1. ¿Qué errores encontraste en el código proporcionado y cómo los corregiste?

En la funcion generar_datos_ventas(), en la frecuencia (freq='B') se utiliza para generar fechas, pero no es la que se usa para generar fechas mensuales esa es la (freq='B')

Tambien en este codigo np.random.randint(2000, 500, size=len(meses)), estaban al reves los rangos solo los inverti. (500, 2000, size=len(meses))

El archivo codigo de ventas .csv se guarda en la carpeta content que es la carpeta de google colab
datos_ventas.to_csv(f'/content/{nombre_empresa}.csv')
Ahora al leer el archivo .csv con el siguiente codigo datos_polleria = pd.read_csv('/wrong_path/Polleria.csv')
Se lee Polleria.csv pero en la carpeta “wrong_path” que no existe, aparte que le falto una tilde a Pollería, ya que anteriormente se definio a la empresa como empresa = 'Pollería'

X_polleria = datos_polleria['Meses'].values.reshape(-1, 2)
X_polleria = datos_polleria['Meses'].map(pd.Timestamp.toordinal).values.reshape(-1, 1)

productos = ['Pollo a la brasa', 'Papitas fritas', 'Ensalada', 'Jugo'], en esta parte agregar este codigo :
datos_polleria['Meses'] = pd.to_datetime(datos_polleria['Meses'])

2. Explica por qué es importante normalizar los datos antes de entrenar un modelo de `Deep Learning`.

mejor rendimiento del entrenamiento, al tener todas las características en el mismo rango, el modelo aprende de manera más eficiente, y es mas rapido.

3. ¿Por qué el uso de `Sequential` en `Keras` es adecuado para este tipo de problema?

Tiene una Arquitetura simple y directa, aparte que es mas escalable.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
np.random.seed(42)
def generar_datos_ventas(nombre_empresa, productos, start, end):
 meses = pd.date_range(start=start, end=end, freq='M')
 datos_ventas = pd.DataFrame({
       'Meses': meses,
 **{producto: np.random.randint(500, 2000, size=len(meses)) for producto in productos}
 })
 datos_ventas.to_csv(f'{nombre_empresa}.csv')
 print(f'Datos generados para {nombre_empresa}')
empresa = 'Pollería'
productos = ['Pollo a la brasa', 'Papitas fritas', 'Ensalada', 'Gaseosa']
generar_datos_ventas(empresa, productos, '2020-01-01', '2024-09-30')
datos_polleria = pd.read_csv('Pollería.csv')
datos_polleria['Meses'] = pd.to_datetime(datos_polleria['Meses'])
X_polleria = datos_polleria['Meses'].map(pd.Timestamp.toordinal).values.reshape(-1, 1)
Y_polleria = datos_polleria[productos].values
scaler = MinMaxScaler(feature_range=(-1, 1))
Y_polleria_scaled = scaler.fit_transform(Y_polleria)
modelo_polleria = Sequential()
modelo_polleria.add(Dense(32, activation='relu', input_shape=(1,)))
modelo_polleria.add(Dense(32, activation='relu'))
modelo_polleria.add(Dense(4))
modelo_polleria.compile(optimizer='sgd', loss='mean_absolute_error')
modelo_polleria.fit(X_polleria, Y_polleria_scaled, epochs=50, verbose=1)
meses_2025 = pd.date_range(start='2025-01-01', end='2025-12-31', freq='M')
X_2025 = np.arange(len(X_polleria), len(X_polleria) + len(meses_2025)).reshape(-1, 1)
predicciones_polleria_scaled = modelo_polleria.predict(X_2025)
predicciones_polleria = scaler.inverse_transform(predicciones_polleria_scaled)
ventas_2025 = predicciones_polleria[10]
plt.bar(productos, ventas_2025, color=['red', 'green', 'blue', 'purple'])
plt.title('Predicción de ventas en 2025 - Pollería')
plt.ylabel('Ventas proyectadas')
plt.xlabel('Productos')
plt.show()
```

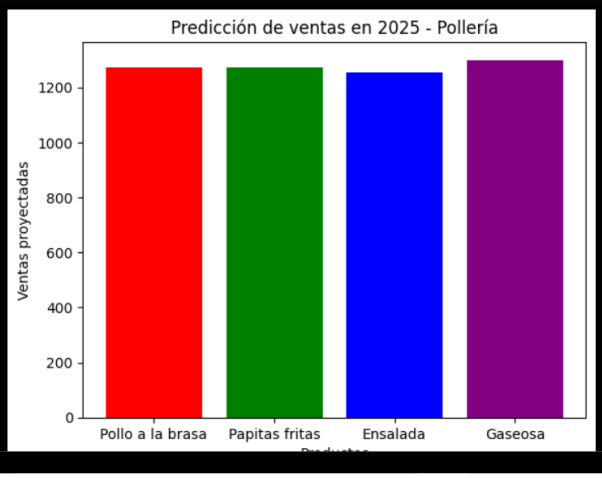
practicas > pollitoconpapa.ipynb > import numpy as np

+ Código + Markdown | Ejecutar todo Reiniciar Borrar todas las salidas | Variables Esquema ··· juanito (Python 3.12.4)

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5115
Epoch 13/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5154
...
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4956
Epoch 50/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5098
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
C:\Users\juanito\AppData\Local\Temp\ipykernel_22492\3994408351.py:35: FutureWarning: 'M' is deprecated and will be removed in a future version, please use '
  meses_2025 = pd.date_range(start='2025-01-01', end='2025-12-31', freq='M')
```



Spaces: 4 CRLF Celda 1 de 1 { }