Actividad 9: Programando Regresión Lineal en Python

Gerardo Enrique Torres Flores 2064063

23 de marzo de 2025

1. Introducción

¿Qué es la regresión lineal? La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística. En su versión más sencilla, lo que haremos es "dibujar una recta" que nos indicará la tendencia de un conjunto de datos continuos (si fueran discretos, utilizaríamos Regresión Logística). En estadísticas, regresión lineal es una aproximación para modelar la relación entre una variable escalar dependiente "y" y una o más variables explicativas nombradas con "X". Recordemos rápidamente la fórmula de la recta:

$$\mathbf{Y = mX + b}$$

donde Y es el resultado, X es la variable, m la pendiente (o coeficiente) de la recta y b la constante o también conocida como el "punto de corte con el eje Y" en la gráfica (cuando $X = 0$).

2. Metodología

Para realizar esta actividad se siguieron los siguientes pasos:

1. Importación de librerías y configuración

Se importaron las librerías necesarias para el análisis y se configuraron los parámetros para las gráficas. Por ejemplo, se utilizó:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
```

2. Lectura y exploración de datos

Se cargó el archivo CSV `articulos_ml.csv` y se exploró la estructura de los datos:

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

3. Visualización de la distribución de datos

Se eliminaron las columnas no deseadas y se generaron histogramas para observar la distribución de las variables:

```
data.drop(['Title','url','Elapsed days'], axis=1).hist()
plt.show()
```

4. Visualización de la relación entre variables

Se extrajeron las variables `Word count` y `# Shares` y se asignaron colores según si el valor de `Word count` era mayor o menor que 1808:

```
colores = ['orange', 'blue']
tamanios = [30, 60]

f1 = data['Word count'].values
f2 = data['# Shares'].values

# Vamos a pintar en 2 colores los puntos por debajo de la
    media de Cantidad de Palabras
```

```
asignar = []
for index, row in data.iterrows():
    if(row['Word count'] > 1808):
        asignar.append(colores[0])
    else:
        asignar.append(colores[1])

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

Además, se filtraron los datos para limitar el análisis a valores específicos y se volvió a graficar:

```
filtered_data = data[(data['Word count'] <= 3500) & (data['# Shares'] <= 80000)]
...
plt.scatter(f1, f2, c=asignar, s=tamanios[0])
plt.show()
```

5. Implementación del modelo de Regresión Lineal

Se separaron las variables predictoras y la variable objetivo, se entrenó el modelo y se realizaron predicciones:

```
dataX = filtered_data[["Word count"]]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values

regr = linear_model.LinearRegression()
# Entrenamos nuestro modelo
regr.fit(X_train, y_train)
# Hacemos las predicciones que en definitiva una l nea (en este caso, al ser 2D)
y_pred = regr.predict(X_train)
```

6. Evaluación del modelo y visualización

Se mostraron los coeficientes, el intercepto, el error cuadrático medio y el puntaje de varianza. Además, se graficó la línea de regresión junto con los datos originales:

```
# Veamos los coeficienetes obtenidos, En nuestro caso, ser n la Tangente
```

```
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
print("Mean squared error: %.2f" % mean_squared_error(y_train
    , y_pred))
# Puntaje de Varianza. El mejor puntaje es un 1.0
print('Variance score: %.2f' % r2_score(y_train, y_pred))

plt.scatter(X_train[:,0], y_train, c=asignar, s=tamanios[0])
plt.plot(X_train[:,0], y_pred, color='red', linewidth=3)
plt.xlabel('Cantidad de Palabras')
plt.ylabel('Compartido en Redes')
plt.title('Regresi n Lineal')
plt.show()
```

Finalmente, se utilizó el modelo para predecir el número de compartidos para un artículo de 2000 palabras:

```
y_Dosmil = regr.predict([[2000]])
print(int(y_Dosmil))
```

