Actividad 9: Regresión Lineal con Python

Martín Alexis Martínez Andrade - 2049334

1. Introducción

La regresión lineal es un algoritmo de aprendizaje supervisado empleado tanto en Machine Learning como en estadística. Su objetivo es modelar la relación entre una variable dependiente (y) y una o más variables independientes (x). En su forma más simple, se trata de "dibujar una recta" que se ajusta al conjunto de datos, siguiendo la fórmula:

$$y = mx + b,$$

donde m representa la pendiente de la recta y b es la ordenada al origen, es decir, el valor de y cuando $x = 0$.

El algoritmo ajusta la recta minimizando la suma de los errores al cuadrado entre la predicción y los valores reales, utilizando métodos como la Ecuación Normal o mediante técnicas iterativas.

2. Metodología

Para la realización de esta actividad se siguieron los siguientes pasos:

1. **Importación de librerías y carga de datos:** Se importaron las librerías necesarias de Python, utilizando Pandas para el manejo del dataset y scikit-learn para construir el modelo.

2. **Visualización y preprocesamiento:** Se visualizaron algunas estadísticas básicas y se filtraron los datos para concentrarse en el rango donde se encontraban la mayoría de los registros (menos de 3500 palabras y menos de 80 000 compartidos).

3. **Entrenamiento del modelo:** Se prepararon los datos de entrada para entrenar un modelo de regresión lineal simple, utilizando la columna "Word count" como X y "# Shares" como la y.

4. **Evaluación y visualización:** Ya entrenado el modelo, se imprimieron los coeficientes, el error cuadrático medio y varianza. Finalmente, se realizó una predicción para un artículo con 2000 palabras.

Código Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Cargamos los datos de entrada
data = pd.read_csv("./articulos_ml.csv")
print("Dimensiones del DataFrame:", data.shape)

# Visualizamos los primeros registros
print(data.head())

print(data.describe())

# Visualizaci n inicial de las caracter sticas,
    descartando columnas no num ricas
data.drop(['Title','url', 'Elapsed days'], axis=1).hist()
plt.show()

# Filtrado de datos: se seleccionan registros con 'Word
    count' <= 3500 y '# Shares' <= 80000
filtered_data = data[(data['Word count'] <= 3500) & (data['#
    Shares'] <= 80000)]

# Configuraci n de colores seg n la media de palabras
    (media = 1808)
colores = ['orange','blue']
sizes = [30,60]

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

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

```

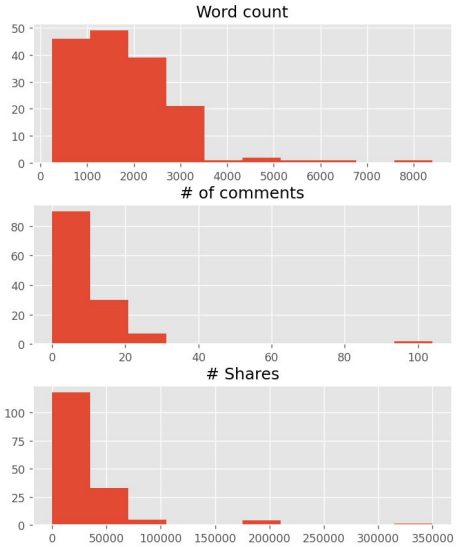
```
plt.scatter(f1, f2, c=asignar, s=sizes[0])
plt.xlabel("Word Count")
plt.ylabel("# Shares")
plt.title("Visualizaci n de Datos Filtrados")
plt.show()

# Preparaci n de los datos para regresi n lineal simple
dataX = filtered_data[["Word count"]]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values

# Creaci n y entrenamiento del modelo de Regresi n Lineal
model = linear_model.LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_train)

print('Coefficients: \n', model.coef_)
print('Intercept (t rmino independiente): \n',
    model.intercept_)
print("Mean squared error: %.2f" %
    mean_squared_error(y_train, y_pred))
print('Variance score: %.2f' % r2_score(y_train, y_pred))

