UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la Información: Desarrollo y Gestión de Software

V. Elaboración de gráficas

IDGS91N - Kevin Iván Aguirre Silva
Extracción de Conocimiento en Bases de Datos - Ing. Luis Enrique Mascote Cano

Chihuahua, Chih., 30 de noviembre de 2025

Índice

1. Introducción

El presente reporte documenta el desarrollo y la evaluación de un modelo de Regresión Lineal Múltiple utilizando las librerías pandas, numpy, matplotlib, y scikit-learn , como parte de la asignatura de Extracción de Conocimiento en Bases de Datos. El objetivo principal fue generar y analizar tres tipos de gráficas —Dispersión (Reales vs. Predichos), Residuos, e Importancia de Características— para validar el desempeño del modelo y explicar la influencia de las variables predictoras sintéticas (Feature_A, Feature_B, Feature_C) en la variable objetivo, como se desarrolla con el código ubicado en la ruta unidad3/evidencia2/script.

2. Desarrollo

Librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

1. Pandas (Pandas, s.f.).
2. NumPy (NumPy, s.f.).
3. Matplotlib (Matplotlib, s.f.).

Preparación de datos de ejemplo

```
np.random.seed(42)
X = np.random.rand(100, 3) * 10
y = 1.5 * X[:, 0] + 2.0 * X[:, 1] - 0.5 * X[:, 2] + 5 + np.random.randn(100) * 2

df = pd.DataFrame(X, columns=['Feature_A', 'Feature_B', 'Feature_C'])
df['Target'] = y
```

División de datos y entrenamiento del modelo

```
X_train, X_test, y_train, y_test = train_test_split(df[['Feature_A', 'Feature_B', 'Feature_C']], df['Target'], test_size=0.3, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Cálculo de residuos

```
residuals = y_test - y_pred
```

Gráfico de dispersión (valores reales vs. predichos)

```
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.6)
# Dibuja la línea de predicción perfecta (y=x)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel(r"Valores Reales ($Y_{test}$)")
plt.ylabel(r"Valores Predichos ($\hat{Y}$)")
plt.title("1. Dispersión: Reales vs. Predichos")
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

Gráfico de residuos (valores predichos vs. residuos)

