

# PRÁCTICA CALIFICADA

Tema: Predicción de Ventas en una Pollería utilizando Deep Learning y Visualización de Datos

Esta práctica está orientada a aplicar tus conocimientos en el manejo de datos utilizando librerías de Python como `Pandas`, `Matplotlib`, y técnicas de `Deep Learning` con `Keras` para realizar predicciones. Debes analizar los datos generados de las ventas mensuales de una pollería y realizar predicciones de las ventas para el año 2025. Además, se requiere identificar y corregir errores en el código proporcionado.

Objetivos:

1. Crear un archivo CSV con datos simulados de ventas mensuales de productos en una pollería.
2. Cargar y manipular los datos utilizando `Pandas`.
3. Implementar un modelo de `Deep Learning` con `Keras` para predecir las ventas en el año 2025.
4. Visualizar los resultados en un gráfico de barras.
5. Identificar y corregir errores en el código proporcionado.

Problema:

Una pollería ha registrado sus ventas mensuales de los siguientes productos entre enero de 2020 y septiembre de 2024:

- Pollo a la brasa
- Papitas fritas
- Ensalada
- Gaseosa

Se te solicita realizar las siguientes tareas:

1. Generación de datos de ventas:

- Simula los datos de ventas mensuales para los productos indicados desde enero de 2020 hasta septiembre de 2024.
- Guarda estos datos en un archivo CSV llamado `Polleria.csv`.

2. Lectura de datos:

- Lee el archivo CSV que contiene los datos de ventas utilizando `Pandas`.

3. Modelo de predicción:

- Implementa un modelo de `Deep Learning` utilizando `Keras` para predecir las ventas de estos productos en el año 2025.

#### 4. Visualización de resultados:

- Grafica las ventas proyectadas para cada producto en el año 2025 usando un gráfico de barras.

#### 5. Corrección de errores:

- El código que se proporciona a continuación contiene varios errores que debes identificar y corregir. Una vez corregido, el código debe ejecutarse correctamente y cumplir con los objetivos planteados.

```
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(1234)

def generar_datos_ventas(nombre_empresa, productos, start, end):
    meses = pd.date_range(start=start, end=end, freq='B')
    datos_ventas = pd.DataFrame({
        'Meses': meses,
        **{producto: np.random.randint(2000, 500, size=len(meses)) for producto in productos}
    })
    datos_ventas.to_csv(f'/content/{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('/wrong_path/Polleria.csv')

productos = ['Pollo a la brasa', 'Papitas fritas', 'Ensalada', 'Jugo']

X_polleria = datos_polleria['Meses'].values.reshape(-1, 2)
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=(2,)))
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()
```

Preguntas :

1. ¿Qué errores encontraste en el código proporcionado y cómo los corregiste?
2. Explica por qué es importante normalizar los datos antes de entrenar un modelo de `Deep Learning`.
3. ¿Por qué el uso de `Sequential` en `Keras` es adecuado para este tipo de problema?