Actividad 10: Programando Regresión Lineal Multiple en Python

García Herrera Carlos Eduardo

March 2025

1 Introduction

La regresión lineal múltiple es una extensión de la regresión lineal simple que se utiliza para modelar la relación entre una variable dependiente y dos o más variables independientes. Esta técnica permite predecir el valor de la variable dependiente basándose en múltiples factores, proporcionando una comprensión más completa y precisa de cómo estas variables influyen en el resultado. A diferencia de la regresión lineal simple, que solo examina la relación entre dos variables, la regresión lineal múltiple toma en cuenta múltiples variables predictoras, lo que la convierte en una herramienta poderosa para análisis de datos complejos. Es ampliamente utilizada en diversos campos, como la economía, las ciencias sociales, la ingeniería y la biología, entre otros.

2 Metodología

2.1 Parte 1: Creacion del Ambiente Virtual

Las siguiente lineas de codigo, se encargan de crear un ambiente virtual llamado VirtualEnv con las librerias necesarias para ejecutar el script

```
#Automatic creation of a virtual environment to run the script and install the libraries
import subprocess
import os
import venv
import sys
script_dir = os.path.dirname(os.path.realpath(__file__))
env_name = os.path.join(script_dir, "VirtualEnv")
if os.path.exists(os.path.join(script_dir, "VirtualEnv")):
    #Checks if the VirtualEnv is activated
    if sys.prefix == sys.base_prefix:
        print("Activating the Virtual Environment...")
        python_exe = os.path.join(env_name, "Scripts", "python")
        subprocess.run([python_exe, __file__])
```

```
else:
    print("Installing the Required Libraries on a New Virtual Environment")
    venv.create(env_name, with_pip=True)

    libraries = ["scikit-learn", "matplotlib","seaborn","pandas","numpy"]
    for lib in libraries:
        subprocess.run([os.path.join(env_name, "Scripts", "pip"), "install", lib],
        check=True)

    python_exe = os.path.join(env_name, "Scripts", "python")
    subprocess.run([python_exe, __file__])
```

2.2 Parte 2: Analisis de los datos

Se cargan los datos contenidos en articulos_ml.csv y se analizan sus distintas propiedades

```
#cargamos los datos de entrada
data = pd.read_csv("./articulos_ml.csv")
#veamos cuantas dimensiones y registros contiene
print("Dimensiones de los datos:")
print(data.shape)
#Veamos los primeros registros
print("Primeros 5 registros:")
print(data.head())
# Ahora veamos algunas estadísticas de nuestros datos
print("Estadististicas de los Datos:")
print(data.describe())
# Visualizamos rápidamente las caraterísticas de entrada
data.drop(['Title','url', 'Elapsed days'],axis=1).hist()
plt.show()

# Vamos a RECORTAR los datos en la zona donde se concentran más los puntos
# esto es en el eje X: entre 0 y 3.500
# y en el eje Y: entre 0 y 80.000
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

# Vamos a pintar en colores los puntos por debajo y por encima de la media de Cantidad
de Palabras
asignar=[]
for index, row in filtered_data.iterrows():
```

```
    if(row['Word count']>1808):
        asignar.append(colores[0])
    else:
        asignar.append(colores[1])
plt.scatter(f1, f2, c=asignar, s=tamanios[0])
plt.show()
```

2.3 Parte 3: Regresion Lineal Multiple

A continuacion se hace el calculo de la regresion Lineal multiple:

```
#Regersion Lineal Multiple
#Vamos a intentar mejorar el Modelo, con una dimensión más:
# Para poder graficar en 3D, haremos una variable nueva que será la suma de los enlaces,
comentarios e imágenes

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

# 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: \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))

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# 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 intercepción
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 que
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=30., azim=65)

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')

# Mostrar gráfico
plt.show()

# Si quiero predecir cuántos "Shares" voy a obtener por un artículo con:
# 2000 palabras y con enlaces: 10, comentarios: 4, imagenes: 6
# según nuestro modelo, hacemos:

z_Dosmil = regr2.predict([[2000, 10+4+6]])
print('Prediccion para 2000 palabras:',int(z_Dosmil[0]))

input("Press any key to Exit")
```

3 Resultados

Al ejecutar el script de pyhton la informacion obtenida es la siguiente:

3.1 Analisis de los Datos

La informacion de los datos es la siguiente:

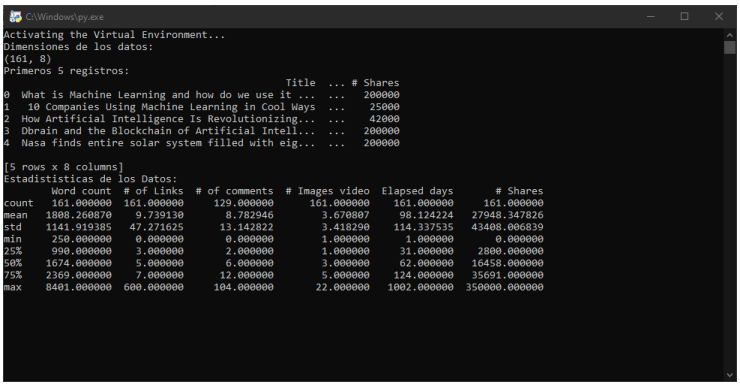

Figure 1: Informacion de los Datos

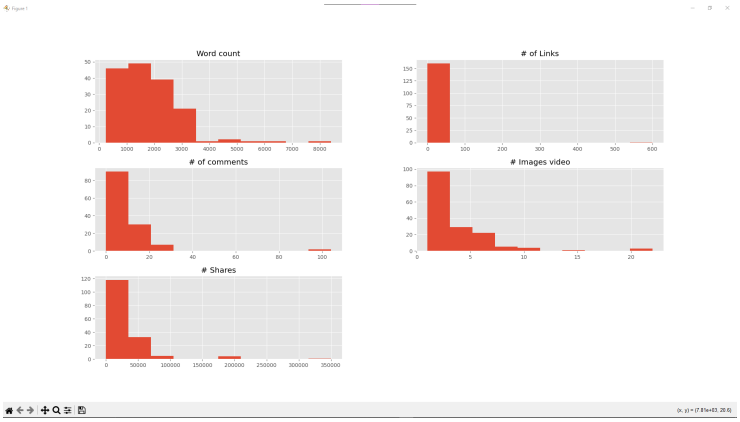

Figure 2: Numero de elementos por Post

4 Interpretacion de Resultados

La ecuación general de una recta en una regresión lineal múltiple es de la forma:

$$y = m_1x_1 + m_2x_2 + \ldots + m_nx_n + b$$

Donde:

- $m_1, m_2, \ldots, m_n$ son los coeficientes de las variables independientes $x_1, x_2, \ldots, x_n$.
- b es el término independiente o intercepto.

Sustituyendo los valores obtenidos de nuestro modelo:

$$y = 6.63216324x_1 - 483.40753769x_2 + b$$

Donde:

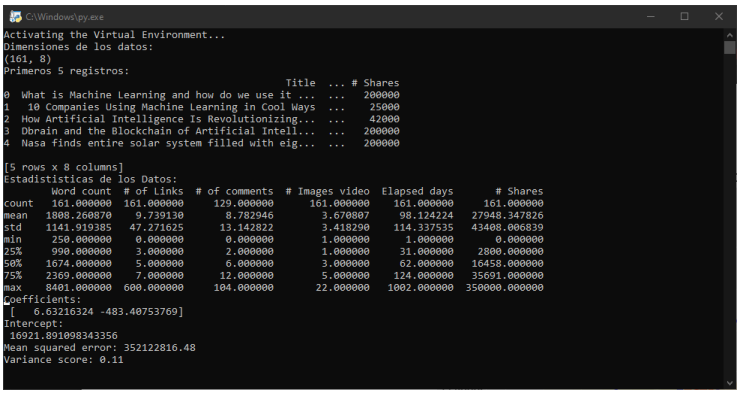

Figure 3: Datos calculados de la regresion Lineal

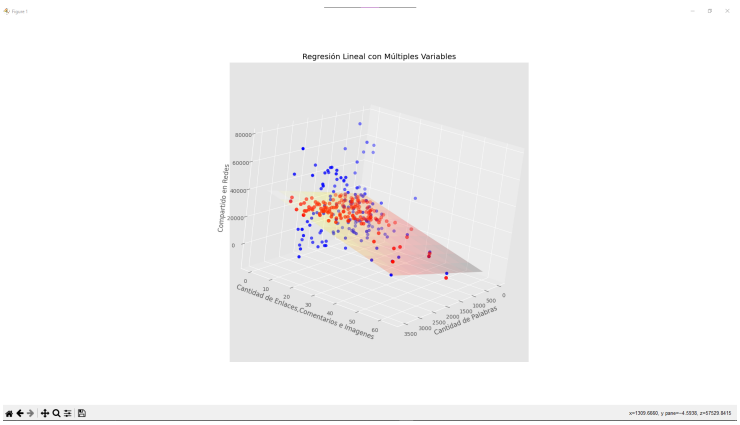

Figure 4: Grafica de la Regresion Lineal sobre los datos

- **Coeficientes (pendientes)**:

$$m_1 = 6.63216324, \quad m_2 = -483.40753769$$

- **Término independiente (intercepto)**: 16921.89
- **Error cuadrático medio**: 352122816.48
- **Puntaje de varianza ($\mathbf{R^2}$)**: 0.11

5 Conclusion

La regresión lineal múltiple es una herramienta estadística poderosa que permite analizar la relación entre una variable dependiente y múltiples variables

independientes. A través de esta técnica, podemos obtener un modelo que predice el valor de la variable dependiente en función de las variables explicativas, proporcionando una comprensión más profunda y detallada de cómo cada factor contribuye al resultado. Aunque ofrece una mayor precisión al considerar múltiples variables, es importante recordar que su efectividad depende de la calidad de los datos y de la suposición de linealidad entre las variables. Además, un puntaje de varianza (R^2) bajo puede indicar que el modelo no está capturando adecuadamente la variabilidad de los datos.