

Actividad #9: Regresión Lineal con Python

Nombre: Dayla Marely Carrizales Ortega

Matrícula: 1952471

1 Introducción

La regresión lineal simple es una regresión lineal con una variable independiente, también llamada variable explicativa, y una variable dependiente, también llamada variable de respuesta. En la regresión lineal simple, la variable dependiente es continua. La regresión lineal simple ayuda a hacer predicciones y a comprender las relaciones entre una variable independiente y una variable dependiente. Debido a su capacidad para transformar datos, pueden utilizarse para simular una amplia gama de relaciones, y debido a su forma, que es más simple que la de las redes neuronales, sus parámetros estadísticos se analizan y comparan con facilidad, lo que permite que se les extraiga información valiosa.

2 Metodología

En esta actividad, implementamos modelos de regresión lineal simple y múltiple utilizando Python y la librería Scikit-Learn. A continuación, describimos los pasos realizados y mostramos fragmentos del código.

2.1 Regresión Lineal Simple

El objetivo de este modelo es predecir el número de veces que se compartirá un artículo en redes sociales, en función de la cantidad de palabras del artículo.

Listing 1: Código para regresión lineal simple

```
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

data = pd.read_csv("./articulos_ml.csv")
```

```

data.shape
data.head()
data.describe()
data.drop(['Title', 'url', 'Elapsed days'], axis=1).hist()
plt.show()
colores=['orange','blue']
tamanios=[30,60]

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

# Vamos a pintar en 2 colores los puntos por debajo de la media de
# Cantidad de Palabras
asignar=[]
for index, row in 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()

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

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()

filtered_data.describe()

dataX =filtered_data[["Word count"]]
X_train = np.array(dataX)
y_train = filtered_data['# Shares'].values

regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(X_train, y_train)

# Hacemos las predicciones que en definitiva una l nea (en este
# caso, al ser 2D)
y_pred = regr.predict(X_train)

# Veamos los coeficientes obtenidos, En nuestro caso, ser n la

```

```

Tangente
print('Coeficientes: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
print("Mean squared error: %.2f" % mean_squared_error(y_train,
y_pred))
# Puntaje de Varianza. El mejor puntaje es un 1.0
print('Variance score: %.2f' % r2_score(y_train, y_pred))

plt.scatter(X_train[:,0], y_train, c=asignar, s=tamamos[0])
plt.plot(X_train[:,0], y_pred, color='red', linewidth=3)

plt.xlabel('Cantidad de Palabras')
plt.ylabel('Compartido en Redes')
plt.title('Regresi n Lineal')

plt.show()

y_Dosmil = regr.predict([[2000]])
print(int(y_Dosmil))

```

3 Resultados

Los resultados obtenidos son los coeficientes calculados para ambas variables, el error cuadrático medio y el puntaje de varianza. El modelo es de calidad baja en cuanto a precisión en la predicción, esto es por los datos que se utilizaron.

4 Conclusión

En esta actividad, aprendí a implementar modelos de regresión lineal en Python utilizando la librería Scikit-Learn. Algo que pude notar al realizar este programs fue que la calidad del modelo depende de la calidad de los datos de entrada, es decir, tendremos un modelo bueno o más preciso si los datos de entrada son precisos.

5 Bibliografía

- Bagnato, J. I. (2020). Aprende Machine Learning en Español Teoría + Práctica Python. Leanpub.
- Saavedra, J. A. (2022, septiembre 16). Regresión Lineal: qué es, para qué sirve, por qué es importante, tipos y ejemplos de uso. Ebac. <https://ebac.mx/blog/regreson-lineal>
- Waples, J. (2024). Regresión lineal simple. Datacamp. <https://www.datacamp.com/es/tutorial/simple-linear-regression>