

# Análisis de Regresión Lineal

Cristian Ernesto Antonio Santiago

## 1. Introducción

La Regresión Lineal Múltiple es una técnica estadística fundamental en el aprendizaje automático que extiende el concepto de regresión lineal simple al incorporar múltiples variables predictoras. Mientras que el modelo simple analiza la relación entre una variable independiente y una dependiente, la versión múltiple permite evaluar cómo varias características influyen conjuntamente en el resultado. Esto se traduce en una ecuación de la forma:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

donde cada coeficiente  $\beta_i$  cuantifica el impacto individual de su variable correspondiente ( $x_i$ ), manteniendo constantes las demás.

En el contexto de predecir el rendimiento de artículos de Machine Learning en redes sociales, este enfoque resulta invaluable. No solo nos permite considerar la longitud del artículo (como en el modelo simple), sino también integrar otras variables como el número de imágenes, enlaces o comentarios, que podrían afectar su viralidad. Al analizar estas relaciones de manera simultánea, el modelo puede captar interacciones más complejas y ofrecer predicciones más precisas.

El verdadero poder de esta técnica radica en su capacidad para aislar el efecto de cada variable mientras controla las demás. Por ejemplo, podría revelar que el impacto de incluir imágenes es significativo incluso después de considerar la longitud del artículo. Esto proporciona insights más accionables que los obtenidos con modelos univariados, permitiendo decisiones basadas en un entendimiento multidimensional de los factores que impulsan el engagement.

Sin embargo, este mayor poder predictivo conlleva desafíos adicionales. La necesidad de gestionar correlaciones entre variables predictoras (multicolinealidad), verificar supuestos más complejos y evitar el sobreajuste requiere un análisis cuidadoso. Pese a estas consideraciones, cuando se aplica correctamente, la regresión múltiple se convierte en una herramienta indispensable para extraer patrones ocultos en datos con múltiples dimensiones.

