

# Regresión Lineal en Python

31 de Marzo de 2025

## 1 Introducción

La regresión lineal es un método estadístico fundamental que modela la relación entre una variable dependiente y una o más variables independientes (regresores). Un modelo que está conformado por una sola variable independiente se conoce como regresión lineal simple, mientras que un modelo con más de una variable independiente se conoce como regresión lineal múltiple.

El modelo asume una relación lineal entre las variables y se expresa mediante la ecuación:

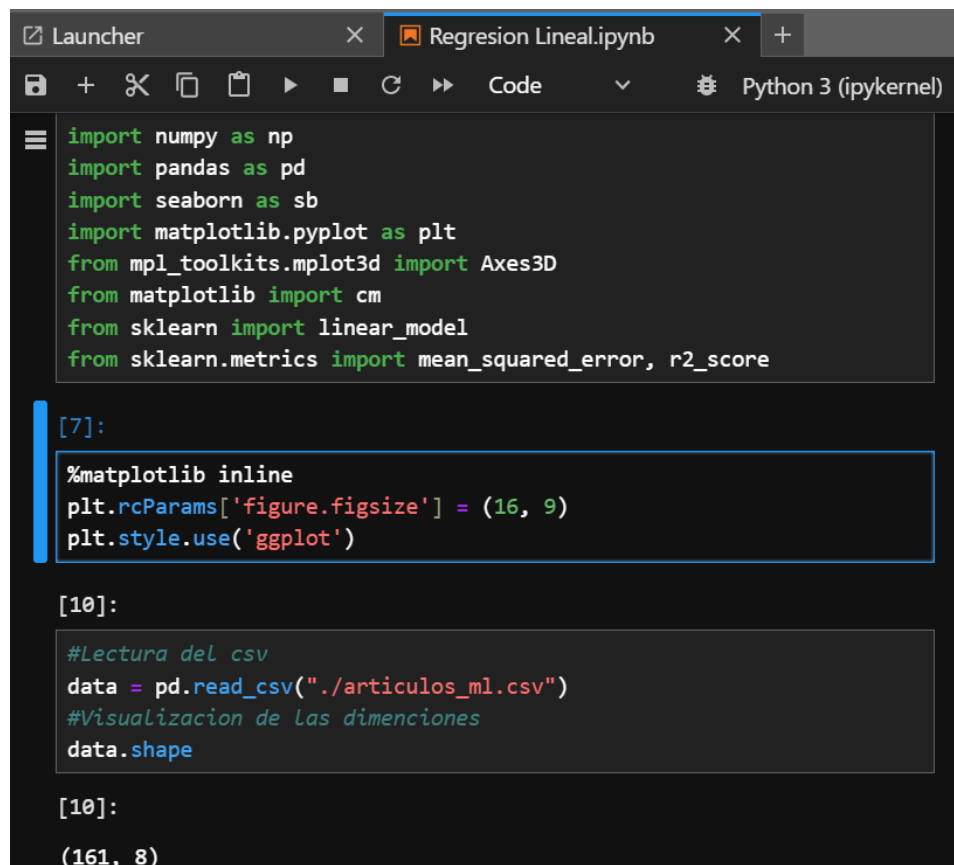
$$y = \beta_0 + \beta_1 x + \epsilon \quad (1)$$

donde:

- $y$  es la variable dependiente.
- $x$  es la variable independiente.
- $\beta_0$  es el intercepto.
- $\beta_1$  es la pendiente.
- $\epsilon$  es el término de error.

## 2 Metodología

Para realizar el ejercicio de regresión lineal, se siguieron los siguientes pasos:



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

[7]:
%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')

[10]:
#Lectura del csv
data = pd.read_csv("./articulos_ml.csv")
#Visualizacion de las dimensiones
data.shape

