# Regresión Lineal

Valeria Ybarra López
Matrícula: 2047880

## 1 Introducción

La regresión lineal es un método utilizado en el análisis de datos, que permite predecir valores desconocidos basandose en otros valores conocidos que están relacionados. Este método establece una ecuación lineal $Y = mx + b$ que representa la relación matemática entre la variable dependiente (desconocida) y la variable independiente (conocida). Se enfoca en representar gráficamente la relación entre dos variables: "x" (variable independiente) y "y" (variable dependiente). La variable "x" se coloca en el eje horizontal (también conocida como variable explicativa o predictiva), mientras que "y", ubicada en el eje vertical, se denomina variable de respuesta (o pronosticada).

En el ámbito del machine learning, los algoritmos analizan grandes volúmenes de datos y determinan la ecuación de regresión lineal a partir de estos.

## 2 Metodología

Para la realización de esta activdad, se siguieron las instrucciones proporcionada en la página 26 del libro "Aprenda Machine Learning".

### 2.1 Creación de carpeta

Se creó una carpeta con el nombre de "Regresión Lineal" en donde se guardó un archivo .csv de entrada proporcionado por el libro patra poder realizar el código en pyhton, en esa misma carpeta se creó un arvhico .py para realizar la actividad.

### 2.2 Código

Primero importaremos las bibliotecas necesarias:

```
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')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

Leemos el archvio csv y cargamos los datos de entrada con la siguiente linea de código:

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

Para ver el tamaño y sus primeros registros (primeras cinco filas) con las siguientes lineas de código:

```
print(data.shape)
print(data.head())
```

Para poder explorar los datos, usaremos:

```
print(data.describe())
```

Con ayuda del método "data.drop()", eliminamos las columnas "Title", "url" y "Elapsed days" del DataFrame "data", y generamos histogramas de las columnas restantes con ".hist()"; utilizando "plt.show()" mostramos los histogramas.

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

Después, filtraremos los datos para conservar únicamente los registros con menos de 3500 palabras y los que tengan una cantidad de compartidos inferior a 80,000. Además, resaltaremos los puntos, utilizando el color azul para los que tienen menos de 1808 palabras (el promedio) y naranja para los que superan esa cantidad.

```
filtered_data = data[(data['Word count'] <= 3500) & (data['# Shares'] <= 800

colores=['orange','blue']
tamanios=[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=tamanios[0])
plt.show()
```

Implementamos una regresión lineal usando la biblioteca SKLearn, para este ejemplo tomaremos "Word Count" como dato de entrada y "#Shares" como las etiquetas. Creamos una instancia del modelo LinearRegression y lo entrenamos usando el método ".fit()". Después imprimimos los coeficientes y puntajes obtenidos.

```
dataX =filtered_data[["Word count"]]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values
regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)
y_pred = regr.predict(X_train)
print('Coefficients: \n', regr.coef_)
print('Independent term: \n', regr.intercept_)
```

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

Por último, probemos nuestro modelo para realizar una predicción. Supongamos que queremos estimar cuántos "compartir" obtendrá el artículo sobre Machine Learning con 2000 palabras:

Usamos el modelo de regresión lineal alamcenado en la variable "regr". El método ".predict()" toma como entrada el número de palabras (en este caso 2000) dentro de una lista. luego el modelo calculo cuántos "compartir" ("Shares") se espera obtener basándose en los datos con los que se entreno:

```
    y_Dosmil = regr.predict([[2000]])
print('Prediccion: ',int(y_Dosmil[0]))
```

# 3 Resultados

Tamaño de los datos:


Figure 1: Resultado de data.shape, se tiene 161 filas y 8 columnas.

Primeras 5 filas del DataFrame:

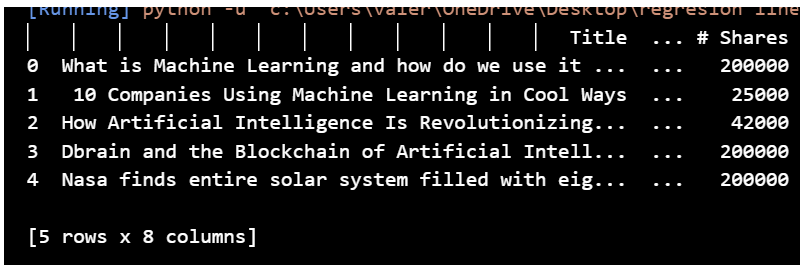


Figure 2: Resultado de data.head

Estadísticas descriptivas de las columnas numéricas del DataFrame:

| | Word count | # of Links | ... | Elapsed days | # Shares |
|---|---|---|---|---|---|
| count | 161.000000 | 161.000000 | ... | 161.000000 | 161.000000 |
| mean | 1808.260870 | 9.739130 | ... | 98.124224 | 27948.347826 |
| std | 1141.919385 | 47.271625 | ... | 114.337535 | 43408.006839 |
| min | 250.000000 | 0.000000 | ... | 1.000000 | 0.000000 |
| 25% | 990.000000 | 3.000000 | ... | 31.000000 | 2800.000000 |
| 50% | 1674.000000 | 5.000000 | ... | 62.000000 | 16458.000000 |
| 75% | 2369.000000 | 7.000000 | ... | 124.000000 | 35691.000000 |
| max | 8401.000000 | 600.000000 | ... | 1002.000000 | 350000.000000 |

[8 rows x 6 columns]

Figure 3: Resultado de data.describe

Histogramas generados:

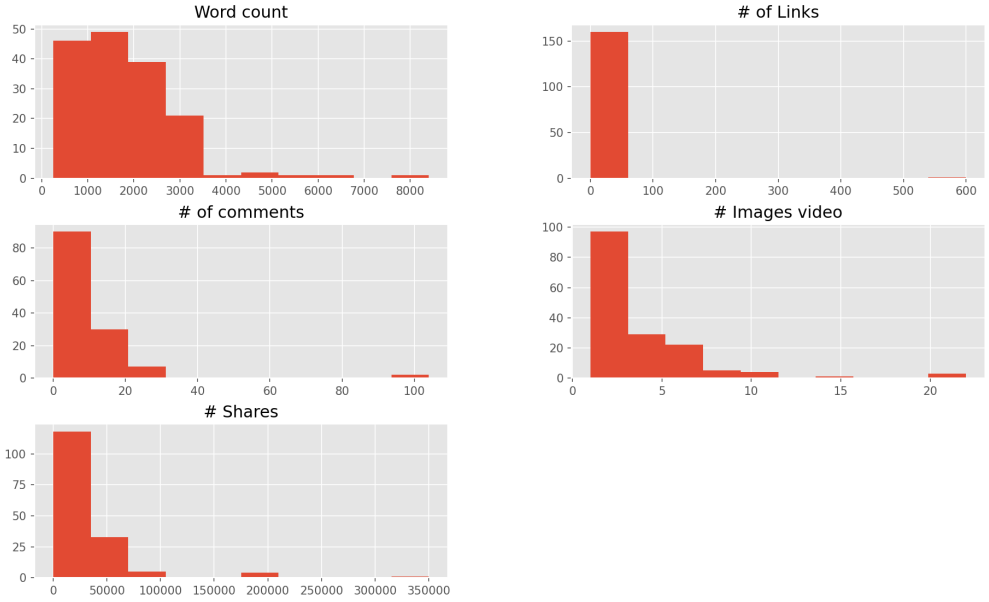


Figure 4: En estas gráficas vemos entre qué valores se concentran la mayoría de registros.

Gráfica Dispersión:

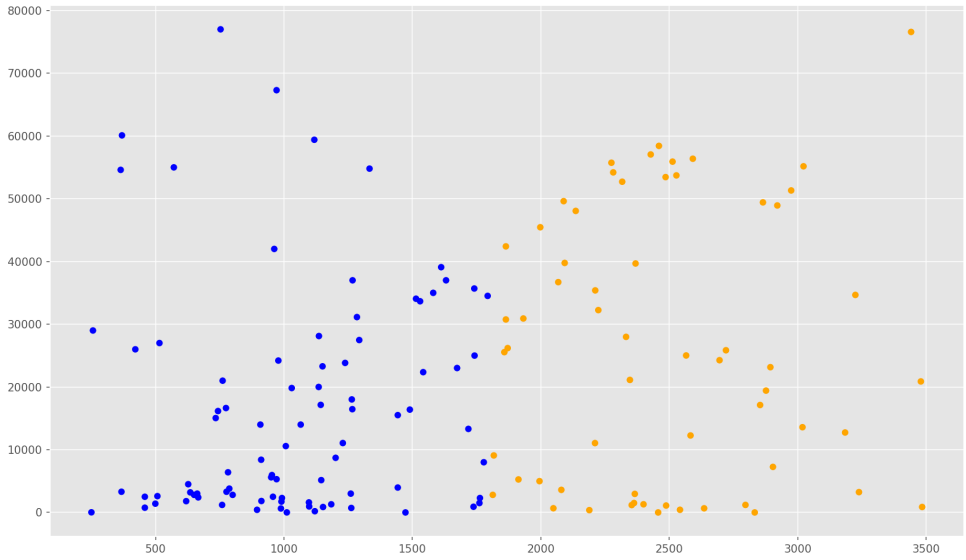


Figure 5: Después de filtrar los datos de cantidad de palabras para quedarnos con los registros con menos de 3500 palabras y también con los que tengan cantidad de compartidos menos a 80.000. Se muestran puntos en azul si tienen menos de 1808 palabras y en naranja si tienen más de 1808 palabras.

Regresión Lineal, Coeficientes y puntajes obtenidos, después de haber entrenado el modelo:

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

Figure 6: De la ecuación de la recta $Y = mx + b$ nuestra pendiente "m" es el coeficiente 5,69 y el término independiente "b" es 11200.

Predicción en regresión lineal simple:

```
Prediccion:  22595
```

Figure 7: Nos devuelve una predicción de 22595 "Shares" para un artículo de 2000 palabras.

# 4 Conclusión

En esta actividad se aprendió a implementar la regresión lineal para predecir valores desconocidos basándonos en datos conocidos. Desde preparar y explorar los datos hasta construir un

modelo predictivo con SKLearn, cada paso ayuda a comprender mejor cómo usar ML para realizar predicciones. Además, se visualizo los datos por medio de gráficos y analizamos métricas, lo que nos permitió evaluar la precisión del modelo.

# 5 Referencias

Bagnato, J. (2020). Aprende Machine Learning en Español.

Amazon Web Services. ¿Qué es la regresión lineal?. https://aws.amazon.com/es/what-is/linear-regression/