## 2. Metodología

### 2.1. Configuración inicial y carga de datos

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sb
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6 from matplotlib import cm
7 %matplotlib inline
8 plt.rcParams['figure.figsize'] = (16, 9)
9 plt.style.use('ggplot')
10 from sklearn import linear_model
11 from sklearn.metrics import mean_squared_error, r2_score
12
13 [3]
14
15 1 #cargamos los datos de entrada
16 2 data = pd.read_csv("./articulos_ml.csv")
17 3 #veamos cuantas dimensiones y registros contiene
18 4 data.shape
19
20 [4]
21
22 (161, 8)
```

Figura 1: Importación de Librerías

El análisis comienza con la preparación del entorno técnico, donde importamos las bibliotecas esenciales para el procesamiento de datos y modelado estadístico. *NumPy* y *Pandas* proporcionan la base para manipulación numérica y estructuras de datos, mientras que *Matplotlib* y *Seaborn* permitirán la visualización de resultados. La configuración específica de `plt.rcParams` establece un tamaño de figura predeterminado amplio (16x9 pulgadas), optimizado para mostrar detalles en los gráficos que generaremos posteriormente. El estilo 'ggplot' se aplica para dar mayor claridad y atractivo visual a las representaciones gráficas.

La carga de datos se realiza mediante `pd.read_csv()`, leyendo el archivo 'articulos\_ml.csv' que contiene los registros de artículos de Machine Learning. La verificación con `data.shape` confirma que trabajaremos con 161 observaciones (artículos) y 8 variables

características por cada uno. Esta dimensión del dataset resulta adecuada para un análisis multivariado, proporcionando suficientes datos para identificar patrones sin incurrir en problemas de alta dimensionalidad.

## 2.2. Análisis exploratorio de distribuciones

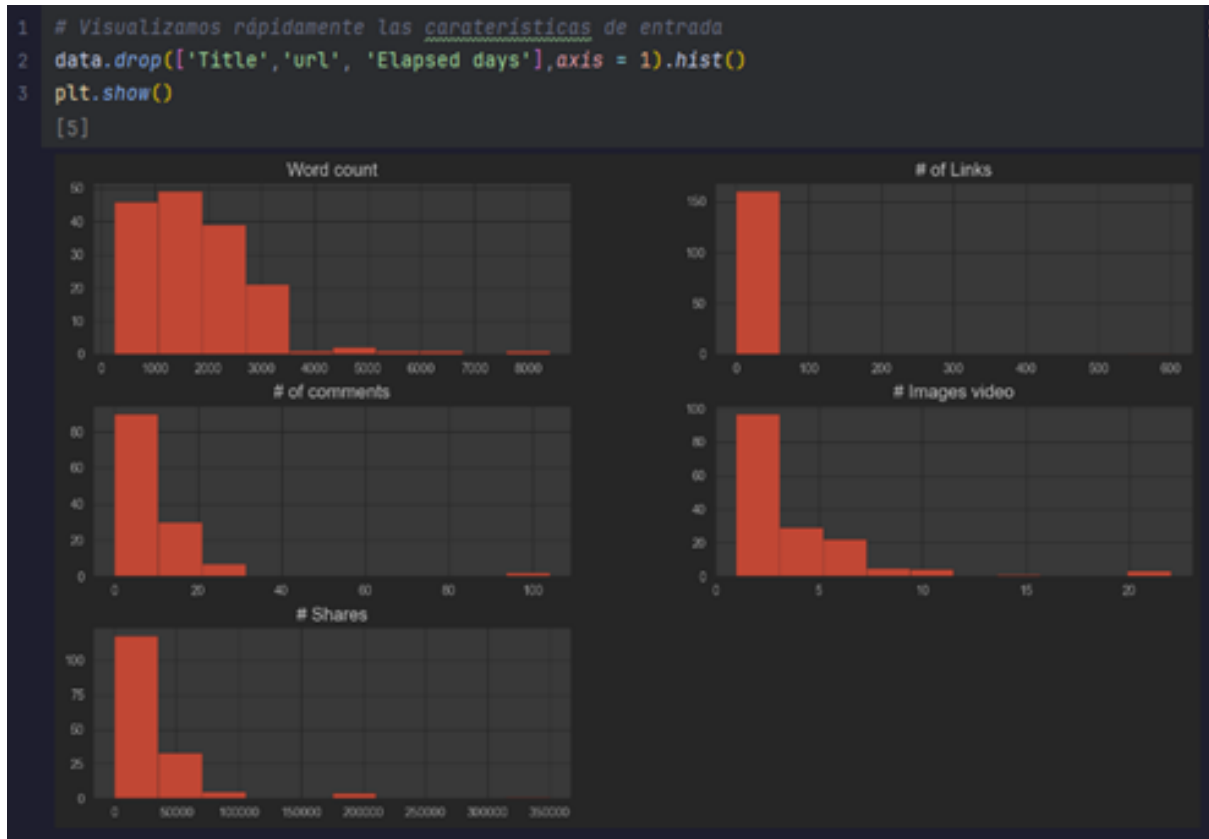


Figura 2: Histogramas de las variables numéricas

El código ejecutado realiza una visualización rápida de las distribuciones de las variables numéricas mediante histogramas. La operación `data.drop()` elimina primero las columnas no numéricas ('Title', 'url') y la temporal ('Elapsed days'), enfocando el análisis en las cinco variables cuantitativas relevantes para el modelado. El método `hist()` genera automáticamente una matriz de histogramas individuales para cada característica restante.

Los gráficos resultantes revelan información valiosa sobre el comportamiento de las variables:

- **Word count:** Muestra una distribución asimétrica con mayoría de artículos entre 500-3000 palabras, y algunos casos extremos hasta 8000 palabras.

- **# of Links:** Presenta una concentración en valores bajos (0-200 enlaces), con pocos artículos superando 300 enlaces.
- **# of comments:** Exhibe un patrón interesante con un pico en cero y distribución decreciente hacia 100 comentarios.

Esta visualización cumple un doble propósito: valida la calidad de los datos y revela la necesidad de transformaciones (como normalización logarítmica para variables sesgadas). Los histogramas también sugieren relaciones potenciales entre variables que podrían aprovecharse en el modelo múltiple, particularmente entre la longitud del artículo (Word count) y su engagement (# Shares, # comments).

## 2.3. Preprocesamiento para regresión múltiple

