

# Regresion Lineal Múltiple

Valeria Rodríguez

23 de Marzo del 2025

## 1 Introducción

La regresión lineal múltiple es un método estadístico que relaciona múltiples variables explicativas con una variable respuesta de forma lineal. Su objetivo es encontrar una ecuación que relacione dos o más variables independientes con una variable dependiente.

## 2 Metodología

Para la realización del presente se utilizó como base lo trabajado en el documento anterior *Regresión Lineal* por el mismo autor. De la misma forma, se utilizó de base el material del libro *Aprende Machine Learning* del autor Juan Ignacio Bagnato, páginas 42-46.

### 2.1 Creación de carpeta de trabajo

Se creó una carpeta llamada 'Regresión Lineal Múltiple' en la cual se copió el archivo csv que contiene los datos de trabajo antes utilizados. Además, se creó un archivo Python llamado 'reg\_lin\_mul' donde se trabajará el ejercicio.

### 2.2 Codificación

Se reutilizó el código del trabajo anterior, desde los imports hasta la selección de datos en 'filtered\_data'. Posteriormente, se incluyeron otros elementos para mejorar la predicción del modelo, utilizando dos variables predictivas en donde la primera variable seguirá siendo la cantidad de palabras y la segunda variable será la suma de 3 columnas de entrada: la cantidad de enlaces, comentarios y cantidad de imágenes.

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

Ahora, se crea el nuevo modelo de regresión lineal, con la variable de salida z y las dimensiones a entrenar x y y. Además, se imprimen los coeficientes y puntajes obtenidos.

```
#Creamos un nuevo objeto de Regresión Lineal
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: ', 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))
```

Posteriormente, se grafica un plano en tres dimensiones, en donde las características de entrada serán de color azul y las proyecciones de color rojo. En este caso, la altura de los puntos respecto al eje Z corresponderán a la cantidad de compartidos del texto.

```
fig = plt.figure()
ax = Axes3D(fig)

# Creamos una malla, sobre la cual graficaremos el plano
xx, yy = np.meshgrid(np.linspace(0, 3500, num=10), np.linspace(0, 60, num=10))

# calculamos los valores del plano para los puntos x e y
nuevoX = (regr2.coef_[0] * xx)
nuevoY = (regr2.coef_[1] * yy)

# calculamos los correspondientes valores para z. Debemos sumar el punto
de intercepcion
z = (nuevoX + nuevoY + regr2.intercept_)
```

```

# Graficamos el plano
ax.plot_surface(xx, yy, z, alpha=0.2, cmap='hot')

# Graficamos en azul los puntos en 3D
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue',s=30)

# Graficamos en rojo, los puntos de proyeccion
ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red',s=40)

# con esto situamos la "camara" con la que visualizamos
ax.view_init(elev=20, azim=45)

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

plt.show()

```

Finalmente, se predice la cantidad de compartidos que tendrá un texto de 2000 palabras con 10 enlaces, 4 comentarios y 6 imágenes.

```

z_Dosmil = regr2.predict([[2000, 10+4+6]])
print('Numero de compartidos en un texto de 2000 palabras, 10 enlaces, 4 co-
mentarios y 6 imágenes: ', int(z_Dosmil))

```

### 3 Resultados

A continuación, se muestran los resultados obtenidos en las diversas fases de codificación mostradas en la sección anterior.

```

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

```

Figure 1: Coeficientes del modelo de regresión.

Se obtuvieron dos coeficientes correspondientes a las dos variables predictivas. El error cuadrado sigue siendo considerablemente alto, sin embargo, ha mejorado en comparación al trabajo anterior. De la misma forma, la varianza ha mejorado significativamente, pero sigue lejos del ideal.

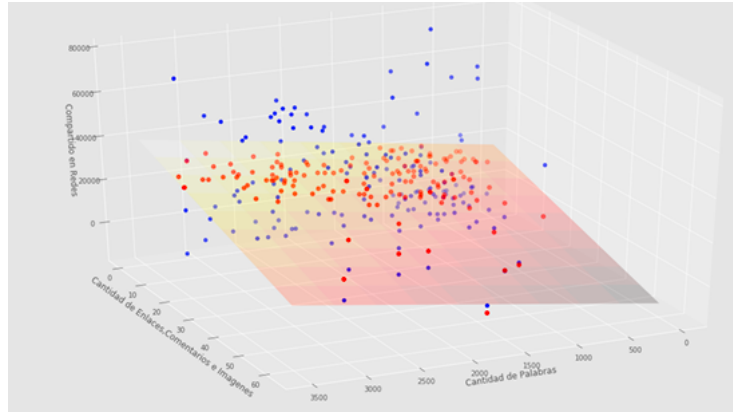


Figure 2: Gráfica de proyecciones de compartidos.

Numero de compartidos en un texto de 2000 palabras, 10 enlaces, 4 comentarios y 6 imágenes: 20518

Figure 3: Estimación de compartido para un texto de 2000 palabras.

Con el nuevo modelo, se han estimado 20518 compartidos para un texto de 2 mil palabras, incluyendo factores como enlaces, imágenes y comentarios. El modelo anterior arrojaba 22595 compartidos, por lo que se puede observar una mejoría en la estimación con el modelo actual.

## 4 Conclusión

En conclusión, considero que un modelo multivariable mejora considerablemente la predicción de una variable dado a que se están considerando más datos que aportan significativamente al resultado de un fenómeno. Si bien el modelo aún no arroja los coeficientes deseados, estos podrían mejorarse modificando el modelo, o bien, ejecutando otro tipo de modelos de regresión lineal para las predicciones.

## References

Material de clase. (2025). UANL