

Actividad #10: Programando Regresión Lineal Múltiple en Python

Nombre: Dayla Marely Carrizales Ortega
Matricula: 1952471

1 Introducción

La regresión lineal múltiple permite generar un modelo lineal en el que el valor de la variable dependiente o respuesta (Y) se determina a partir de un conjunto de variables independientes llamadas predictores (X1, X2, X3...). Es una extensión de la regresión lineal simple, por lo que es fundamental comprender esta última. Los modelos de regresión múltiple pueden emplearse para predecir el valor de la variable dependiente o para evaluar la influencia que tienen los predictores sobre ella (esto último se debe que analizar con cautela para no malinterpretar causa-efecto).

2 Metodología

Para esta actividad, implementamos un modelo de regresión lineal múltiple en Python. El código mostrado a continuación, describe los pasos realizados para obtener los resultados deseados.

Listing 1: Código para regresión lineal múltiple

```
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")

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

regr2 = linear_model.LinearRegression()

regr2.fit(XY_train, z_train)

z_pred = regr2.predict(XY_train)

print('Coeficientes (regresi n m ltiple): \n', regr2.coef_)

print("Mean squared error (regresi n m ltiple): %.2f" %
      mean_squared_error(z_train, z_pred))

print('Variance score (regresi n m ltiple): %.2f' % r2_score(
      z_train, z_pred))

regr1 = linear_model.LinearRegression()

regr1.fit(filtered_data[['Word count']], z_train)

y_pred = regr1.predict(filtered_data[['Word count']])

z_Dosmil = regr2.predict([[2000, 10 + 4 + 6]])
print("Predicci n (regresi n m ltiple) para 2000 palabras,
      enlaces, comentarios e im genes: ", int(z_Dosmil[0]))

mejoraEnError = mean_squared_error(z_train, y_pred) -
      mean_squared_error(z_train, z_pred)
print("Mejora en el error cuadr tico medio: ", mejoraEnError)

mejoraEnVarianza = r2_score(z_train, z_pred) - r2_score(z_train,
      y_pred)
print("Mejora en el puntaje de varianza: ", mejoraEnVarianza)

closest_idx = (filtered_data['Word count'] - 2000).abs().idxmin()

y_pred_closest = y_pred[closest_idx]
z_pred_closest = z_pred[closest_idx]

diferenciaComparir = z_Dosmil[0] - z_pred_closest
print("Diferencia entre las predicciones de ambos modelos para el
      art culo con el 'Word count' m s cercano a 2000 palabras: ",
      int(diferenciaComparir))

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

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')

ax.scatter(XY_train[:, 0], XY_train[:, 1], z_train, c='blue', s=30)

ax.scatter(XY_train[:, 0], XY_train[:, 1], z_pred, c='red', s=40)

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')

plt.show()

```

Los coeficientes obtenidos reflejan la influencia de cada variable predictiva en la variable dependiente, mientras que el error cuadrático medio y el puntaje de varianza nos dan una idea del rendimiento del modelo.

3 Resultados

El modelo de regresión lineal múltiple proporciona un ajuste mejorado en comparación con la regresión lineal simple, aunque la precisión sigue siendo limitada debido a la naturaleza de los datos. La suma de las variables adicionales mejora el modelo ligeramente, como se observa en el puntaje de varianza.

4 Conclusión

En conclusión, se implementó el modelo de regresión lineal anteriormente utilizado pero con el enfoque de regresión lineal múltiple. El modelo ha mejorado un poco pero si los datos fueran más precisos se tendrían aún mejores resultados más óptimos.

5 Bibliografía

- Bagnato, J. I. (2020). Aprende Machine Learning en Español Teoría + Práctica Python. Leanpub.
- Rodrigo, J. A. (s/f). Introducción a la Regresión Lineal Múltiple. Cienciadatos.net. Recuperado el 23 de marzo de 2025, de <https://cienciadatos.net/documentos/25-regresion-lineal-multiple>