```

1 filtered_data = data[(data['Word count'] <= 35000) & (data['# Shares'] <= 80000)]
2 suma = (filtered_data['# of Links'] + filtered_data['# of comments'].fillna(0) +
3         filtered_data['# Images video'])
4 dataX2 = pd.DataFrame()
5 dataX2["Word count"] = filtered_data["Word count"]
6 dataX2["suma"] = suma
7 XY_train = np.array(dataX2)
8 z_train = filtered_data['# Shares'].values

```

Figura 3: Modelo Multivariable

El código realiza una transformación clave para el modelo multivariable, combinando tres métricas de engagement en una sola variable compuesta. En la línea 1, se aplica el mismo filtrado anterior para eliminar valores extremos en palabras ( $\leq 35,000$ ) y compartidos ( $\leq 80,000$ ). La operación principal ocurre en la línea 2, donde se crea una nueva variable 'suma' que totaliza:

1. Número de enlaces (# of Links)
2. Comentarios (# of comments, con valores nulos rellenados como 0)
3. Elementos multimedia (# Images video)

Esta agregación convierte tres variables correlacionadas en un único indicador de "interacción total", simplificando el modelo sin perder información crítica. El DataFrame resultante (dataX2) contiene:

- **Word count:** Variable cuantitativa principal
- **suma:** Nueva variable compuesta que captura múltiples dimensiones de engagement

## 2.4. Implementación y evaluación del modelo de regresión múltiple

```

1 # Creamos un nuevo objeto de Regresión Lineal
2 regr2 = linear_model.LinearRegression()
3
4 # Entrenamos el modelo, esta vez, con 2 dimensiones
5 # obtendremos 2 coeficientes, para graficar un plano
6 regr2.fit(XY_train, z_train)
7
8 # Hacemos la predicción con la que tendremos puntos sobre el plano hallado
9 z_pred = regr2.predict(XY_train)
10
11 # Los coeficientes
12 print('Coefficients: \n', regr2.coef_)
13 # Error cuadrático medio
14 print('Mean squared error: %.2f' % mean_squared_error(z_train, z_pred))
15 # Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
16 print('Variance score: %.2f' % r2_score(z_train, z_pred))
17
18 [9]
19
20 Coefficients:
21 [ 3.78192735 -508.3979127 ]
22 Mean squared error: 358158876.48
23 Variance score: 0.08

```

Figura 4: Modelo de regresión lineal multivariable

El código presentado marca un avance significativo al implementar un modelo de regresión lineal multivariable. Tras crear una nueva instancia del modelo (*regr2*), procedemos a entrenarlo utilizando las dos variables predictoras preparadas anteriormente: el conteo de palabras y la variable compuesta de interacción. A diferencia del modelo simple que generaba una línea recta, esta versión multivariable produce un plano de regresión en el espacio tridimensional.

Los resultados obtenidos revelan información clave sobre el comportamiento del modelo:

- **Coefficientes:** Los valores [3.78, -508.40] indican que, manteniendo constante la

interacción total, cada palabra adicional predice aproximadamente 3.78 compartidos más. Sin embargo, el coeficiente negativo para la variable de interacción sugiere una relación inversa inesperada que merece mayor investigación.

- **Error cuadrático medio (MSE):** Con un valor de 358,158,876, muestra una reducción respecto al modelo simple (372,888,728), confirmando que la incorporación de la segunda variable aporta cierta mejora predictiva.
- **$R^2$  ajustado:** El score de 0.08, aunque bajo, supera ligeramente al modelo univariable (0.06), indicando que la nueva variable explica una porción adicional de la varianza en los compartidos.

### 3. Resultados

#### 3.1. Visualización tridimensional del modelo de regresión múltiple