[10]:
(161, 8)
```

Figure 1: Importación de bibliotecas y preparación de datos.

## 2.1 Visualización General de los Datos de Entrada

En esta sección se eliminan las variables categóricas del DataFrame y se visualiza la distribución de los datos. En este ejercicio, se eligió que la variable dependiente del modelo de regresión lineal simple fuera el número de compartidos, mientras que la variable regresora fuera la cantidad de palabras.



Figure 2: Distribución de los datos después de eliminar las variables categóricas.

## 2.2 Filtración de los Datos

Ya que escogimos nuestras variables dependiente e independiente, visualizamos su relación con una gráfica scatterplot. También establecemos un límite superior al valor de los datos, ya que la mayoría de los datos de estas dos variables se concentran en los valores más pequeños, aunque existen algunos datos atípicos. Para evitar que estas anomalías influyan en el entrenamiento de nuestro modelo, no las incluimos en nuestro dataset.

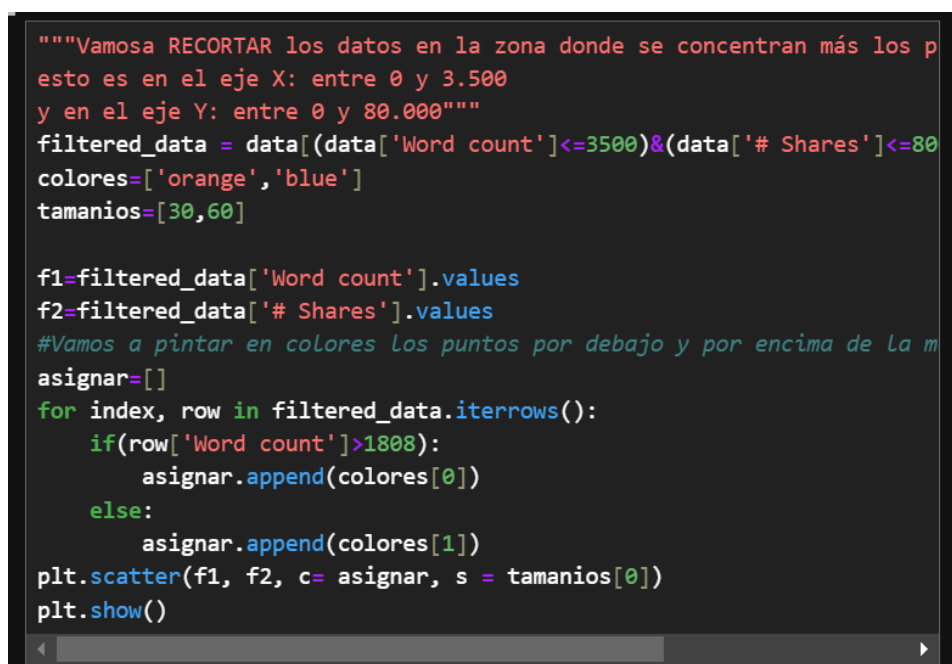


Figure 3: Filtración de datos.

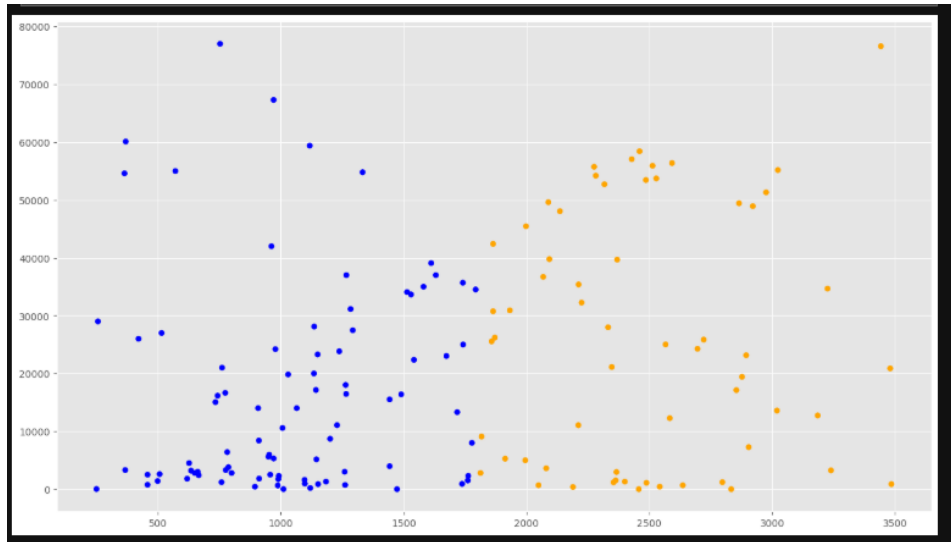


Figure 4: Gráfica de datos.

### 2.3 División de Datos en Conjuntos de Entrenamiento

Asignamos los valores de la columna “Word Count” como  $X$ , y los valores de la columna “# Shares” como nuestra  $Y$ .