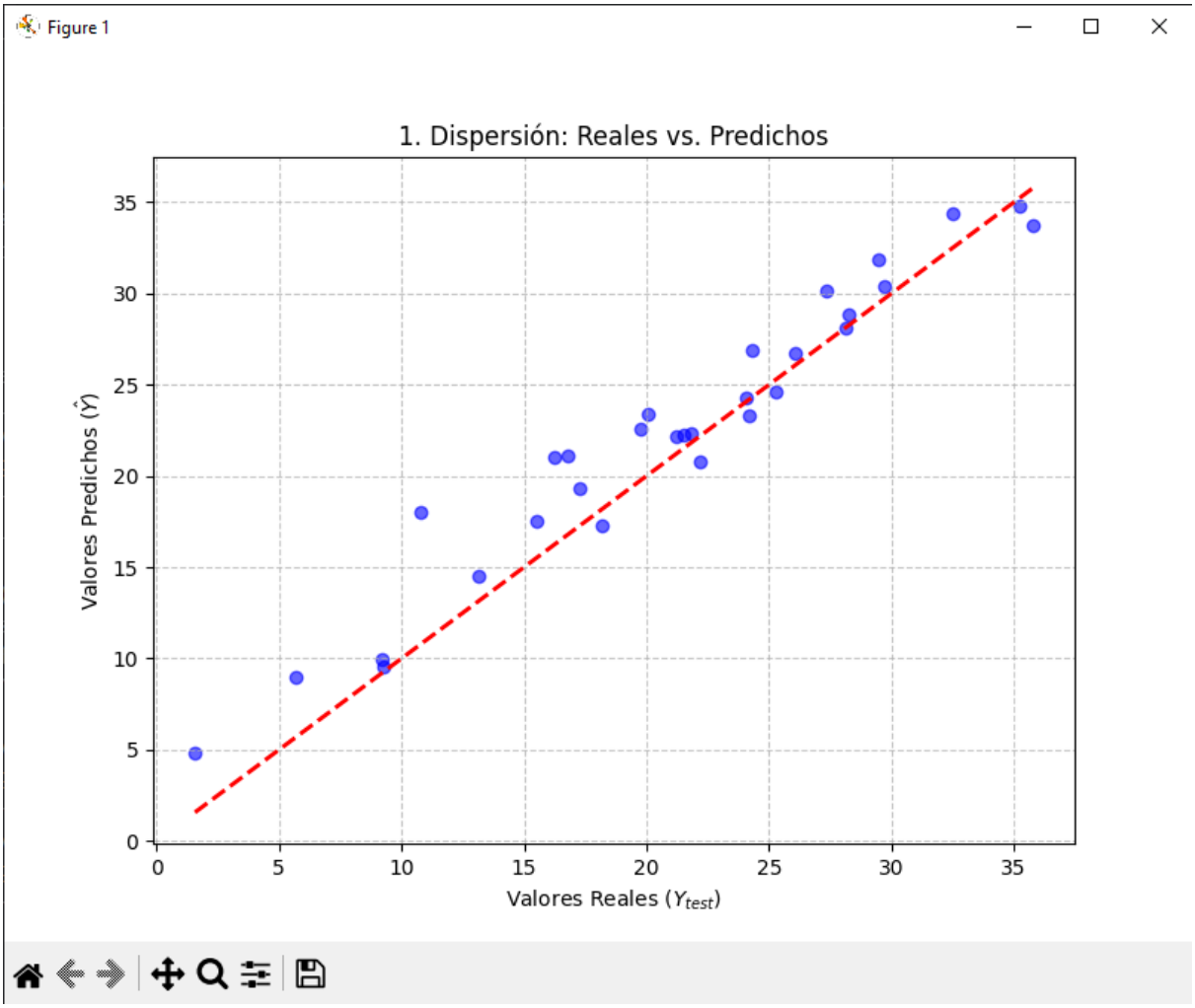
```
plt.figure(figsize=(8, 6))
plt.scatter(y_pred, residuals, color='purple', alpha=0.6)
# Dibuja la línea horizontal en Residuos = 0
plt.axhline(y=0, color='r', linestyle='-', linewidth=1)
plt.xlabel(r"Valores Predichos ($\hat{Y}$)")
plt.ylabel(r"Residuos ($Y_{real} - \hat{Y}$)")
plt.title("2. Gráfico de Residuos")
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

Gráfico de barras de coeficientes/importancia de características

```
# Usamos el intercepto y los coeficientes del modelo
feature_names = X_train.columns.tolist()
coefficients = [model.intercept_] + model.coef_.tolist()
labels = ['Intercepto'] + feature_names

# Convertimos a serie para facilitar el ploteo
coef_series = pd.Series(coefficients, index=labels)

plt.figure(figsize=(10, 6))

# Usamos el valor absoluto para ordenar y ver la magnitud del impacto
coef_series.abs().sort_values(ascending=False).plot(kind='bar', color='darkorange')
plt.ylabel("Magnitud Absoluta del Coeficiente")
plt.title("3. Importancia de las Características (Magnitud Absoluta de Coeficientes)")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

