

Inteligencia Artificial
Act9: Regresión Lineal Simple

Arturo Garza Rodríguez

March 2025

1. Introducción

¿Qué es una regresión lineal?

La regresión lineal es un método estadístico utilizado para predecir el valor de una variable dependiente a partir de una o más variables independientes mediante una relación matemática lineal. Esta técnica busca ajustar una ecuación lineal que minimice las diferencias entre los valores observados y los valores predichos. A través de este modelo, se estima la relación entre las variables y se pueden hacer predicciones sobre valores futuros basándose en datos conocidos.

La tarea de la regresión en el aprendizaje automático consiste en predecir un parámetro (Y) a partir de un parámetro conocido X. El algoritmo deberá minimizar el coste de una función de error cuadrático y esos coeficientes corresponderán con la recta óptima.

2. Metodología

2.1. Requerimientos

2.1.1. Jupyter notebook

Para el desarrollo de este ejercicio, estaremos codificando en una Jupyter Notebook, lo que nos permitirá ir probando parte por parte y viendo los resultados al momento.

2.1.2. Descargar dataset

Es necesario descargar el archivo 'articulos_ml.csv' que se nos proporciona en el libro 'Aprende Machine Learning' de Juan Ignacio Bagnato.

2.2. Desarrollo de código

2.2.1. Imports



```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

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

data = pd.read_csv('C:/Users/artur/OneDrive/Documentos/GitHub/IA_EJ2025/Mar/Actividad_9/articulos_ml.csv')
```

Figura 1: Como se puede observar, se realizaron imports para distintos paquetes, entre ellos matplotlib, pandas, numpy y sklearn.

2.2.2. Información relevante del dataset

```
data.shape
```

```
[3]
```

```
... (161, 8)
```

Figura 2: Podemos comprobar que las dimensiones del dataset son de 161 x 8.

```
data.head()
```

```
[4]
```

```
...
```

	Title	url	Word count	# of Links	# of comments	# Images video	Elapsed days	# Shares
0	What is Machine Learning and how do we use it ...	https://blog.signals.network/what-is-machine-l...	1888	1	2.0	2	34	200000
1	10 Companies Using Machine Learning in Cool Ways	NaN	1742	9	NaN	9	5	25000
2	How Artificial Intelligence Is Revolutionizing...	NaN	962	6	0.0	1	10	42000
3	Dbrain and the Blockchain of Artificial Intell...	NaN	1221	3	NaN	2	68	200000
4	Nasa finds entire solar system filled with eig...	NaN	2039	1	104.0	4	131	200000

Figura 3: Vemos los primeros 5 registros del dataset a través del comando `data.head()`.

```
data.describe()
```

```
[5]
```

```
...
```