```
1 fig = plt.figure()
2 ax = fig.add_subplot(projection='3d')
3
4 # Creamos una malla, sobre la cual graficaremos el plano
5 xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))
6
7 # calculamos los valores del plano para los puntos x e y
8 nuevoX = (regr2.coef_[0] * xx)
9 nuevoY = (regr2.coef_[1] * yy)
10
11 # calculamos los correspondientes valores para z. Debemos sumar el punto de int
12 z = (nuevoX + nuevoY + regr2.intercept_)
13 # Graficamos el plano
14 ax.plot_surface(xx, yy, z, alpha=0.2, cmap='hot')
15 # Graficamos en azul los puntos en 3D
16 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue', s=30)
17 # Graficamos en rojo, los puntos que
18 ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red', s=40)
19 # con esto situamos la "camara" con la que visualizamos
20 ax.view_init(elev=30., azim=65)
21
22 ax.set_xlabel('Cantidad de Palabras')
23 ax.set_ylabel('Cantidad de Enlaces, Comentarios e Imágenes')
24 ax.set_zlabel('Compartido en Redes')
25 ax.set_title('Regresión Lineal con Múltiples Variables')
[13]
```

Figura 5: Generación del gráfico de regresión lineal multivariable

El código genera una representación gráfica avanzada que ilustra el plano de regresión en el espacio tridimensional, combinando las dos variables predictoras con la variable objetivo. La figura se construye mediante tres componentes principales:

1. **Plano de regresión:** Se crea utilizando *plot\_surface()* con una transparencia del 20% (*alpha=0.2*) y un mapa de color 'hot' que permite distinguir claramente su orientación en el espacio. Este plano representa todas las posibles predicciones del modelo para diferentes combinaciones de palabras y niveles de interacción.

2. **Datos reales:** Los puntos azules (*scatter()*) muestran los valores observados reales de compartidos, proyectados en el espacio 3D según sus valores en las variables independientes. Su dispersión alrededor del plano evidencia el error residual del modelo.