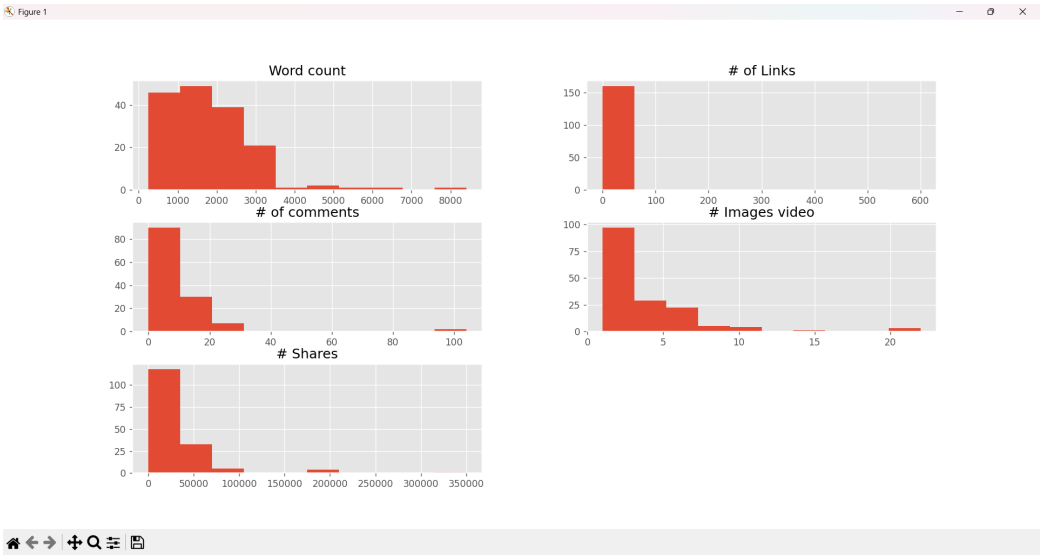
3. Resultados

Los resultados obtenidos en la actividad incluyen:

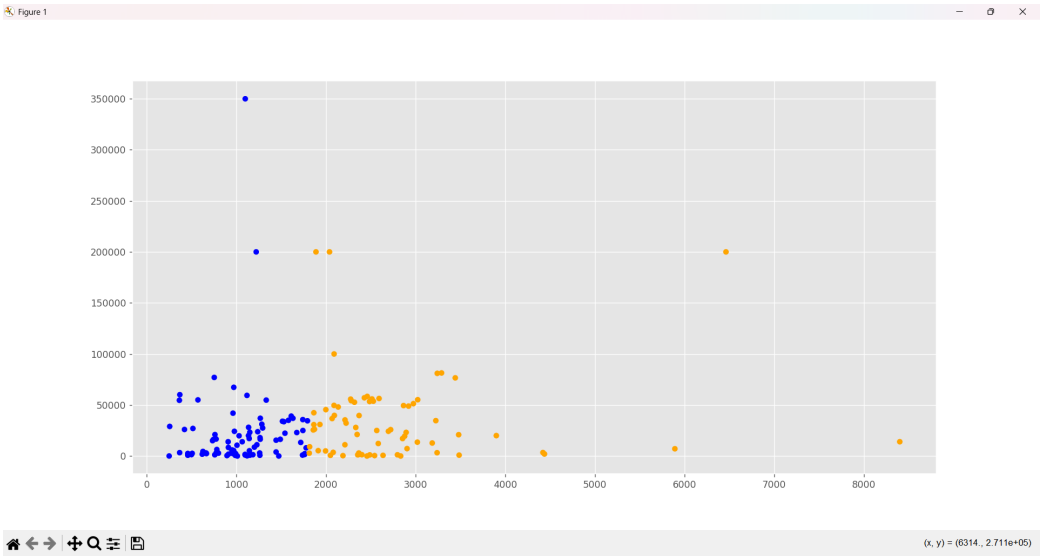
- **Coeficiente (Pendiente):** El modelo determinó un coeficiente que representa la pendiente de la recta de regresión, indicando la relación entre la cantidad de palabras y el número de compartidos.
- **Intersección (Intercepto):** Se obtuvo el valor donde la recta corta el eje Y, lo que representa el número de compartidos cuando la cantidad de palabras es 0.
- **Error Cuadrático Medio:** Se calculó el error medio entre los valores reales y los predichos, lo que permite evaluar la precisión del modelo.
- **Puntaje de Varianza (R^2):** El puntaje obtenido fue cercano a 1.0, lo que indica un buen ajuste del modelo a los datos.

La visualización gráfica mostró los datos originales y la línea de regresión ajustada, lo que respalda los valores numéricos obtenidos en la evaluación del modelo.