	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

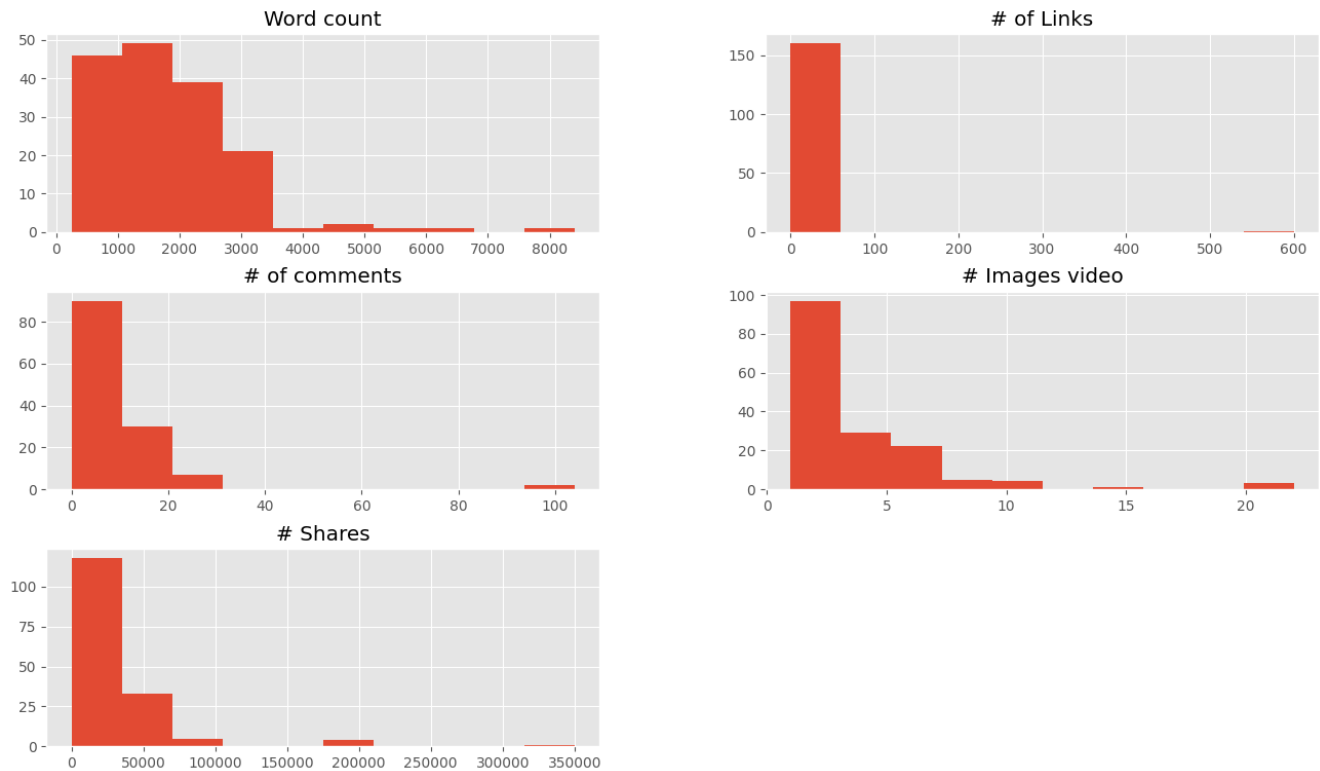
Figura 4: `data.describe()` es útil para obtener una visión rápida de la distribución y características de los datos en el DataFrame.

2.2.3. Histograma

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

[6] Python

Figura 5: Se eliminan las columnas 'Title', 'url' y 'Elapsed days' y se genera un histograma.



2.2.4. Filtración de registros

```
filtered_data=data[(data['Word count']<=3500)&(data['# Shares']<=80000)]

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

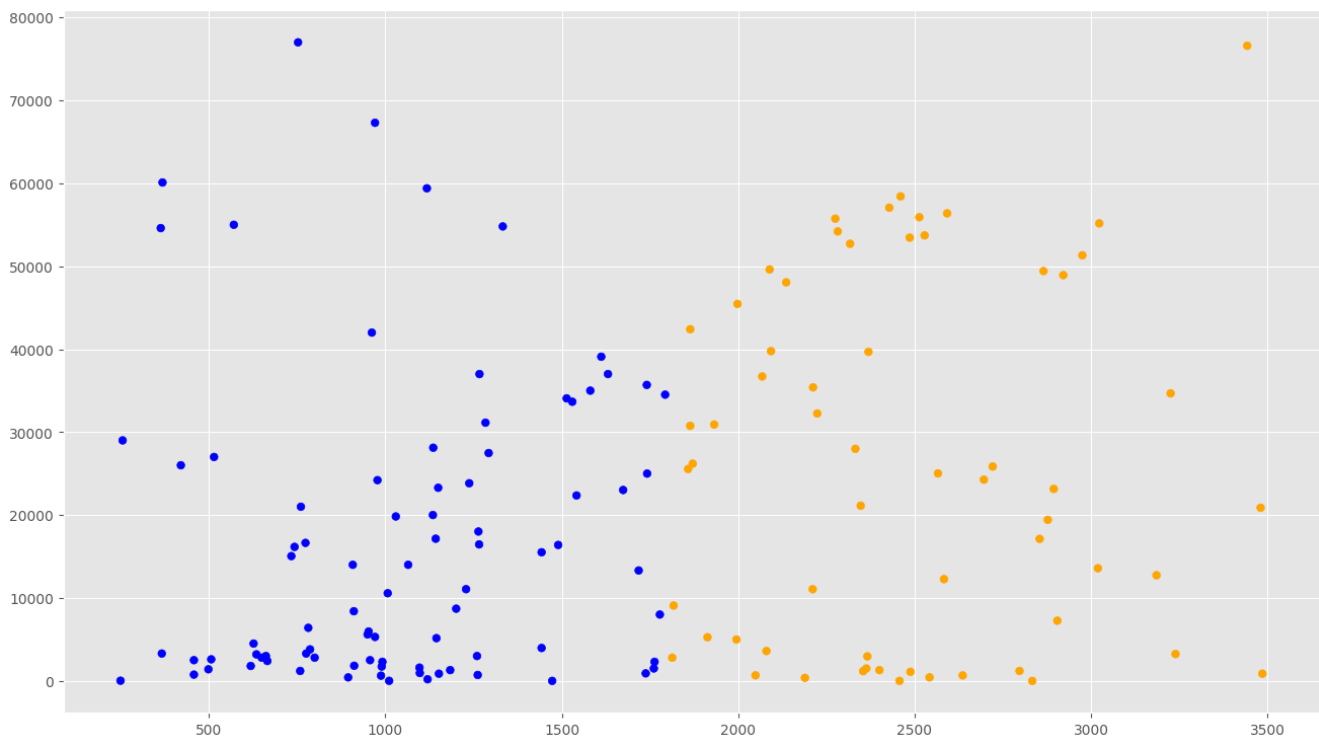
f1=filtered_data['Word count'].values
f2=filtered_data['# Shares'].values

asignar=[]
for index,row in filtered_data.iterrows():
    if(row['Word count']>1800):
        asignar.append(colores[0])
    else:
        asignar.append(colores[1])

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

[7] Python

Figura 6: Se filtran los registros cuyo conteo de palabras sea ≤ 3500 y cantidad de compartidos ≤ 80000 y se categorizan por colores.



3. Resultados

En todos los pasos anteriores se mostraron los resultados al momento, por último, tenemos el resultado de hacer la regresión lineal.

3.1. Regresión lineal

