Act 9: Programando Regresión lineal en python

Ana Isabel Loera Gil

23 de marzo del 2025

1 Introduccción

Una regresión lineal es una técnica de análisis de datos, la cual nos permite predecir el valor de datos desconocidos mediante el uso de otro valor relacionado y conocido. Se modela matemáticamente la variable dependiente y la independiente como una ecuación lineal.

2 Metodología

Para comenzar a programar la regresión lineal en python debemos instalar las siguientes librerias en dado caso de que aún no contenemos con ellas:

- pandas
- seaborn
- matplotlib
- scikit-learn

Para instalarlas se debe ejecutar el siguiente comando en la consola: pip install pandas seaborn matplotlib scikit-learn Con ello las librerias se instalan en nuestra computadora una vez hecho esto podemos comenzar con el código.

2.1 Importar las librerías

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
plt.rcParams['figure.figsize']=(16,9)
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

2.2 Configurar Pandas para que muestre todas las columnas sin truncarlas

```
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 500)
pd.set_option('display.max_colwidth', None)
```

2.3 Cargar los datos de entrada y visualizarlos

```
#cargar datos de entrada
data = pd.read_csv("./articulos_ml.csv")
# Muestra las dimensiones del DataFrame
print("Forma de los datos:", data.shape)
#Muestra los primeros 5 registros
print(data.head())
#Estadísticas de los datos
print(data.describe())
#Visualización de las caracteristicas de entrada
data.drop(['Title','url','Elapsed days'],axis=1).hist()
plt.show()
```

2.4 Generación de gráfico de dispersión

```
#filtrar datos
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

#Pintar en colores los puntis por debajo y encima de la media de Catidad 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()
```

2.5 Regresion lineal con Python y SKlearn

```
#regresion lineal con python y sklearn
dataX = filtered_data[['Word count']]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values

#Creacion del objeto regresión linear
regresion= linear_model.LinearRegression()

#Entrenamiento del modelo
regresion.fit(X_train,y_train)

#Predicciones del modelo
y_pred= regresion.predict(X_train)
#Coeficientes obtenidos
print('Coeficientes: \n', regresion.coef_)
#Valor donde corta el eje Y (en X=0)
print('Terminos independientes: \n',regresion.intercept_)
#Error cuadrado medio
print('Cuadrado medio del error: %.2f' %mean_squared_error(y_train, y_pred))
#Valor de la varianza
print('Valor de la varianza: %.2f' %r2_score(y_train, y_pred))
```

2.6 Predicción en regresión lineal simple

```
y_dosmil= regresion.predict([[2000]])
print(int(y_dosmil))
```

3 Resultados

Se muestran los primeros 5 registros

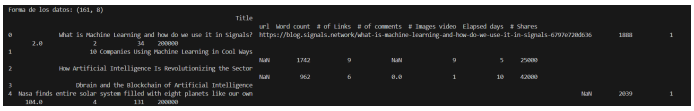

Figure 1: Registros

Las estadísticas de los datos son las siguientes:

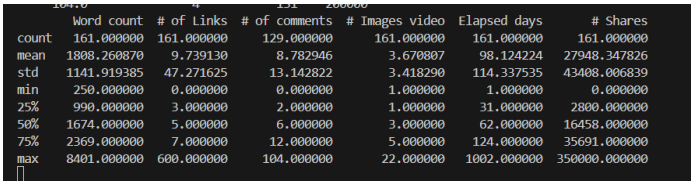

Figure 2: Enter Caption

La media de palabras en los artículos es de 1808. El artículo más corto tiene 250 palabras y el más extenso 8401. Intentaremos ver con nuestra relación lineal, si hay una correlación entre la cantidad de palabras del texto y la cantidad de Shares obtenidos.

Las gráficas de visualización de los datos son las siguientes:

Figure 3: gráfica de barras

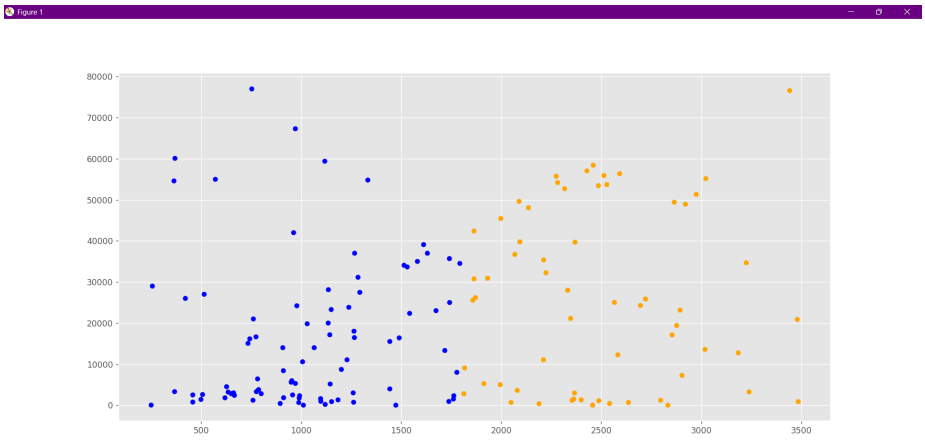

Figure 4: Gráfico de dispersión con registros con menos de 3500 palabras y que tengan Cantidad de compartidos menos a 80,000

```
Coeficientes:
 [5.69765366]
Terminos independientes:
 11200.30322307416
Cuadrado medio del error: 372888728.34
Valor de la varianza: 0.06
```

Figure 5: Regresión lineal

Como se puede observar de la ecuación de la recta y=mX+b la pendiente "m" es el coeficiente 5.69 y el término independiente "b" es 11,200. Se tiene un error cuadrático medio muy grande, por lo que este modelo no será bueno.

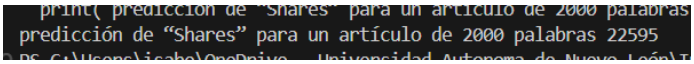

Figure 6: Predicción de 22595 "Shares" para un artículo de 2000 palabras

Lo cual significa que, según la relación aprendida entre el número de palabras y la cantidad de veces que se comparte el artículo (a través del modelo de regresión), un artículo con 2,000 palabras debería obtener aproximadamente 22,595 compartidos.

4 Conclusión

Una regresión lineal nos ayuda a predecir el valor de una variable dependiente en función de las variables independientes, en este ejemplo en particular el modelo no era el más optimpo al obtener un valor extremadamente grande, lo cual puede deberse a distintos factores, que tendremos que ir analizando y ajustando.

5 Referencias bibliograficas

https://aws.amazon.com/es/what-is/linear-regression/ Ignacio Bagnato, J. (2020). Aprende machine learning. Leanpub.