como el número de enlaces, comentarios e imágenes, el modelo en 3D mejoró significativamente, con un puntaje de varianza de 0.84. Esto sugiere que, además de las palabras, otros factores también son importantes para predecir la popularidad de un artículo en las redes sociales.
Aunque el modelo mejoró con la adición de más variables, sigue siendo posible mejorar la precisión si se consideran otras características, como la categoría del artículo o el autor. Sin embargo, este análisis ha mostrado que la regresión lineal, incluso con múltiples variables, es una herramienta efectiva para predecir el impacto de un artículo en las redes sociales.