

Inteligencia Artificial
Act10: Regresión Lineal Múltiple

Arturo Garza Rodríguez

March 2025

1. Introducción

¿Qué es una regresión lineal múltiple?

La regresión lineal múltiple es un modelo estadístico que analiza la relación entre una variable dependiente continua y varias variables independientes. Este modelo se basa en una combinación lineal de los predictores, donde cada uno se multiplica por un coeficiente estimado. Se usa para entender cómo múltiples factores influyen en una variable, y se evalúa su significancia mediante pruebas estadísticas.

2. Metodología

2.1. Requerimientos

2.1.1. Jupyter notebook

Igual que en el caso de la regresión lineal simple, 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

Se utilizará el mismo dataset que en el ejercicio anterior '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

A screenshot of a Jupyter Notebook interface with a dark theme. The code cell contains the following Python code:

```
import pandas as pd
import numpy as np
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
%matplotlib inline

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')

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

The status bar at the bottom shows '[9] ✓ 0.0s' on the left and 'Python' on the right.

Figura 1: Se utiliza la misma configuración que el ejercicio anterior, con la diferencia de que se filtran los datos desde la declaración del dataset.

Se utilizarán dos variables regresoras para este caso, la primera seguirá siendo la cantidad de palabras, mientras que la segunda variable será la suma de 3 columnas de entrada: la cantidad de enlaces, comentarios y cantidad de imágenes.

2.2.2. Suma de columnas

```
suma = (filtered_data["# of Links"] +
        filtered_data['# of comments'].fillna(0) +
        filtered_data['# Images video'])

dataX2 = pd.DataFrame()
dataX2["Word count"] = filtered_data["Word count"]
dataX2["suma"] = suma

XY_train = np.array(dataX2)
z_train = filtered_data['# Shares'].values
```

[11] ✓ 0.0s Python

Figura 2: Se declara la segunda variable regresora como la suma de las 3 columnas rellenoando los valores nulos de ' of comments' con 0.

Generamos un dataframe que almacene ambas variables regresoras. Creamos un array con los registros combinados de 'Word count' y 'suma' y declaramos la variable z, como la variable dependiente.

2.2.3. Modelo regresor

```
regr2 = linear_model.LinearRegression()

# Entrenamos el modelo, esta vez, con 2 dimensiones
# Obtendremos 2 coeficientes, para graficar un plano
regr2.fit(XY_train, z_train)