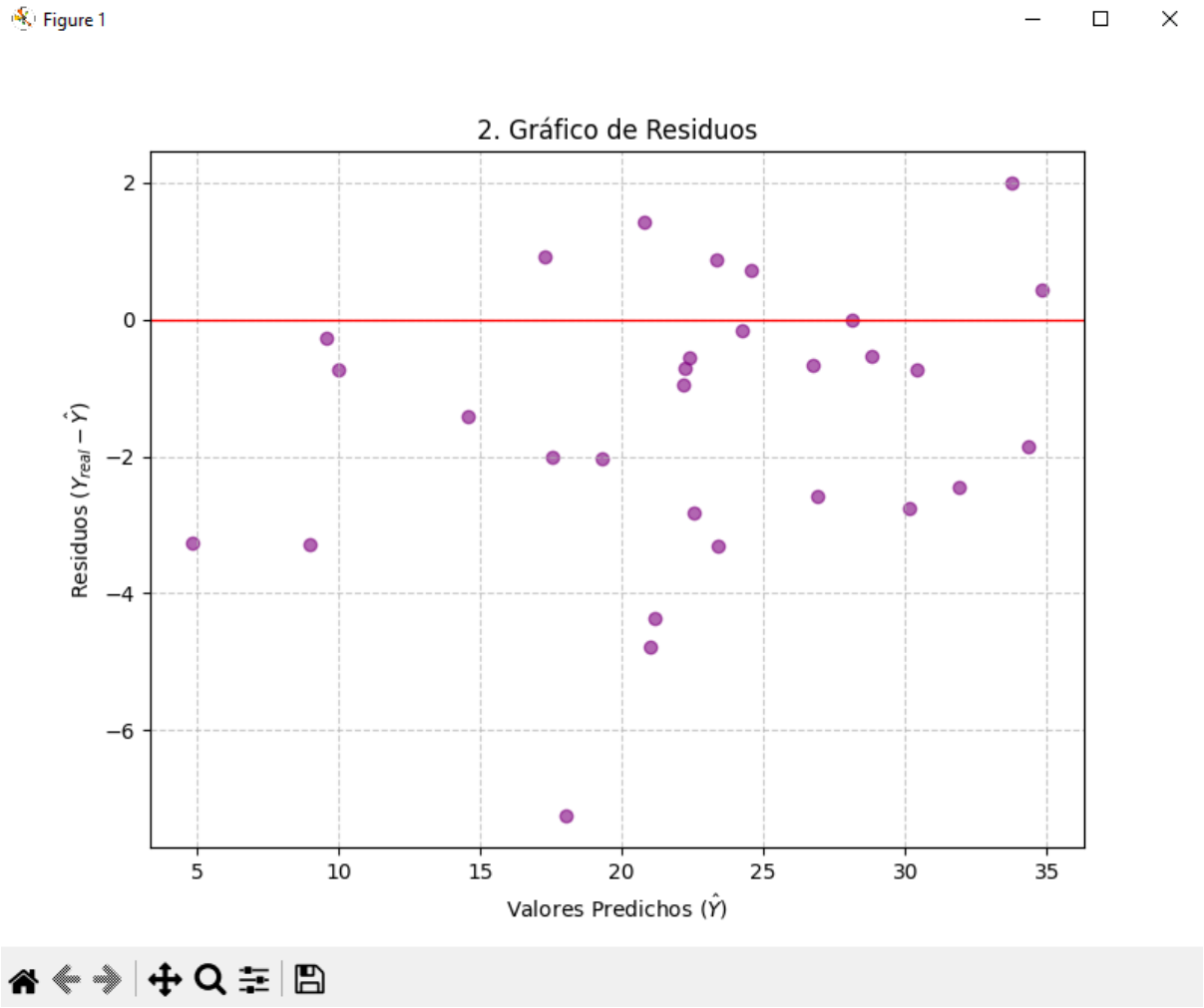
3. Resultados

Gráfico de dispersión (valores reales vs. predichos)