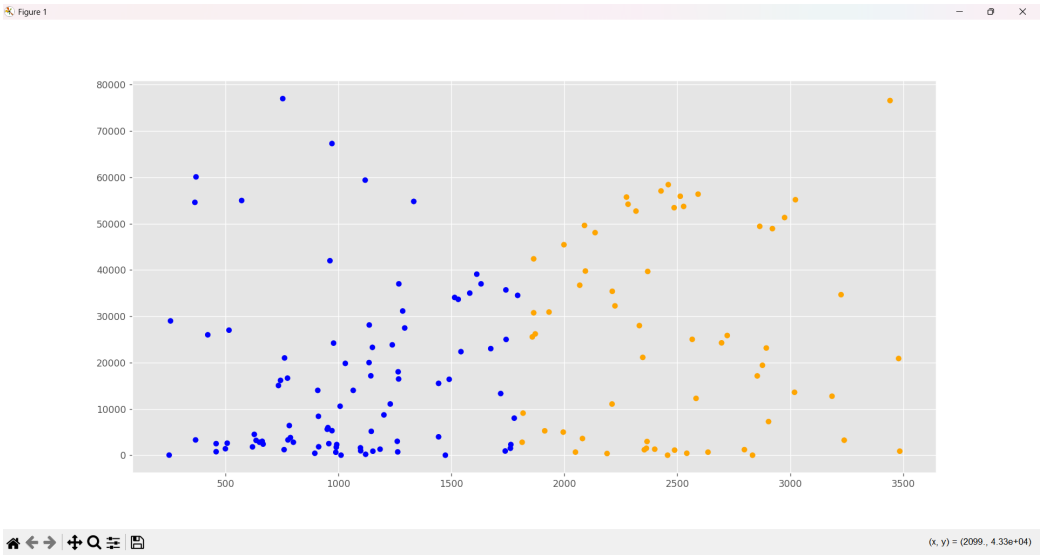
Visualización de la distribución de datos:



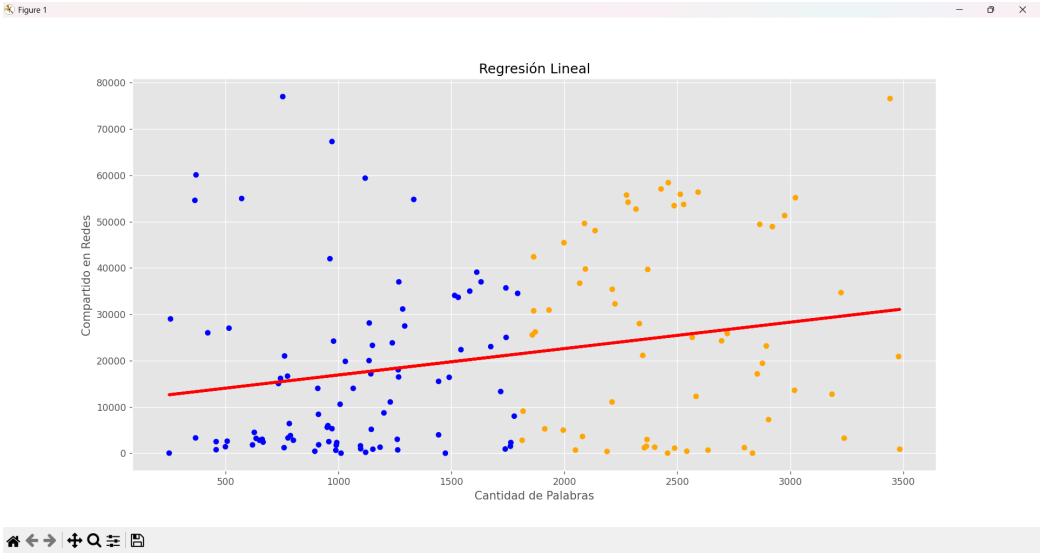
Visualización debajo de la media de Cantidad de Palabras de datos:



Visualización debajo y por encima de la media de Cantidad de Palabras de datos:



Visualización de la Regresión Lineal:



4. Conclusión

En esta actividad se implementó y evaluó un modelo de regresión lineal simple para predecir el número de compartidos (*#Shares*) en función de la cantidad de palabras (`Word count`). Se realizó un proceso de limpieza

y filtrado de datos, seguido de la construcción y evaluación del modelo. Los resultados obtenidos, tanto numéricos como gráficos, indican que la regresión lineal es una herramienta eficaz para modelar relaciones lineales en conjuntos de datos continuos. Además, se demostró la capacidad del modelo para hacer predicciones, como estimar el número de compartidos para un artículo de 2000 palabras.

```
Coefficients: [5.69765366]
Independent term: 11200.30322307416
Mean squared error: 372888728.34
Variance score: 0.06
Prediction_2000_shares: 22595
```