```
#Asignamos los datos para el entrenamiento
dataX = filtered_data[["Word count"]]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values
```

Figure 5: División de datos.

### 2.4 Ajuste del Modelo

En esta parte, se crea el modelo de regresión lineal usando Scikit-Learn y se ajusta usando los datos de entrenamiento. Además, generamos las predicciones de  $Y$  del modelo, las cuales se pueden agregar a la gráfica que realizamos anteriormente.

```
#creamos el objeto de regresion lineal
regr = linear_model.LinearRegression()

#Entrenamos el modelo
regr.fit(X_train, y_train)

#Hacemos las predicciones
y_pred = regr.predict(X_train)
```

Figure 6: Ajuste del Modelo.

### 2.5 Evaluación del Modelo

Ya que tenemos el modelo creado y ajustado, podemos imprimir los valores del intercepto y la pendiente. Por lo tanto, nuestro modelo de regresión lineal simple queda de la siguiente manera:

```
#Vemos los coeficientes obtenidos
print("Coeficientes: ", regr.coef_)
#Valor donde corta el eje x
print("independent term: ", regr.intercept_)
#Error cuadrático medio
print("MSE: %.2f" % mean_squared_error(y_train, y_pred))
#Puntaje de varianza
print("Variance score: %.2f" % r2_score(y_train, y_pred))
```

Coeficientes: [5.69765366]  
independent term: 11200.30322307416  
MSE: 372888728.34  
Variance score: 0.06

Figure 7: Evaluación del modelo.

$$y = 11200.303 + 5.7x \quad (2)$$

Es importante notar que el valor del error medio cuadrado es bastante grande, lo que sugiere que un modelo de regresión lineal simple puede no ser la mejor opción para explicar la relación entre los datos.

### 3 Resultados

Para verificar los resultados, visualizamos nuevamente el scatterplot, pero agregándole la recta del modelo de regresión lineal para evaluar su ajuste. También podemos evaluar su desempeño verificando las predicciones del modelo con valores específicos de la gráfica.



Figure 8: Resultados.

Para evaluar el desempeño del modelo, verificamos su predicción del número de compartidos en redes cuando hay 2000 palabras en el post, es decir, cuál es nuestra  $y$  cuando  $x = 2000$ . En la gráfica se pueden observar dos datos alrededor de las 2000 palabras: uno cerca de  $y = 5000$  y otro cercano a  $y = 46000$ . Estos valores son muy distintos, lo que nos lleva a verificar si el modelo acierta en alguna de estas dos predicciones.

```
#Predecimos la cantidad de veces que sera compartido un articulo con
y_Dosmil = regr.predict([[2000]])
print(int(y_Dosmil))
```

22595

Figure 9: Prueba.

El modelo no acertó en ninguno de los dos valores, pero su predicción se encuentra a una mediación de estos datos. Si volvemos a examinar la gráfica, podemos notar que la relación entre nuestras dos variables no parece ser lineal. Si lo fuera, la mayoría de los datos estarían más próximos a la recta de regresión.

Antes de probar un modelo de regresión de otro tipo, podemos intentar agregar una variable independiente adicional al modelo para mejorar los resultados, lo que nos llevaría a un modelo de regresión lineal múltiple.

## 4 Conclusión

Los modelos de regresión lineal son herramientas estadísticas utilizadas para explicar la relación entre conjuntos de datos de manera lineal. Existen dos tipos principales:

- Regresión lineal simple: compuesta por una variable dependiente y una variable independiente o regresora.
- Regresión lineal múltiple: incluye más de una variable regresora.

En este caso, se intentó comprobar si el número de palabras en un post tenía una relación lineal y causal con el número de compartidos. Al graficar los datos y entrenar el modelo, se observó que esto no era cierto. La forma de la gráfica no indica una correlación fuerte entre las dos variables, y el modelo refleja esta falta de relación.

El modelo no acertó en su predicción para ninguno de los dos valores cercanos a  $x = 2000$ , aunque su predicción estuvo a mediación de estos datos. Antes de probar un modelo de regresión de otro tipo, podemos intentar agregar una variable independiente adicional para mejorar los resultados, lo que nos llevaría a un modelo de regresión lineal múltiple.