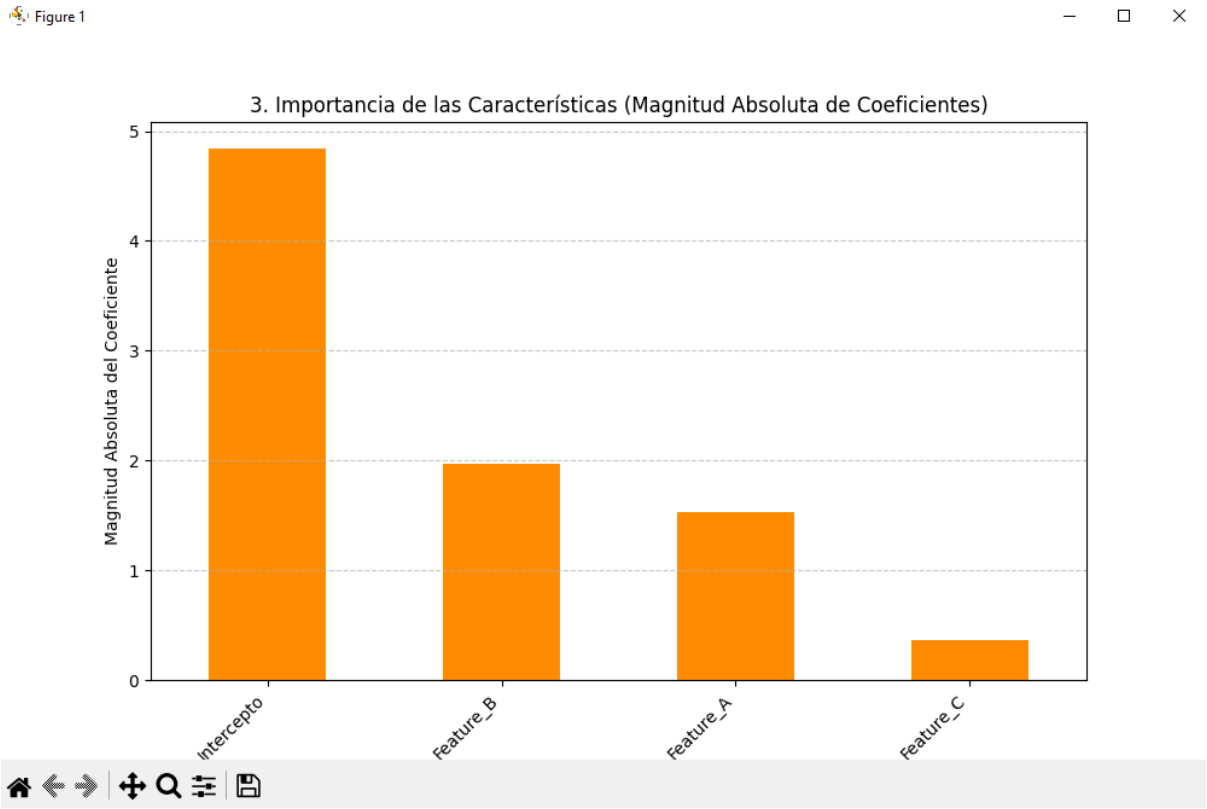
El gráfico muestra una fuerte correlación lineal positiva entre los valores reales y los predichos, lo cual es excelente. La mayoría de los puntos azules se agrupan muy cerca de la línea roja ideal, indicando que el modelo de Regresión Lineal tiene una alta precisión. Solo unos pocos puntos se desvían, representando los errores más grandes de la predicción. En general, este resultado sugiere un ajuste muy bueno a los datos de prueba (Efren, 2024).

Gráfico de barras de coeficientes/importancia de características



El gráfico muestra que los residuos están distribuidos aleatoriamente por encima y por debajo de la línea de cero. No se observan patrones definidos (como curvas o formas de embudo), lo cual es una excelente señal para el modelo. La dispersión vertical de los errores parece constante (homocedasticidad) a lo largo del eje X. Esto valida las suposiciones clave de la regresión y confirma que tu modelo lineal es apropiado para los datos (Miro, s.f.).

Gráfico de barras de coeficientes/importancia de características



Este gráfico muestra la magnitud del impacto que cada variable tiene en la predicción del modelo. La altura de la barra indica la importancia, siendo el Intercepto la base (predicción sin características). De las variables predictoras, Feature_B tiene el mayor coeficiente, lo que la convierte en la característica más influyente en el valor objetivo. En contraste, Feature_C tiene el menor impacto predictivo (Miro, s.f.).

4. Conclusiones

Los resultados gráficos confirman que el modelo de Regresión Lineal desarrollado es altamente efectivo y válido. El Gráfico de Dispersión demostró una alta precisión, con puntos agrupados firmemente alrededor de la línea de predicción ideal. Además, el Gráfico de Residuos validó las suposiciones clave al mostrar una distribución aleatoria y constante de los errores. Finalmente, el análisis de coeficientes identificó a Feature_B como la característica con el mayor poder predictivo sobre el valor objetivo , mientras que Feature_C presentó el menor impacto. Estos hallazgos aseguran la robustez y la interpretabilidad del modelo de regresión.

5. Referencias

Efren. (28 de Mayo de 2024). *Diagrama de dispersión: qué es y cómo se hace*. Obtenido de Venngage: https://es.venngage.com/blog/diagrama-de-dispersion/

Matplotlib. (s.f.). Obtenido de Matplotlib: https://matplotlib.org/

Miro. (s.f.). *Gráfico de barras*. Obtenido de miro: https://miro.com/es/graficos/que-es-grafico-barras/

NumPy. (s.f.). Obtenido de NumPy: https://numpy.org/

Pandas. (s.f.). *pandas - Python Data Analysis Library*. Obtenido de https://pandas.pydata.org/