# Hacemos la predicción con la que tendremos puntos sobre el plano hallado
z_pred = regr2.predict(XY_train)

# Los coeficientes
print('Coefficients: \n', regr2.coef_)

# Error cuadrático medio
print("Mean squared error: %.2f" % mean_squared_error(z_train, z_pred))

# Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
print('Variance score: %.2f' % r2_score(z_train, z_pred))
```

✓ 0.0s Python

Definimos el modelo de regresión lineal como *regr2* a través de *linear_model.LinearRegression()*. Entrenamos el modelo con el arreglo previamente generado con ambas columnas como primer parámetro, y como segundo, la variable *z_train*. Predecimos los valores y los almacenamos en *z_pred*. De este modelo de regresión lineal con dos variables regresoras, se encontró que:

- Coefficients: [6.63216324 -483.40753769]
- Mean squared error: 352122816.48
- Variance score: 0.11

3. Resultados

Para representar los valores de la regresión lineal múltiple producida en los segmentos anteriores, utilizamos una gráfica 3d.

3.1. Regresión lineal

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d') # Ensure 3D projection

xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))

nuevoX = (regr2.coef_[0] * xx)
nuevoY = (regr2.coef_[1] * yy)
z = (nuevoX + nuevoY + regr2.intercept_)

ax.plot_surface(xx, yy, z, alpha=0.2, cmap='hot')

# Plot the original data points in blue
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue', s=30, label='Actual data')

# Plot the predicted data points in red
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red', s=40, label='Predicted data')

ax.view_init(elev=30., azim=65)

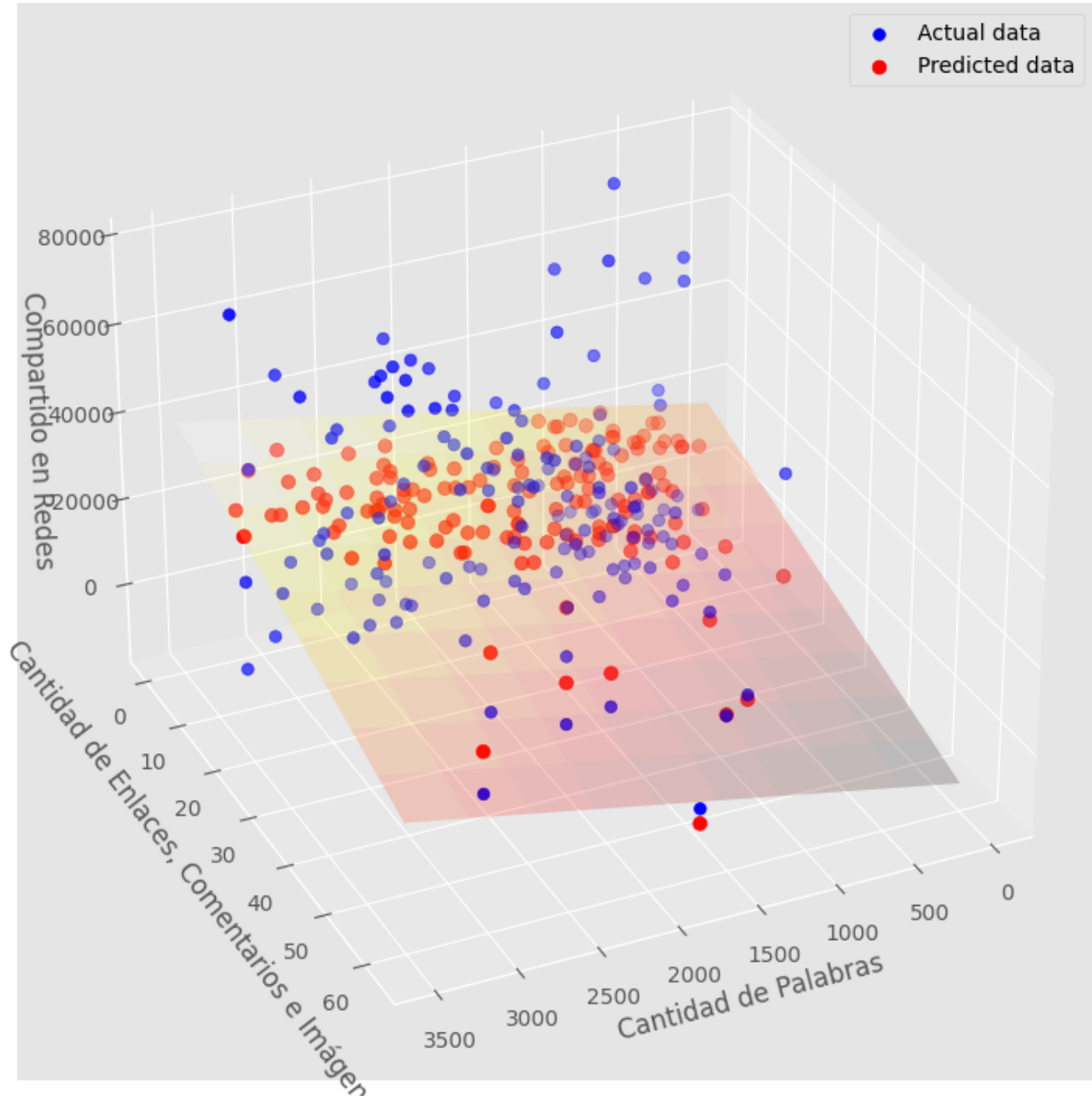
ax.set_xlabel('Cantidad de Palabras')
ax.set_ylabel('Cantidad de Enlaces, Comentarios e Imágenes')
ax.set_zlabel('Compartido en Redes')
ax.set_title('Regresión Lineal con Múltiples Variables')

ax.legend()

# Show the plot
plt.show()
```

El código crea una gráfica 3D de un modelo de regresión lineal múltiple, mostrando la superficie ajustada por el modelo, los datos reales (en azul) y los datos predichos (en rojo). Se visualiza cómo las variables independientes influyen en la variable dependiente y se incluyen etiquetas para los ejes y la leyenda.

Regresión Lineal con Múltiples Variables



4. Conclusión

El resultado de la estimación ha mejorado, pero aún está lejos de un ideal. No se aprecia demasiado en la gráfica anterior, pero siguen estando demasiado separados los valores esperados de los reales. Dándoles diferentes pesos a las variables a través de los coeficientes encontramos que unos influyen de forma considerable mientras que otros se anulan entre sí, o no aportan nada a la estimación. En este caso, tenemos solo 2 variables, 1 con un coeficiente de 6.6 y la 2 con coeficiente -483.4, es decir, por cada unidad de incremento en la variable 2 la estimación reducirá en 483.4 unidades. Podemos observar como una variables es considerablemente más determinante en el modelo que la otra.

5. Referencias

- Material de clase
- <https://www.ibm.com/docs/es/cognos-analytics/11.1.0?topic=tests-multiple-linear-regression>
- https://www.ugr.es/~montero/matematicas/regresion_lineal.pdf
- <https://www.rankia.com/diccionario/trading/regresion-lineal-multiple>