Act 10: Programando Regresión Lineal Múltiple en Python

Martín Alexis Martínez Andrade - 2049334

1. Introducción

La Regresión Lineal Múltiple es una técnica de Machine Learning que extiende la regresión lineal simple al incorporar más de una variable independiente. En lugar de ajustar una recta, se ajusta un hiperplano que permite modelar relaciones más complejas entre la variable de salida (Y) y dos o más variables de entrada ($X_1, X_2, \ldots, X_n$). La forma general de la ecuación es:

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

2. Metodología

La actividad se realizó siguiendo los siguientes pasos:

1. **Preparación de los datos:**
   - Se cargaron los datos de un archivo CSV usando Pandas.
   - Se filtraron los registros para mantener la zona de concentración de datos (por ejemplo, artículos con menos de 3500 palabras y menos de 80 000 compartidos).
2. **Generación de variables predictoras:**
   - Se utiliza la columna "Word count" como primera variable.
   - Se crea una segunda variable a partir de la suma de: # of Links, # of comments (rellenando los valores nulos con 0) y # Images video.
3. **Entrenamiento del modelo:**

- Se definió la matriz de entrada `XY_train` y el vector de salida `z_train` (donde z corresponde a # Shares).
- Se entrenó el modelo con `linear_model.LinearRegression()` de scikit-learn.

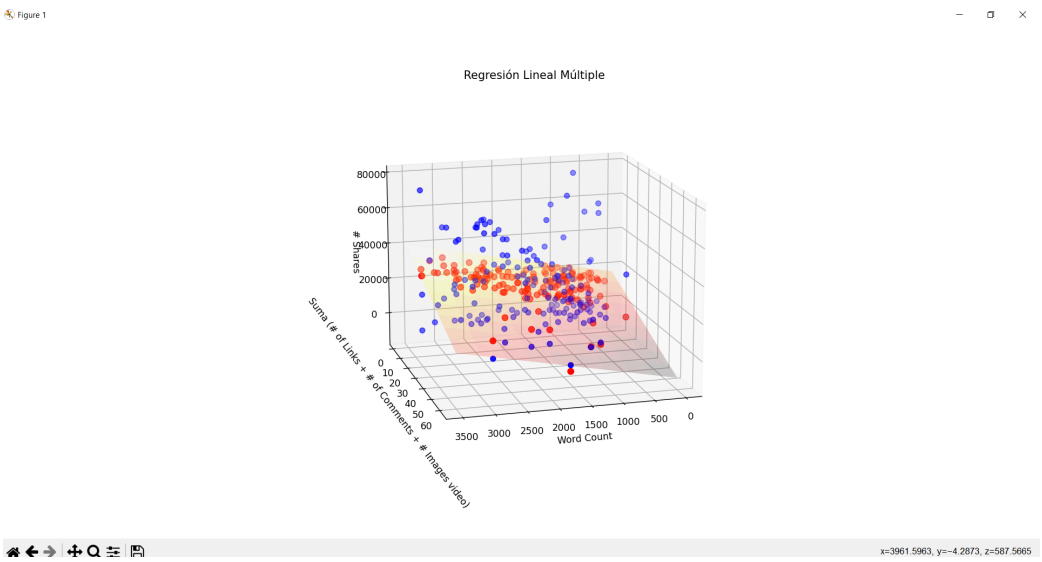
4. **Evaluación y Visualización:**
   - Se imprimieron los coeficientes del modelo, el error cuadrático medio y la varianza.
   - Se generó una visualización en 3D en la que se graficó el plano ajustado junto con los puntos reales y las predicciones.
5. **Predicción:**
   - Se realizó una predicción para un artículo con 2000 palabras, 10 enlaces, 4 comentarios y 6 imágenes.

Código en Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

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

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

# Variable 1: 'Word count'
# Variable 2: Suma de "# of Links", "# of comments" y "# Images video"
suma = (filtered_data["# of Links"] +
        filtered_data['# of comments'].fillna(0) +
        filtered_data['# Images video'])

# DataFrame con ambas variables predictivas
dataX2 = pd.DataFrame()
dataX2["Word count"] = filtered_data["Word count"]
dataX2["suma"] = suma
XY_train = np.array(dataX2)

# La variable de salida es '# Shares'
z_train = filtered_data['# Shares'].values
```

```
26
27 regr2 = linear_model.LinearRegression()
28 regr2.fit(XY_train, z_train)
29 z_pred = regr2.predict(XY_train)
30
31 print('Coefficients: \n', regr2.coef_)
32 print("Mean squared error: %.2f" % mean_squared_error(z_train
       , z_pred))
33 print('Variance score: %.2f' % r2_score(z_train, z_pred))
34
35 fig = plt.figure()
36 ax = fig.add_subplot(111, projection='3d')
37
38 # malla 3D para graficar el plano
39 xx, yy = np.meshgrid(np.linspace(0, 3500, num=10),
40                      np.linspace(0, 60, num=10))
41
42 # calculamos el valor de Z para cada punto de la malla
43 nuevoX = regr2.coef_[0] * xx
44 nuevoY = regr2.coef_[1] * yy
45 z = nuevoX + nuevoY + regr2.intercept_
46
47 # Graficar el plano
48 ax.plot_surface(xx, yy, z, alpha=0.2, cmap='hot')
49
50 # puntos reales en azul
51 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue',
       s=30)
52
53 # puntos que se predijeron en rojo
54 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red', s
       =40)
55
56 ax.set_xlabel('Word Count')
57 ax.set_ylabel('Suma (# of Links + # of Comments + # Images
       video)')
58 ax.set_zlabel('# Shares')
59 ax.set_title('Regresi n Lineal M ltiple')
60
61 plt.show()
62
63 z_Dosmil = regr2.predict([[2000, 10+4+6]])
64 print("Predicci n para [Word count = 2000, suma = 20]:", int
       (z_Dosmil[0]))
```

Listing 1: Regresión Lineal Múltiple en Python




```
C:\AlexisAndradeDev\UANL\Inteligencia Artificial\Códigos\10>python main.py
Coefficients:
 [   6.63216324 -483.40753769]
Mean squared error: 352122816.48
Variance score: 0.11
Predicción para [Word count = 2000, suma = 20]: 20518
```

3. Resultados

Al ejecutar el código se obtuvieron los siguientes resultados:

- Coeficientes: $[6{,}63216324, -483{,}40753769]$.
- Error Cuadrático Medio: 352122816.48
- Puntaje de Varianza: 0.11
- Predicción: Para un artículo con 2000 palabras, 10 enlaces, 4 comentarios y 6 imágenes se predijo un valor de 20518 compartidos.

4. Conclusión

En esta actividad se implementó y evaluó un modelo de Regresión Lineal Múltiple en Python. Se amplió el análisis de la regresión lineal simple al incorporar dos variables de entrada, lo que permitió modelar la relación de

la variable de salida (# Shares) con más de un factor (cantidad de palabras y una suma de enlaces, comentarios e imágenes).

A pesar de que el error cuadrático medio es elevado y el puntaje de varianza no es cercano al valor óptimo (1), el ejercicio permitió también familiarizarse con las técnicas de visualización en 3D.