```
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: \n", mean_squared_error(y_train, y_pred))
print("Variance score \n:", r2_score(y_train, y_pred))

plt.scatter(f1,f2,c=asignar,s=tamamos[0])
plt.plot(f1, y_pred, color='red', linewidth=2)
plt.show()
```

Python

Figura 7: Se definen las variables X_{train} y y_{train} y se genera un modelo de regresión lineal a través de `linear_model.LinearRegression()`.

Se encontró que:

- Coefficients: [5.69765366]
- Independent term: 11200.30322307416
- Mean squared error: 372888728.3398661
- Variance score: 0.05519842281951404

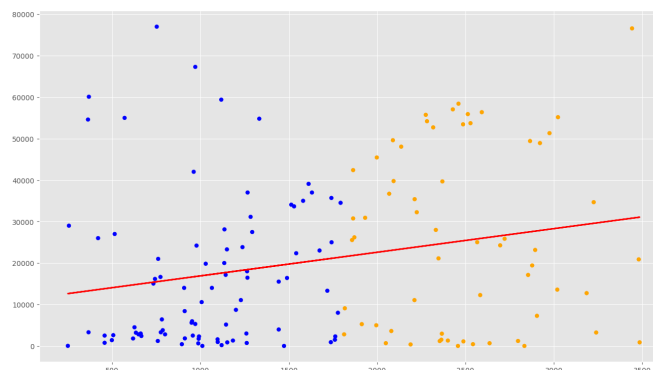


Figura 8: Se aprecia cómo la línea regresora es el valor que el modelo espera dada una x .

4. Conclusión

Encuentro el uso de una regresión lineal bastante útil para determinar un estimado de algo dado otro algo, podremos no tener la información necesaria para conocer este dato (variable dependiente), pero con los registros anteriores podemos aproximar el valor esperado.

Es importante destacar que para poder generar un modelo certero necesitamos de suficientes registros previos para así poder ponderar correctamente los coeficientes en la ecuación de regresión.

5. Referencias

- Material de clase
- <https://aws.amazon.com/es/what-is/linear-regression/>
- <https://www.ibm.com/mx-es/topics/linear-regression>
- <https://ebac.mx/blog/regreson-lineal>