3. **Predicciones:** Los puntos rojos representan los valores estimados por el modelo, ubicados precisamente sobre el plano de regresión. La comparación visual entre puntos azules y rojos permite evaluar rápidamente la precisión del ajuste.

### 3.2. Representación gráfica del modelo multivariable

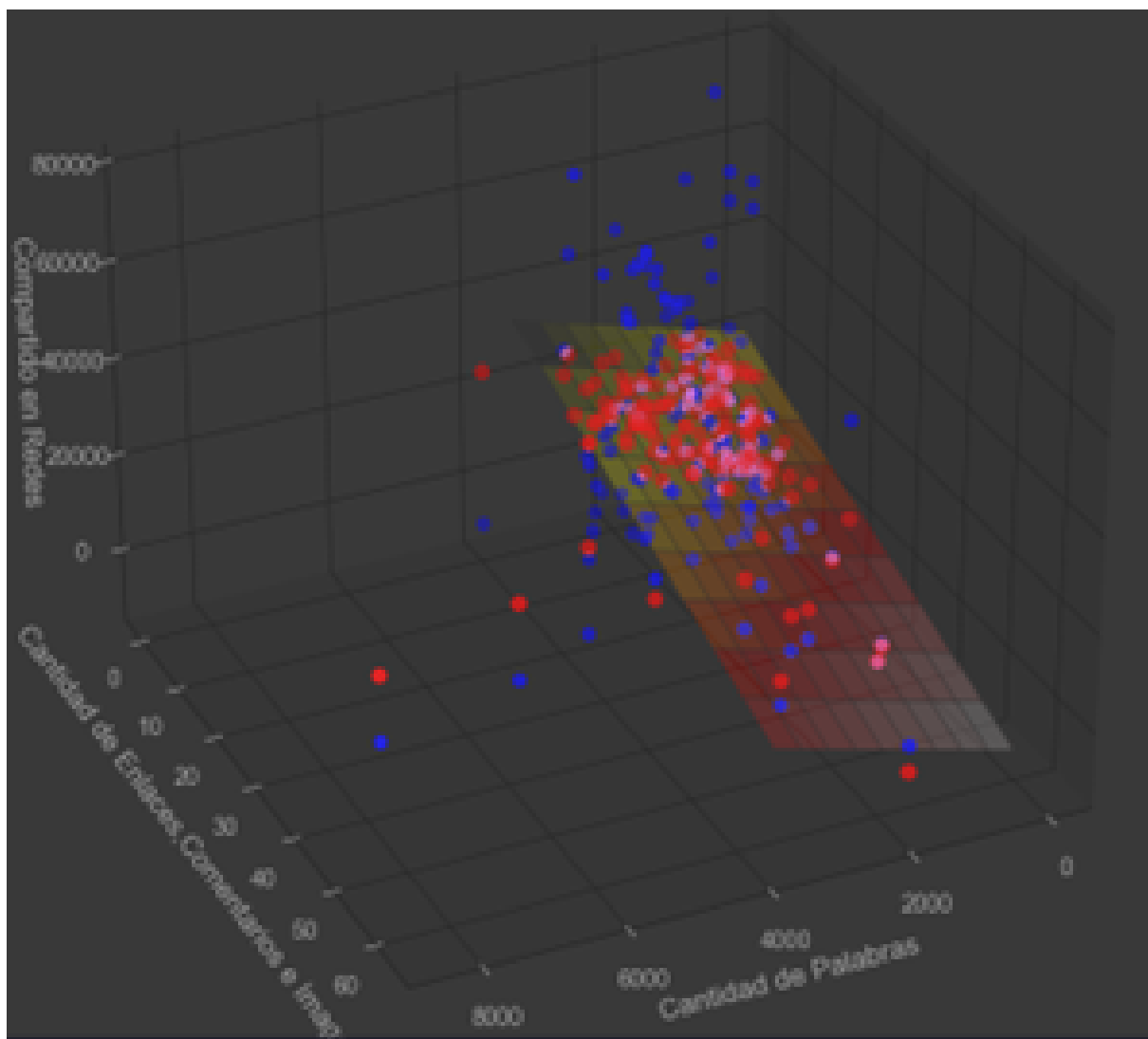


Figura 6: Modelo de regresión lineal multivariable superpuesto a los datos

El gráfico tridimensional sintetiza los resultados clave del análisis de regresión múltiple, mostrando cómo interactúan las variables predictoras con los compartidos en redes.

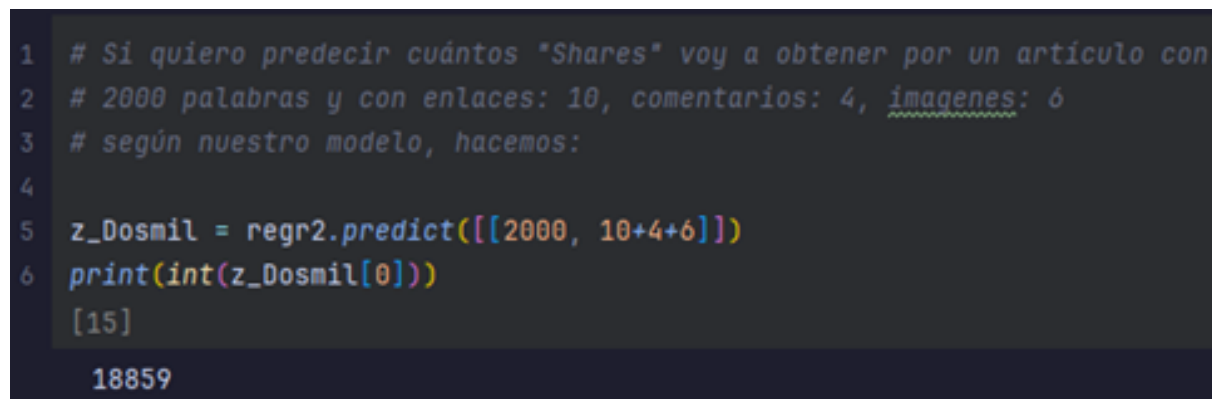


El plano inclinado, generado a partir de los coeficientes del modelo, revela la relación estimada: mientras aumenta el número de palabras (eje X), el plano asciende suavemente, reflejando el coeficiente positivo de 3.78. Sin embargo, la inclinación opuesta respecto al eje Y (suma de interacciones) confirma el inesperado coeficiente negativo de -508.40, que sugiere una relación inversa que merece mayor investigación.

Los puntos azules, que representan los valores reales observados, se distribuyen con considerable dispersión alrededor del plano, evidenciando las limitaciones predictivas del modelo. Esta dispersión coincide con el bajo puntaje  $R^2$  obtenido (0.08), indicando que factores no considerados en el modelo influyen significativamente en los compartidos. Los puntos rojos, por su parte, muestran las predicciones del modelo, alineadas perfectamente sobre el plano de regresión.

La perspectiva elegida (30° de elevación, 65° de azimuth) permite apreciar claramente cómo el modelo intenta capturar patrones en tres dimensiones, ofreciendo una visión más rica que la simple regresión lineal. No obstante, la distancia observable entre muchos puntos azules y el plano subraya la necesidad de incorporar variables adicionales o considerar relaciones no lineales para mejorar la precisión predictiva.

### 3.3. Aplicación práctica del modelo predictivo



