**TERCER BLOQUE:**

**# 1) Tomar las métricas desde variables existentes o usar fallback con tus números**

* **Error handling:** La excepción Exception es demasiado amplia y puede atrapar errores que no se han considerado en la lógica del programa. Es recomendable especificar el tipo de error esperado (por ejemplo, ValueError) para obtener un mejor error message.
* **Fallback values**: Los valores de fallback están hardcodeados en el código. Si estos valores cambian, es importante actualizarlos en varios lugares. Considera pasar los valores como parámetros o utilizar una función que devuelva estos valores.
* **Variable naming:** Algunas variables tienen nombres que no son descriptivos (por ejemplo, row\_cv). Considera renombrarlas para hacer que el código sea más fácil de entender.
* **Dataframe creation:** La creación del dataframe df\_comp puede ser optimizada mediante la utilización de pandas' concat o merge funciones en lugar de crear un diccionario y luego convertirlo a dataframe.
* **Graphing:** El gráfico creado es simple, pero podría ser mejorado con etiquetas más descriptivas, título más informativo y legendas personalizadas.

En código se ve:

import pandas as pd

# ...

try:

best\_k = 8

row\_cv = summary\_df.loc[summary\_df['k'] == best\_k].iloc[0]

rmse\_cv, mae\_cv, r2\_cv, medae\_cv = float(row\_cv['RMSE\_mean']), float(row\_cv['MAE\_mean']), float(row\_cv['R2\_mean']), float(row\_cv['MedAE\_mean'])

rmse\_test, mae\_test, r2\_test, medae\_test = 1.6992, 1.0541, 0.8742, 0.6708

except ValueError:

print("Error al obtener métricas de CV y test")

# ...

data = {

"Métrica": ["RMSE", "MAE", "R²", "MedAE"],

"CV (promedio 5 folds, k=8)": [rmse\_cv, mae\_cv, r2\_cv, medae\_cv],

"Test externo (holdout 20%, k=8)": [rmse\_test, mae\_test, r2\_test, medae\_test],

}

df\_comp = pd.concat([pd.DataFrame(data), pd.DataFrame({"Diferencia (Test - CV)": df\_comp.iloc[:, 1] - df\_comp.iloc[:, 0]})], axis=1)

print("=== Comparación CV vs Test externo ===")

print(df\_comp.to\_string(index=False, float\_format=lambda x: f"{x:.4f}"))

# ...

metrics = df\_comp["Métrica"].tolist()

cv\_vals = df\_comp.loc[0, "CV (promedio 5 folds, k=8)"].values

test\_vals = df\_comp.iloc[:, 1].values

x = np.arange(len(metrics))

width = 0.35

plt.figure(figsize=(8, 5))

plt.bar(x - width/2, cv\_vals, width, label="CV (5 folds)")

plt.bar(x + width/2, test\_vals, width, label="Test externo")

plt.xticks(x, metrics)

plt.ylabel("Valor")

plt.title("KNN Regresión — Comparación CV vs Test externo (k=8)")

plt.legend()

plt.tight\_layout()

plt.show()