# Predicci n: n mero de Shares para un art culo de 2000
    palabras
y_hat = model.predict([[2000]])
print("Predicci n para 2000 palabras:", int(y_hat[0]))
```

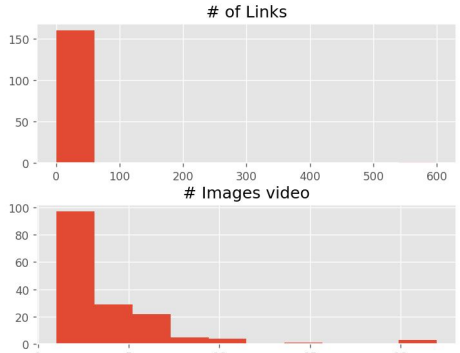
Listing 1: Regresión Lineal con Python

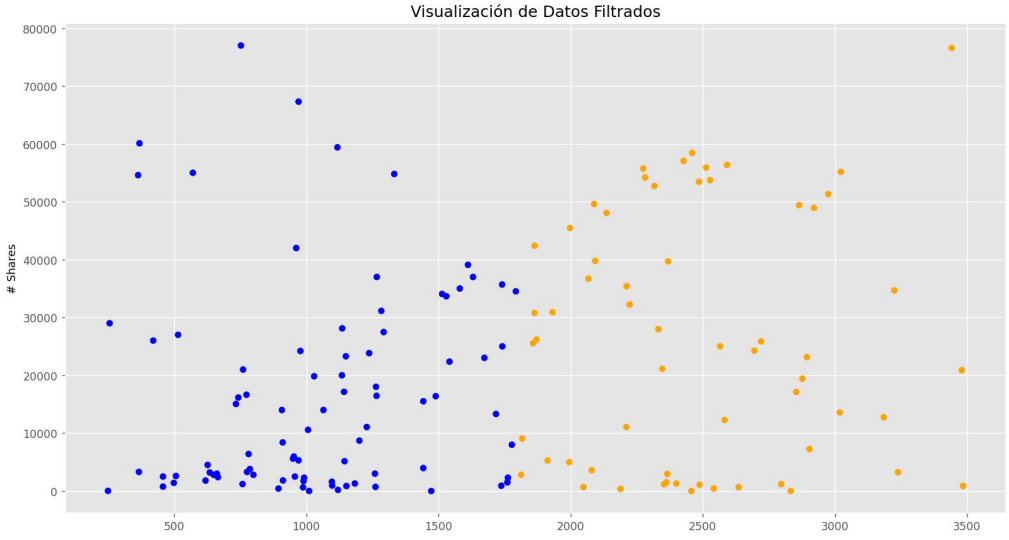
3. Resultados

Al ejecutar el código se obtuvieron, por ejemplo, los siguientes resultados:

- Coeficiente m: 5.69765366
- b: 11200.303223074163
- Error Cuadrático Medio: 372888728.34
- Puntaje de Varianza: 0.06
- Predicción: Para un artículo de 2000 palabras, el modelo predice 22595 compartidos.


Word count
# of comments
Shares


# of Links
Images video


Visualización de Datos Filtrados
Shares

```
C:\AlexisAndradeDev\UANL\Inteligencia Artificial\Códigos\9>python 9.py
Dimensiones del DataFrame: (161, 8)
                                          Title  ... # Shares
0  What is Machine Learning and how do we use it ...  ...   200000
1   10 Companies Using Machine Learning in Cool Ways  ...    25000
2  How Artificial Intelligence Is Revolutionizing...  ...    42000
3  Dbrain and the Blockchain of Artificial Intell...  ...   200000
4  Nasa finds entire solar system filled with eig...  ...   200000

[5 rows x 8 columns]
       Word count  # of Links  # of comments  # Images video  Elapsed days       # Shares
count   161.000000  161.000000     129.000000      161.000000    161.000000     161.000000
mean   1808.260870    9.739130       8.782946        3.670807     98.124224   27948.347826
std    1141.919385   47.271625      13.142822        3.418290    114.337535   43408.006839
min     250.000000    0.000000       0.000000        1.000000      1.000000       0.000000
25%     990.000000    3.000000       2.000000        1.000000     31.000000    2800.000000
50%    1674.000000    5.000000       6.000000        3.000000     62.000000   16458.000000
75%    2369.000000    7.000000      12.000000        5.000000    124.000000   35691.000000
max    8401.000000  600.000000     104.000000       22.000000   1002.000000  350000.000000
Coefficients:
 [5.69765366]
Intercept (término independiente):
 11200.30322307416
Mean squared error: 372888728.34
Variance score: 0.06
Predicción para 2000 palabras: 22595
```

4. Conclusión

En esta actividad se creó un modelo de regresión lineal simple utilizando Python y Scikit-Learn. Se llevó a cabo la importación y preprocesamiento de datos, la visualización inicial de las variables y el filtrado de registros para hacer un análisis.

El entrenamiento del modelo mostró cómo se pueden extraer coeficientes que definen la recta de mejor ajuste, a partir de la cual se realizaron predicciones, estimando la cantidad de veces que se compartiría un artículo en función de su número de palabras.