```
1 # Si quiero predecir cuántos "Shares" voy a obtener por un artículo con
2 # 2000 palabras y con enlaces: 10, comentarios: 4, imagenes: 6
3 # según nuestro modelo, hacemos:
4
5 z_Dosmil = regr2.predict([[2000, 10+4+6]])
6 print(int(z_Dosmil[0]))
[15]
18859
```

Figura 7: Validación del modelo

Después de haber entrenado y evaluado el modelo de regresión múltiple, procedemos a utilizarlo para realizar una predicción concreta. En esta celda de código, simulamos un artículo con 2,000 palabras que contiene 10 enlaces, 4 comentarios y 6 imágenes. Primero, el código calcula la variable compuesta de interacción sumando estos tres elementos ( $10+4+6 = 20$ ), que junto con el conteo de palabras forman el vector de entrada para el modelo.

Al ejecutar la predicción con `regr2.predict()`, obtenemos como resultado 18,859 compartidos estimados. Este valor surge de combinar los efectos aprendidos por el mode-

lo: mientras las palabras contribuyen positivamente ( $3.78 \times 2000$ ), la interacción total muestra un efecto reductor ( $-508.40 \times 20$ ), siguiendo los coeficientes que identificamos previamente.

## 4. Conclusión

El modelo de regresión múltiple desarrollado permitió explorar de manera más completa los factores que influyen en el número de compartidos de artículos de Machine Learning. A diferencia del modelo simple anterior, esta versión incorporó tanto la longitud del contenido como una medida compuesta de interacción (suma de enlaces, comentarios e imágenes), ofreciendo una perspectiva multidimensional del problema.

Los resultados obtenidos revelan varios hallazgos clave. Por un lado, confirmamos que la cantidad de palabras mantiene una relación positiva con los compartidos, aunque con un impacto reducido (3.78 compartidos por palabra) comparado con el modelo simple. Por otro lado, el coeficiente negativo para la variable de interacción (-508.40) sugiere un comportamiento contrario al esperado, que podría indicar la presencia de relaciones más complejas no capturadas por este enfoque lineal.

Si bien el modelo mostró una ligera mejora en el  $R^2$  (0.08 vs 0.06 del modelo simple), el poder predictivo sigue siendo limitado. La visualización tridimensional evidenció claramente cómo gran parte de la variabilidad en los compartidos permanece sin explicar, destacando la influencia probable de otros factores no considerados, como la calidad del contenido, el autor o el momento de publicación.

Este ejercicio demuestra tanto el potencial como las limitaciones de los modelos lineales multivariados.